

# Microsoft System Center 2012 R2

## **Cmdlet Reference for System Center 2012 R2 Operations Manager**

---

Microsoft Corporation

Published: November 1, 2013

### **Applies To**

System Center 2012 R2 Operations Manager

### **Feedback**

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

# Copyright

---

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

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

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

© 2013 Microsoft Corporation. All rights reserved.

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

## Revision History

Release Date	Changes
November 1, 2013	Initial release of this document.

---

## Contents

Add-SCOMADAgentAssignment .....	9
Add-SCOMAgentlessManagedComputer.....	15
Add-SCOMAlertResolutionState .....	18
Add-SCOMConnector .....	22
Add-SCOMNotificationChannel .....	27
Add-SCOMNotificationSubscriber .....	41
Add-SCOMNotificationSubscription.....	45
Add-SCOMRunAsAccount.....	55
Add-SCOMRunAsProfile .....	71
Add-SCOMSubscriberSchedule .....	77
Add-SCOMTierConnector.....	84
Add-SCOMTieredManagementGroup .....	89
Add-SCOMUserRole .....	96
Approve-SCOMPendingManagement .....	104
Clear-SCOMSubscriberSchedule .....	107
Deny-SCOMPendingManagement .....	111
Disable-SCOMAgentProxy .....	114
Disable-SCOMDiscovery .....	117
Disable-SCOMMonitor .....	122
Disable-SCOMNotificationSubscription .....	127
Disable-SCOMOperationalDataReporting.....	130
Disable-SCOMRule .....	132
Enable-SCOMAgentProxy .....	137
Enable-SCOMDiscovery.....	140
Enable-SCOMMonitor.....	145

---

Enable-SCOMNotificationSubscription .....	150
Enable-SCOMOperationalDataReporting.....	152
Enable-SCOMRule .....	154
Exit-SCOMCEIP .....	159
Export-SCOMEffectiveMonitoringConfiguration .....	162
Export-SCOMManagementPack .....	167
Get-SCOMAccessLicense .....	170
Get-SCOMADAgentAssignment.....	174
Get-SCOMAgent.....	178
Get-SCOMAgentApprovalSetting .....	182
Get-SCOMAgentlessManagedComputer .....	185
Get-SCOMAlert.....	189
Get-SCOMAlertHistory .....	197
Get-SCOMAlertResolutionSetting .....	199
Get-SCOMAlertResolutionState .....	202
Get-SCOMClass .....	206
Get-SCOMClassInstance .....	211
Get-SCOMCommand .....	216
Get-SCOMConnector .....	219
Get-SCOMDatabaseGroomingSetting .....	223
Get-SCOMDataWarehouseSetting.....	226
Get-SCOMDiagnostic .....	229
Get-SCOMDiscovery .....	234
Get-SCOMErrorReportingSetting .....	239
Get-SCOMEvent.....	242
Get-SCOMGatewayManagementServer .....	248
Get-SCOMGroup .....	251

---

Get-SCOMHeartbeatSetting .....	255
Get-SCOMLicense.....	258
Get-SCOMLocation .....	261
Get-SCOMMaintenanceMode .....	267
Get-SCOMManagementGroup .....	271
Get-SCOMManagementGroupConnection.....	274
Get-SCOMManagementPack .....	277
Get-SCOMManagementServer .....	283
Get-SCOMMonitor .....	286
Get-SCOMNotificationChannel.....	293
Get-SCOMNotificationSubscriber .....	296
Get-SCOMNotificationSubscription .....	299
Get-SCOMOverride .....	303
Get-SCOMOverrideResult .....	310
Get-SCOMParentManagementServer.....	316
Get-SCOMPendingManagement.....	319
Get-SCOMRecovery .....	322
Get-SCOMRelationship .....	327
Get-SCOMRelationshipInstance .....	333
Get-SCOMReportingSetting .....	337
Get-SCOMResourcePool .....	340
Get-SCOMRMSEmulator.....	346
Get-SCOMRule.....	349
Get-SCOMRunAsAccount .....	355
Get-SCOMRunAsDistribution .....	359
Get-SCOMRunAsProfile .....	363
Get-SCOMTask .....	368

---

Get-SCOMTaskResult .....	374
Get-SCOMTierConnector .....	379
Get-SCOMTieredManagementGroup.....	384
Get-SCOMUserRole .....	389
Get-SCOMWebAddressSetting .....	393
Import-SCOMManagementPack.....	396
Install-SCOMAgent.....	401
Join-SCOMCEIP .....	406
New-SCOMLocation .....	409
New-SCOMManagementGroupConnection .....	413
New-SCOMResourcePool.....	416
Remove-SCOMADAgentAssignment .....	420
Remove-SCOMAgentlessManagedComputer.....	425
Remove-SCOMAlertResolutionState .....	428
Remove-SCOMConnector.....	432
Remove-SCOMDisabledClassInstance.....	435
Remove-SCOMLocation.....	438
Remove-SCOMManagementGroupConnection .....	444
Remove-SCOMManagementPack .....	447
Remove-SCOMNotificationChannel .....	450
Remove-SCOMNotificationSubscriber .....	454
Remove-SCOMNotificationSubscription.....	458
Remove-SCOMResourcePool.....	462
Remove-SCOMRMSEmulator .....	466
Remove-SCOMRunAsAccount.....	469
Remove-SCOMRunAsProfile .....	473
Remove-SCOMSubscriberSchedule .....	477

---

Remove-SCOMTierConnector.....	482
Remove-SCOMTieredManagementGroup .....	486
Repair-SCOMAgent.....	490
Resolve-SCOMAlert .....	493
Set-SCOMAgentApprovalSetting .....	500
Set-SCOMAgentlessManagedComputer.....	505
Set-SCOMAlert .....	508
Set-SCOMAlertResolutionSetting.....	516
Set-SCOMDatabaseGroomingSetting .....	520
Set-SCOMDataWarehouseSetting .....	527
Set-SCOMErrorReportingSetting .....	531
Set-SCOMHeartbeatSetting .....	536
Set-SCOMLicense .....	540
Set-SCOMLocation.....	544
Set-SCOMMaintenanceMode.....	550
Set-SCOMManagementGroupConnection .....	554
Set-SCOMParentManagementServer .....	556
Set-SCOMReportingSetting.....	561
Set-SCOMResourcePool.....	565
Set-SCOMRMSEmulator .....	573
Set-SCOMRunAsDistribution.....	577
Set-SCOMRunAsProfile .....	584
Set-SCOMUserRole .....	590
Set-SCOMWebAddressSetting.....	597
Start-SCOMMaintenanceMode.....	602
Start-SCOMTask .....	606
Test-SCOMCEIP .....	610

---

Test-SCOMOperationalDataReporting.....	613
Uninstall-SCOMAgent.....	616
Update-SCOMADAgentAssignment.....	619
Update-SCOMLocation.....	624
Update-SCOMRunAsAccount .....	629
Write-SCOMCommand.....	638

# Add-SCOMADAgentAssignment

---

## Add-SCOMADAgentAssignment

Assigns AD DS agent-managed computers 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 assigns one or more Active Directory Domain Services (AD DS) agent-managed computers to the System Center 2012 – Operations Manager management group.

Before you use this cmdlet to assign agent-managed computers to the management group, use MOMADAdmin.exe to create an AD DS container for an Operations Manager management group in the domain of the computers.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-Domain<String>**

Specifies the name of the domain or domain controller 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[]>**

Specifies an array of names of computers. Operations Manager excludes the computers that you specify from the agent assignment.

Aliases	none
Required?	false

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

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

Specifies an array of names of **ManagementServer** objects. This parameter specifies the management servers that Operations Manager uses as failovers for the target agents.

By default, Operations Manager uses all non-gateway management servers for failover.

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

### **-LdapQuery<String>**

Specifies the LDAP query in the domain which selects the target agent computers.

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

### **-PrimaryServer<ManagementServer>**

Specifies a **ManagementServer** object. This parameter specifies the primary management server for the target agent-managed computer. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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

### **-RunAsProfile<ManagementPackSecureReference>**

Specifies the Run As profile that Operations Manager uses to run AD DS query workflow.

The Run As profile must be associated with the same RunAs account that you specify when you run MOMADAdmin.exe to create an AD DS container for a System Center 2012 – Operations Manager management group.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Assign AD DS agent-managed computers to the management group**

This example assigns an AD DS agent-managed computer to the management group.

The first command gets the management server object named DivMgmtSvr01.fabrikam.net, and stores the object in the \$PrimaryServer variable.

---

The second command specifies an LDAP query that selects computers with names like SQLServer\*.fabrikam.net, and stores the object in the \$Ldap variable.

The third command assigns the AD DS agent-managed computers stored in \$Ldap to the management group in the fabrikam.net domain. The command excludes the computer named SQLServer05.fabrikam.net from the management group.

```
PS C:\> $PrimaryServer = Get-SCOMManagementServer -Name "DivMgmtSvr01.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: Set failovers for an AD DS agent assignment

This example assigns an AD DS agent-managed computer to the management group and sets the failovers for the agent.

The first command gets the management server object named DivMgmtSvr01.fabrikam.net, and stores the object in the \$PrimaryServer variable.

The second command gets the management servers with a name like FabrikamSecondary\*.fabrikam.net, and stores the objects in the \$Failovers variable.

The third command gets the Runs As profile object named "Active Directory Based Agent Assignment Account" and stores the object in the \$RunAs variable.

The fourth command specifies an LDAP query that selects computers with names like SQLServer\*.fabrikam.net, and stores the result in the \$Ldap variable.

The fifth command assigns the AD DS agent-managed computers stored in \$Ldap to the management group in the fabrikam.net domain. The command specifies the primary management server stored in \$PrimaryServer for the agent-managed computers. The command specifies the management servers stored in \$Failovers as failovers for the agent-managed computers. The command specifies the Run As profile stored in \$RunAs for the agent-managed computers.

```
PS C:\> $PrimaryServer = Get-SCOMManagementServer -Name "DivMgmtSvr01.fabrikam.net"
PS C:\> $Failovers = Get-SCOMManagementServer -Name "FabrikamSecondary*.fabrikam.net"
PS C:\> $RunAs = Get-SCOMRunAsProfile -Name "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
```

## Related topics

[Get-SCOMADAgentAssignment](#)

[Update-SCOMADAgentAssignment](#)

[Remove-SCOMADAgentAssignment](#)

---

# Add-SCOMAgentlessManagedComputer

---

## Add-SCOMAgentlessManagedComputer

Adds agentless-managed computers 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 one or more managed computers that do not have System Center 2012 – Operations Manager agents to a management group. Specify the *ManagedByAgent* parameter to have a managed agent, called a proxy agent, monitor the agentless managed computers. Specify the *ManagedByManagementServer* parameter to have a management server monitor the agentless-managed computers.

### Parameters

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

Specifies an array of names of Domain Name System (DNS) hosts. This parameter specifies the agentless-managed computers that you want to add to the management group.

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

---

## **-ManagedByAgent<AgentManagedComputer>**

Specifies an **AgentManagedComputer** object. This parameter specifies the Operations Manager agent that performs agentless monitoring of the agentless-managed computers. The action account of the agent that performs the monitoring must have local administrative rights on the computer that it monitors.

To obtain an **AgentManagedComputer** object, use the **Get-SCOMADAgent** cmdlet.

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

## **-ManagedByManagementServer<ManagementServer>**

Specifies a **ManagementServer** object. This parameter specifies the primary management server that performs agentless monitoring of the agentless-managed computers. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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: Add an agentless-managed computer to a management group

This command specifies that the management server named MgmtServer01 performs agentless monitoring of the computer named server01. The *Confirm* parameter specifies that the cmdlet prompts the user to confirm the operation before the command runs.

```
PS C:\> Add-SCOMAgentlessManagedComputer -DNSHostName "server01.contoso.com" -  
ManagedByManagementServer (Get-SCOMManagementServer -ComputerName  
"MgmtServer01.Contoso.com") -Confirm
```

### Related topics

[Get-SCOMAgentlessManagedComputer](#)

[Set-SCOMAgentlessManagedComputer](#)

---

# 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 for the management group.

### Parameters

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

Specifies an array of names of computers to establish connections with. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

## **-Name<String>**

Specifies a name for the alert resolution state.

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

## **-ResolutionStateCode<Byte>**

Specifies a resolution state ID. Operations Manager defines two resolution states: New (0) and Closed (255). You can assign custom resolution states any value from 2 through 254.

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

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

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

Specifies a shortcut key for applying 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: Add a new alert resolution state**

This command adds a new alert resolution state named Investigating that has the resolution state code 10.

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

## **Related topics**

[Get-SCOMAlertResolutionState](#)

[Remove-SCOMAlertResolutionState](#)

---

# Add-SCOMConnector

---

## Add-SCOMConnector

Creates an Operations Manager 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 creates a (CI) connector for System Center 2012 – Operations Manager. You can use a CI connector to synchronize data between Operations Manager and other management systems.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

## **-Description<String>**

Specifies a description of the connector. If you do not specify a description, the parameter defaults to the value of the **DisplayName** parameter.

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

## **-DiscoveryDatalsManaged**

Indicates that Operations Manager manages the discovered data.

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

---

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

### **-DiscoveryDatalsShared**

Indicates that Operations Manager shares the discovered data.

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

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

Specifies a display name for the connector.

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

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

Specifies the GUID of the connector.

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

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

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

Specifies 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 an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

### **<CommonParameters>**

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

---

## Examples

### Example 1: Create a connector

This command creates a connector named DivCustomConnector.

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

### Related topics

[Get-SCOMConnector](#)

[Remove-SCOMConnector](#)

---

# Add-SCOMNotificationChannel

---

## Add-SCOMNotificationChannel

Adds a notification channel in Operations Manager.

### 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-SCOMNotificationChannel** cmdlet adds a notification channel in System Center 2012 – Operations Manager. Notification channels inform administrators of an alert, or they run automation in response to an alert. A notification channel uses a delivery mechanism in Operations Manager, such as email, instant messaging (IM), Short Message Service (SMS), or command, to deliver notifications.

---

## Parameters

### **-Anonymous**

Indicates that an anonymous Simple Mail Transfer Protocol (SMTP) server sends email notifications.

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

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

Specifies 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>**

Specifies an argument list to a command channel.

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

---

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

Specifies an array of secondary SMS devices.

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

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

Specifies an array of backup SMTP servers. Backup servers use the same configuration as the primary SMTP server.

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

## **-Body<String>**

Specifies the body of a notification message.

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

---

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

### **-BodyAsHtml**

Indicates that the service formats the body of the message as HTML.

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

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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>**

Specifies the content type of an IM. By default, the content type 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential".

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies a description for a notification channel. If you do not set a value, the default is the value of the *DisplayName* parameter.

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

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

Specifies a primary SMS device.

---

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

### **-DeviceEnumerationInterval<TimeSpan>**

Specifies a time interval between SMS notifications. By default, the notification channel waits 10 seconds.

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

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

Specifies a display name for a notification channel. If you do not set a value, the default is the value of the *Name* parameter.

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

---

## **-Encoding<String>**

Specifies the encoding format for a notification message. An SMTP channel uses UTF-8 encoding by default. An IM channel also uses UTF-8 encoding by default. An SMS channel uses 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>**

Specifies the **From** header in a notification e-mail message.

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

## **-Header<Hashtable>**

Specifies headers, as key-value pairs, in a notification email message.

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

---

## **-MaxPrimaryRecipientsPerMail<Int32>**

Specifies the maximum number of recipients for a notification email message.

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

## **-Name<String>**

Specifies the name of a notification channel.

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

## **-NoSubjectEncoding**

Indicates that the cmdlet does not encode the email subject line.

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

---

## **-Port<UInt32>**

Specifies the port number for a channel.

An SMTP channel uses port 25 by default. An IM channel that uses TCP uses port 5060 by default, and an IM channel that uses Transport Layer Security (TLS) uses port 5061 by default.

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

## **-PreferredUserName<String>**

Specifies the preferred user name of an IM channel. If you do not set a value, the default is the value of the *Username* parameter.

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

## **-PrimaryRetryInterval<TimeSpan>**

Specifies a retry interval. The service attempts to switch back to the primary SMTP server or SMS device. By default, the interval is five minutes.

Aliases	none
Required?	false
Position?	named

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

### **-ReplyTo<String>**

Specifies the **Reply-to** header in a notification email message.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCManagementGroupConnection** cmdlet.

Enter a connection object for a management group, such as one that the **Get-SCManagementGroupConnection** cmdlet returns.

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

### **-Server<String>**

Specifies a server that provides SMTP and IM channels for notifications.

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

---

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

### **-SipAuthentication<SipNotificationAuthenticationProtocols>**

Specifies a Session Initiation Protocol (SIP) authentication mechanism (NTLM or Kerberos) for IM channels. By default, SIP authentication uses NTLM authentication.

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

### **-SipProtocol<SipTransportProtocol>**

Specifies a SIP transport protocol (TCP or TLS) for IM channels. By default, SIP uses TCP as a transport protocol.

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

### **-Sms**

Indicates that the cmdlet sends notifications by using SMS.

---

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

### **-Subject<String>**

Specifies a subject line for the notification email message.

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

### **-UserName<Uri>**

Specifies a return address for IM.

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

### **-WorkingDirectory<String>**

Specifies a working directory for a command channel. By default, a command channel uses the %systemdrive% as a working directory.

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: Add a notification channel

This example adds a standard SMTP email channel.

The first command stores the subject line in the variable named \$Subject.

The second command stores the message body in the variable named \$Body.

The third command uses the **Add-SCOMNotificationChannel** cmdlet to add an 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: Add a notification channel for IM

This example adds an IM channel that has a return address of scomadmin.

The first command stores the message body in the variable named \$Body.

The second command uses the **Add-SCOMNotificationChannel** cmdlet to add an IM channel.

```
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: Add a notification channel for SMS

This example adds an SMS channel that has Unicode encoding.

The first command stores the message body in the variable named \$Body.

---

The second command uses the **Add-SCOMNotificationChannel** cmdlet to add an SMS channel.

```
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: Add a command notification channel

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

The first command stores the path to the executable file in the variable named \$Path.

The second command stores the arguments in the variable named \$Arg.

The third command stores the working directory in the variable named \$WorkingDir.

The fourth command uses the **Add-SCOMNotificationChannel** cmdlet to add a command notification channel.

```
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
```

## Related topics

[Get-SCOMNotificationChannel](#)

[Remove-SCOMNotificationChannel](#)

---

# Add-SCOMNotificationSubscriber

---

## Add-SCOMNotificationSubscriber

Adds a notification subscriber in Operations Manager.

### 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** cmdlet adds a notification subscriber in System Center 2012 – Operations Manager.

Notification subscribers receive notifications from Operations Manager.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of notification addresses. Use username@domain.com form for email addresses, SMS:<address> for SMS addresses, and SIP:<address> for IM addresses. Specify the name of a channel for a command channel.

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

### **-DeviceTable<Hashtable>**

Specifies a table of named addresses. Use username@domain.com form for email addresses, SMS:<address> for SMS addresses, and SIP:<address> for IM addresses. Specify the name of a channel for a command channel.

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

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

Specifies the name of a notification subscriber.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Add a notification subscriber

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

```
PS C:\> Add-SCOMNotificationSubscriber -Name "Sarah Jones" -DeviceList
"SarahJones@contoso.com", "sms:2065551212", "sip:SarahJ"
```

### Example 2: Add a notification subscriber with a command channel

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

The first command uses the **Get-SCOMNotificationChannel** cmdlet to get the notification channel that has the display name OnNotify.exe, and stores the result in the \$CommandChannel variable.

The second command adds a notification subscriber by using the *Name* parameter.

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

## Related topics

[Get-SCOMNotificationSubscriber](#)

[Remove-SCOMNotificationSubscriber](#)

---

# Add-SCOMNotificationSubscription

---

## Add-SCOMNotificationSubscription

Adds a 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 notification subscription in System Center 2012 – Operations Manager.

In this release there is no cmdlet support for scoping subscriptions based on a criteria, but you can achieve this result through Windows PowerShell by accessing the SCOM SDK directly.

### Parameters

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

Specifies an array of subscribers to blind carbon copy (BCC) on a message for a subscription.

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

---

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

Specifies an array of subscribers to carbon copy (CC) on a message for a subscription.

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

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

Specifies an array of notification channels. To obtain a notification channel object, use the **Get-SCOMNotificationChannel** cmdlet.

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

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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>**

Specifies the criteria xml that indicates how to filter alerts for this subscription. Criteria can filter on particular rules or monitors, or properties of the alert, but cannot filter on classes or groups.

You can manually copy the criteria xml 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>**

Specifies a delay, in HH:MM:SS, for sending notifications if conditions remain unchanged for this period of time. By default, Operations Manager sends notifications immediately.

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

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

Specifies a description for the notification subscription. If you do not specify a description, the parameter defaults to the value of the *DisplayName* parameter.

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

### **-Disabled**

Indicates that the cmdlet creates a subscription but leaves it in a disabled state.

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

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

Specifies a display name for the subscription. If you do not specify a description, the parameter defaults to the value of the *Name* parameter.

---

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

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

Specifies a name for the subscription.

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

### **-OnlyOnResolutionChange**

Indicates that the notification occurs 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>**

Specifies a polling interval for alerts. If you do not specify this parameter, the interval defaults to 1 minute.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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 an array of subscribers for this subscription. To obtain a notification subscriber object, use the **Get-SCOMNotificationSubscriber** cmdlet.

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

---

## <CommonParameters>

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

## Examples

### Example 1: Add a notification subscription

This example adds a new notification subscription for all alerts.

The first command uses the **Get-SCOMNotificationSubscriber** cmdlet to get subscribers and stores the result in the `$Subscriber` variable.

The second command uses the **Get-SCOMNotificationChannel** cmdlet to get the notification channel, and stores the result in the `$Channel` variable.

The third command uses the **Add-SCOMNotificationSubscription** cmdlet with the *Name*, *Subscriber*, and *Channel* parameters.

```
PS C:\> $Subscriber = Get-SCOMNotificationSubscriber -Name "John Smith","Sarah Jones"
PS C:\> $Channel = Get-SCOMNotificationChannel -DisplayName "EmailChannel"
PS C:\> Add-SCOMNotificationSubscription -Name "NewSubscription1" -Subscriber $Subscriber -
Channel $Channel
```

### Example 2: Add a notification subscription with conditions

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.

The first command uses the **Get-SCOMNotificationSubscriber** cmdlet to get subscribers and stores the result in the `$Subscriber` variable.

The second command uses the **Get-SCOMNotificationChannel** cmdlet to get the notification channel, and stores the result in the `$Channel` variable.

The third command uses the **Add-SCOMNotificationSubscription** cmdlet with the *Name*, *Subscriber*, and *Channel* parameters.

```
PS C:\> $Subscriber = Get-SCOMNotificationSubscriber -Name "John Smith","Sarah Jones"
PS C:\> $Channel = Get-SCOMNotificationChannel -DisplayName "EmailChannel"
PS C:\> Add-SCOMNotificationSubscription -Name "NewSubscription2" -Subscriber $Subscriber -
Channel $Channel -Delay "1:00:00" -Disabled
```

### Example 3: Add a new subscription for instances of a specific class

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.

---

The first command uses the **Get-SCOMNotificationSubscriber** cmdlet to get subscribers and stores the result in the `$Subscriber` variable.

The second command uses the **Get-SCOMNotificationChannel** cmdlet to get the notification channel, and stores the result in the `$Channel` variable.

The third command uses the **Add-SCOMNotificationSubscription** cmdlet with the *Name*, *Subscriber*, and *Channel* parameters. The cmdlet stores the result in the `$Subscription` variable.

The fourth command uses the **Get-SCOMClass** cmdlet and stores the result in the `$HealthService` variable.

The fifth command uses the `Add` method with the ID of the `$HealthService` variable.

The sixth command uses the `Update` method.

```
PS C:\> $Subscriber = Get-SCOMNotificationSubscriber -Name "John Smith","Sarah Jones"
PS C:\> $Channel = Get-SCOMNotificationChannel -DisplayName "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: Add a notification subscription by using the SDK

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

The first command uses the **Get-SCOMNotificationSubscriber** cmdlet to get subscribers and stores the result in the `$Subscriber` variable.

The second command uses the **Get-SCOMNotificationChannel** cmdlet to get the notification channel, and stores the result in the `$Channel` variable.

The third command uses the **Add-SCOMNotificationSubscription** cmdlet with the *Name*, *Subscriber*, and *Channel* parameters. The cmdlet stores the result in the `$Subscription` variable.

The fourth command uses the **Get-SCOMGroup** cmdlet to get the All Windows Computers group, and stores the result in the `$WindowsComputers` variable.

The fifth command uses the `Add` method with the `Id` of the `$WindowsComputers` variable.

The sixth command uses the `Update` method.

```
PS C:\> $Subscriber = Get-SCOMNotificationSubscriber -Name "John Smith","Sarah Jones"
PS C:\> $Channel = Get-SCOMNotificationChannel -DisplayName "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: Add a notification subscription for a specific monitor

This example adds a new notification subscription for all critical alerts raised by the monitor that has the display name ContosoMonitor.

The first command uses the **Get-SCOMNotificationSubscriber** cmdlet to get subscribers and stores the result in the `$Subscriber` variable.

The second command uses the **Get-SCOMNotificationChannel** cmdlet to get the notification channel, and stores the result in the `$Channel` variable.

The third command uses the **Get-SCOMMonitor** cmdlet and stores the result in the `$Monitor` variable.

The fourth command stores the XML criteria in the `$Criteria` variable.

The fifth command uses the **Add-SCOMNotificationSubscription** cmdlet with the *Name*, *Subscriber*, *Channel*, and *Criteria* parameters.

```
PS C:\> $Subscriber = Get-SCOMNotificationSubscriber -Name "John Smith","Sarah Jones"
PS C:\> $Channel = Get-SCOMNotificationChannel -DisplayName "EmailChannel"
PS C:\> $Monitor = Get-SCOMMonitor -DisplayName "ContosoMonitor"
PS C:\> $Criteria = @"
<><And xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <><Expression>
    <><SimpleExpression>
      <><ValueExpression>
        <><Property>ProblemId</Property>
      <></ValueExpression>
      <><Operator>Equal</Operator>
      <><ValueExpression>
        <><Value>$($monitor.Id)</Value>
      <></ValueExpression>
    <></SimpleExpression>
  <></Expression>
  <><Expression>
    <><SimpleExpression>
      <><ValueExpression>
        <><Property>Severity</Property>
      <></ValueExpression>
      <><Operator>Equal</Operator>
      <><ValueExpression>
        <><Value>2</Value>
      <></ValueExpression>
    <></SimpleExpression>
  <></Expression>
<></And>
"@
PS C:\> Add-SCOMNotificationSubscription -Name "Subscription03" -Subscriber $Subscriber -
Channel $Channel -Criteria $Criteria
```

---

## Related topics

[Disable-SCOMNotificationSubscription](#)

[Enable-SCOMNotificationSubscription](#)

[Get-SCOMNotificationSubscription](#)

[Remove-SCOMNotificationSubscription](#)

---

# Add-SCOMRunAsAccount

---

## Add-SCOMRunAsAccount

Adds a Run As account to a 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 a management group. A Run As account enables users to specify the necessary permissions for use with rules, tasks, monitors, and discoveries targeted to specific computers on an as-needed basis.

System Center 2012 – Operations Manager distributes the Run As account credentials to either all agent-managed computers (the less secure option) or only to computers that you specify (the more

---

secure option). By default, all new accounts have the more secure distribution option. To modify the account distribution policy, use the **Set-SCOMRunAsDistribution** cmdlet.

## Parameters

### **-ActionAccount**

Indicates that the account is an action account. An action account specifies credentials that the MonitoringHost management process uses to perform monitoring activities.

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

### **-AuthProtocolAndKey<PSCredential>**

Specifies a **PSCredential** object that includes the Simple Network Management Protocol (SNMP) authentication protocol and key. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet.

If this parameter appears, the cmdlet must also specify the *UserName* and *Passphrase* parameters. Specify the protocol name MD5 or SHA for the *Username* parameter and the key for the *Passphrase* parameter.

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

### **-Basic**

Indicates that the Run As account is a Basic Authentication account, which uses basic web authentication.

---

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

## **-Binary**

Indicates that the Run As account is a Binary Authentication account, which uses authentication that the user defines.

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

## **-CommunityString**

Indicates that the Run As account is a Community String account, which uses community string authentication in Simple Network Management Protocol (SNMP) version 2.

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

---

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

## **-Context<String>**

Specifies the SNMP version 3 context.

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

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false

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

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

Specifies the account description. If this parameter does not appear, the default is the display name.

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

### **-Digest**

Indicates that the Run As account is a Digest Authentication account, which uses standard digest web authentication.

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

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

Specifies the account name.

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

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

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

Specifies the Secure Shell (SSH) key passphrase 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>**

Specifies 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>**

Specifies a **PSCredential** object that stores the SNMP privacy protocol and key. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet.

---

If you specify this parameter appears, you must also specify the *UserName* and *Passphrase* parameters. Specify the protocol name AES or DES for the *Username* parameter, and the key for the *Passphrase* parameter.

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

### **-Privileged**

Indicates that the cross-platform maintenance account has privileged access.

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

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

Specifies the credential for the Run As account.

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

---

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

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

## **-SCXMaintenance**

Indicates that the account is a cross-platform maintenance Run As account.

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

## **-SCXMonitoring**

Indicates that the Run As account is a Basic Authentication account, which uses basic web authentication.

Aliases	none
Required?	false
Position?	named

---

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

### **-Simple**

Indicates that the account is a Simple Authentication Run As account.

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

### **-SnmpV3**

Indicates that the account is an SNMP version 3 Run As account.

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

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

Specifies the account community string.

Aliases	none
Required?	true
Position?	2

---

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

### **-Su**

Indicates that the cross-platform maintenance account uses superuser elevation to perform privileged actions.

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

### **-Sudo**

Indicates that the cross-platform account uses sudo elevation to perform privileged actions. The sudo program enables users to run programs that have the security permissions of another user account.

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

### **-SuPassword<SecureString>**

Specifies the superuser password for a cross-platform maintenance account.

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

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

### **-UserName<String>**

Specifies the user name for the account. This parameter is valid only for SNMP version 3 and cross-platform maintenance accounts. Otherwise, use the *RunAsCredential* parameter.

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

### **-Windows**

Indicates that the account is a Run As account for Windows, which uses Windows credentials for authentication. This is the default account type if the cmdlet does not specify a different type.

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

### **-Confirm**

Prompts you for confirmation before executing the command.

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

## **-WhatIf**

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

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

## **<CommonParameters>**

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

## **Examples**

### **Example 1: Add a Windows Run As account**

This command adds a Run As account that uses Windows authentication.

```
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: Add a Community String Run As account**

This example adds a Run As account that uses Community String authentication.

The first command prompts the user to enter the community string for the account and stores the input as a secure string in the variable named \$CommunityString.

The second account creates the account and specifies the string stored in \$CommunityString as the community string for the account.

---

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

### **Example 3: Add a Basic Authentication Run As account**

This command adds a Run As account that uses basic web authentication.

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

### **Example 4: Add a Simple Authentication Run As account**

This command adds a Run As account that uses simple authentication.

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

### **Example 5: Add a Digest Authentication Run As account**

This command adds a Run As account that uses standard digest web authentication.

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

### **Example 6: Add a Binary Authentication Run As account**

This command adds a Run As account that uses binary authentication.

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

### **Example 7: Add an action account**

This command adds an action account.

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

### **Example 8: Add an SNMP version 3 account without context, authentication, or privacy**

This command adds an SNMP version 3 account that has no context, authentication protocol, or privacy protocol.

```
PS C:\> Add-SCOMRunAsAccount -Snmpv3 -Name "Contoso.Snmp1" -UserName "snmpuser"
```

---

## Example 9: Add an SNMP version 3 account with context, authentication, and privacy

This example adds an SNMP version 3 account that specifies context, authentication protocol, and privacy protocol.

The first command gets the SNMP version 3 privacy protocol and key for the account and assigns them to the variable named \$Auth.

The second command gets the SNMP version 3 authentication protocol and key for the account and assigns them to the variable named \$Privacy.

The third command creates the account, uses the credentials stored in \$Auth for the authentication protocol and key, and uses the credentials stored in \$Privacy for the privacy protocol and key.

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

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

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

## Example 10: Add an SCX monitoring account with sudo elevation

This command adds an SCX monitoring account that uses sudo elevation.

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

## Example 11: Add an SCX maintenance account with privileged access

This example adds an SCX maintenance account that has privileged access and uses a passphrase-protected SSH key.

The first command prompts the user to enter the passphrase and stores the passphrase as a secure string in the variable named \$Passphrase.

The second command creates the account by using the passphrase stored in \$Passphrase.

```
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: Add an SCX maintenance account without privileged access that uses sudo elevation

This command adds an SCX maintenance account that does not have privileged access by specifying a user name and password and sudo elevation.

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

---

## Example 13: Add an SCX maintenance account that uses superuser elevation

This example adds an SCX maintenance account that does not have privileged access by specifying a user name and password and a superuser account for elevation.

The first command prompts the user to enter the password, converts the user input to a secure string, and stores the password in the \$SuPassword variable.

The second command creates the account by specifying the password that is stored in \$SuPassword as the superuser password.

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

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMRunAsAccount](#)

[New-SCOMRunAsAccount](#)

[Remove-SCOMRunAsAccount](#)

[Update-SCOMRunAsAccount](#)

---

# Add-SCOMRunAsProfile

---

## Add-SCOMRunAsProfile

Adds a Run As 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 Run As profile. A Run As profile is a group of associated Run As accounts that help manage credentials and their distribution to different computers.

### Parameters

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

Specifies 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 an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. Valid formats include a NetBIOS name, an IP address, or a

---

fully qualified domain name (FQDN). To specify the local computer, specify the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type "Get-Help Get-Credential".

This account must have access to the server that the *ComputerName* parameter specifies, if that parameter appears.

If you do not specify this parameter, the default is the account for the current user.

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

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

Specifies a description of the Run As profile. If this parameter is not specified, the default description is the description specified in the *DisplayName* parameter.

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 Run As profile. If this parameter is not specified, the default display name is the name specified in the *Name* parameter.

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

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

Specifies a globally unique identifier (GUID) to identify the Run As profile. If this parameter is not specified, the cmdlet generates a new GUID.

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

---

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

Specifies an array of **ManagementPack** objects. The cmdlet saves the RunAs profile in these management packs. To obtain a **ManagementPack** object, use the **Get-SCOMManagementPack** 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 the Run As profile.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type "Get-Help about\_OpsMgr\_Connections".

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: Add a Run As profile to a management pack**

This example adds a Run As profile to a management pack.

---

The first command gets the management packs that have names that end with DefaultUser and stores those management packs in the variable named \$Mp.

The second command creates the Run As profile named Contoso.MonitoringProfile and adds the management packs stored in \$Mp to it.

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

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

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMRunAsProfile](#)

[Remove-SCOMRunAsProfile](#)

[Set-SCOMRunAsProfile](#)

---

# Add-SCOMSubscriberSchedule

---

## Add-SCOMSubscriberSchedule

Adds a new schedule entry for a notification subscriber.

### 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 schedule entry for a notification subscriber. Notification subscribers are users who receive notifications when System Center 2012 – Operations Manager raises an alert on a monitored system.

If you do not add entries to a subscriber's existing schedule, the subscriber continues to receive notifications.

### Parameters

#### -AllDay

Indicates that the schedule entry applies to the whole day.

Aliases	none
Required?	true
Position?	named
Default Value	none

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

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, specify the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential".

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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>**

Specifies the date at which the schedule entry ends. If this command does not specify *StartDate* and *EndDate* parameters, the schedule entry applies to all dates.

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

## **-EndTime<DateTime>**

Specifies the time at which the schedule entry ends.

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

---

## -Exclude

Indicates that the command excludes the specified times and dates from the schedule. If this parameter does not appear, the schedule includes only the specified times.

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

## -PassThru

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

## -SCSession<Connection[]>

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type "Get-Help about\_OpsMgr\_Connections".

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

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

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

Specifies the date at which the schedule entry starts. If this command does not specify *StartDate* and *EndDate* parameters, the schedule entry applies to all dates.

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

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

Specifies the start time for the schedule entry.

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

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

Specifies the notification subscriber. To obtain a notification subscriber object, use the **Get-SCOMNotificationSubscriber** cmdlet, or create a new object by using the **Add-SCOMNotificationSubscriber** cmdlet.

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

## **-TimeZone<String>**

Specifies the time zone for the schedule entry. If you specify wildcards as part of the time zone, the cmdlet performs wildcard matching against the display names for time zones. If this parameter does not appear, the cmdlet defaults to the current user time zone.

Examples of valid values for this parameter 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: Add a schedule for a notification subscriber**

This command adds a schedule window from 9 A.M. to 5 P.M. on Mondays, Wednesdays, and Fridays in all date ranges for a notification subscriber named Katarina. The command uses the **Get-**

---

**SCOMNotificationSubscriber** cmdlet to get the specified subscriber and passes that subscriber to the **Add-SCOMSubscriberSchedule** cmdlet by using the pipeline operator.

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

## Example 2: Add multiple schedule entries for a notification subscriber

This command adds two schedule entries to the notification subscriber named Cesar in the USA Central time zone. The command uses the **Get-SCOMNotificationSubscriber** cmdlet to get the specified subscriber and passes that subscriber to the **Add-SCOMSubscriberSchedule** cmdlet by using the pipeline operator. The command adds the entry with the specified values. The command specifies the *PassThru* parameter in order to pass the user to another instance of the **Add-SCOMSubscriberSchedule** cmdlet by using the pipeline operator. The command adds a second schedule entry with the specified values.

```
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
```

## Related topics

[Add-SCOMNotificationSubscriber](#)

[Clear-SCOMSubscriberSchedule](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMNotificationSubscriber](#)

[Remove-SCOMSubscriberSchedule](#)

---

# 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 System Center 2012 – Operations Manager management group tier. Connectors communicate monitoring information between systems in a tiered management group.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, specify the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

---

## **-Connector<MonitoringConnector>**

Specifies the System Center 2012 – 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

## **-PassThru**

Indicates that the cmdlet creates an object. This parameter allows you to use this cmdlet in a 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 an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

### **-Tier<TieredManagementGroup>**

Specifies the tiered management group in Operations Manager.

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: Add a connector to a tiered management group

This example adds a connector to an existing tiered management group.

The first command gets tiered management groups from the server, and passes them to the **Select-Object** cmdlet by using the pipeline operator. That cmdlet selects the first tiered management group. For more information, type `Get-Help Select-Object`. The command stores the first tiered management group in the \$Tier variable.

The second command adds a connector named MyProductConnector. The command then passes output to the **Add-SCOMTierConnector** cmdlet by using the pipeline operator. That cmdlet adds the connector to the tiered management group in the \$Tier variable.

```
PS C:\> $Tier = Get-SCOMTieredManagementGroup | Select-Object -First 1
PS C:\> Add-SCOMConnector -Name MyProductConnector | Add-SCOMTierConnector -Tier $Tier
```

---

## Related topics

[Add-SCOMTierConnector](#)

[Add-SCOMTieredManagementGroup](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMTierConnector](#)

[Get-SCOMTieredManagementGroup](#)

[Remove-SCOMTieredManagementGroup](#)

---

# Add-SCOMTieredManagementGroup

---

## Add-SCOMTieredManagementGroup

Adds a tiered management group to a 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 a management group. A tiered management group is part of a connected management group that has peer-to-peer connections between its members and that shares member data in a single System Center 2012 – Operations Manager console.

### Parameters

#### -AvailableForConnectors

Indicates that the new tier is available for connectors.

Aliases	none
Required?	false
Position?	named

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

### **-CacheConfiguration<CacheConfiguration>**

Specifies the cache configuration for the tiered management group.

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

### **-CacheMode<CacheMode>**

Specifies the cache mode 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 an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

### **-ConnectionCredential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about the **Get-Credential** cmdlet, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-InactivityTimeout<TimeSpan>**

Specifies the inactivity timeout for the tiered management group.

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

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

Specifies 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>**

Specifies the Run As credential that a management group uses 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 an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

### **-SendReceiveTimeout<TimeSpan>**

Specifies the send/receive timeout for the tiered management group as a **TimeSpan** object.

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

### **-ServerName<String>**

Specifies the name of the SDK server with which to connect in the new 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: Add a tiered management group

This command adds a tiered management group named New Tier to the server named SCOM02.contoso.com.

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

### Example 2: Add a tiered management group that is available to connectors

This command adds a new tiered management group named New Tier for Connectors to the server named SCOM02.contoso.com. This tiered management group is available to connectors and uses the Run As account named TierAccount.

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

## Related topics

[Get-SCOMManagementGroupConnection](#)

[New-SCOMManagementGroupConnection](#)

[Set-SCOMManagementGroupConnection](#)

---

# Add-SCOMUserRole

---

## Add-SCOMUserRole

Adds a user role to a 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 user role to a management group. A user role consists of two features:

- Profile. Defines the collection of operations to which the user role has access.
- Scope. Defines the boundaries for profile operations; for example, tasks and groups.

---

## Parameters

### -AdvancedOperator

Indicates that the new role is an Advanced Operator. This role grants members the ability to override the configuration of rules and monitors for specific targets or groups of targets within the configured scope. The Advanced Operator role also grants all of the permissions that the Operator and Read-Only Operator profiles grant.

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. This role grants members the ability to create, edit, and delete monitoring configuration that includes tasks, rules, monitors, and views, within the configured scope. For convenience, an Author role can have permissions for specific groups. The Author role also grants all of the permissions in the Advanced Operator, Operator, and Read-Only Operator roles.

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

### -ClassScope<ManagementPackClass[]>

Specifies an array of **ManagementPackClass** objects that represent the classes to which an Author role has access. To obtain a **ManagementPackClass** object, use the **Get-SCOMClass** cmdlet.

If you do not specify a value for this parameter, the role has access to all classes. To deny the role access to all classes, 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 an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

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

Specifies a description of the user role. If you do not specify a value for this parameter, the cmdlet uses the value of the *DisplayName* parameter.

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. If this parameter does not appear, the default is the value in the *Name* parameter.

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

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

Specifies an array of **MonitoringObjectGroup** objects that represent the groups to which a user role has access. To obtain a monitoring object group object, use the **Get-MonitoringObjectGroup** cmdlet.

If you do not specify a value for this parameter, the role has access to all groups. To deny the role access to all groups, 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. This role grants members the ability to interact with alerts, run tasks, and access views according to their configured scope. The Operator role also grants all of the permissions that the Read-Only Operator profile grants.

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. This role grants members the ability to view alerts and access views according to their configured scope.

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

## -SCSession<Connection[]>

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

## -TaskScope<ManagementPackTask[]>

Specifies an array of **ManagementPackTask** objects that represent the tasks to which an Author, Operator, or Advanced Operator role has access. To obtain a **ManagementPackTask** object, use the **ManagementPack.GetTask** cmdlet. If this parameter does not appear, the role has access to all tasks. To deny the Author, Operator, or Advanced Operator role access to all tasks, 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[]>**

Specifies an array that contains the names of users who are part of the 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: Add a new Read Only Operator user role

This command adds a Read Only Operator user role named Script Role.

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

### Example 2: Add a new user role by specifying user names, tasks, and groups

This example adds a user role based on user names, tasks, and groups.

The first two commands get all task and group objects that have SQL in their names and store the objects in the \$ApprovedTasks and \$ApprovedGroups variables, respectively.

The last command creates an Operator user role named SQL Operator, which contains the users Katarina and Cesar. The user role gives these two users access to the tasks and groups that are stored in \$ApprovedTasks and \$ApprovedGroups.

```
PS C:\> $ApprovedTasks = Get-SCOMTask -Name '*SQL*'
PS C:\> $ApprovedGroups = Get-SCOMGroup -DisplayName '*SQL*'
PS C:\> Add-SCOMUserRole -Name 'SQL Operator' -Operator -GroupScope $ApprovedGroups -
TaskScope $ApprovedTasks -User 'Contoso\Katarina','Contoso\Cesar'
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMUserRole](#)

[Set-SCOMUserRole](#)

---

# Approve-SCOMPendingManagement

---

## Approve-SCOMPendingManagement

Approves pending agent management actions.

### Syntax

Parameter Set: FromAgentPendingAction

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

### Detailed Description

The **Approve-SCOMPendingManagement** cmdlet approves pending management actions in System Center 2012 – Operations Manager.

### Parameters

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

Specifies the credentials for a pending action. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type " Get-Help Get-Credential".

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

#### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

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

Specifies an array of pending actions to approve. 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, pending administrator approval.

## **Examples**

### **Example 1: Retrieve pending management entries**

This command retrieves the agent management entries that are pending with an action of ManualApproval. By using the *WhatIf* parameter, the cmdlet displays the actions that occur if the command was implemented.

---

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

## Related topics

[Deny-SCOMPendingManagement](#)

[Get-SCOMPendingManagement](#)

---

# Clear-SCOMSubscriberSchedule

---

## Clear-SCOMSubscriberSchedule

Removes all entries from 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 from a notification subscriber's schedule. Notification subscribers are users who receive notifications when System Center 2012 – Operations Manager raises an alert on a monitored system. A clear schedule indicates an always on subscription that can notify at any time.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-**Help** Get-Credential".

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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 an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type "Get-**Help** about\_OpsMgr\_Connections".

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

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

Specifies the notification subscriber. To get a notification subscriber object, use the **Get-SCOMNotificationSubscriber** cmdlet, or create a new object by using the **Add-SCOMNotificationSubscriber** 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: Clear all subscriber schedules

This command resets the schedules of all subscribers. The command uses the **Get-SCOMNotificationSubscriber** cmdlet to get all notification subscribers and then passes them to the **Clear-SCOMSubscriberSchedule** cmdlet by using the pipeline operator.

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

## Related topics

[Add-SCOMNotificationSubscriber](#)

[Add-SCOMSubscriberSchedule](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMNotificationSubscriber](#)

[Remove-SCOMSubscriberSchedule](#)

---

# Deny-SCOMPendingManagement

---

## Deny-SCOMPendingManagement

Denies pending agent management actions.

### Syntax

Parameter Set: FromAgentPendingAction

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

### Detailed Description

The **Deny-SCOMPendingManagement** cmdlet denies pending management actions in System Center 2012 – Operations Manager.

### Parameters

#### -PassThru

Indicates that the cmdlet creates or modifies an object that a command can use in 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

#### -PendingAction<AgentPendingAction[]>

Specifies an array of pending actions to deny. 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: Deny pending management entries

This command retrieves a list of agent management entries that are pending with an action of ManualApproval, and passes the output to the **Deny-SCOMPendingManagement** cmdlet by using the pipeline operator. By using the *WhatIf* parameter, the cmdlet 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 agent for other computers.

### Syntax

Parameter Set: FromAgent

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

### Detailed Description

The **Disable-SCOMAgentProxy** cmdlet disables System Center 2012 – Operations Manager agents from acting as a proxy agents and discovering managed objects on other computers.

### Parameters

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

Specifies an array of **AgentManagedComputer** objects. This parameter specifies the Operations Manager agents to disable from acting as proxy agents. To obtain an **AgentManagedComputer** object, use the **Get-SCOMAgent** cmdlet.

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

#### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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: Disable a proxy agent

This command gets the Operations Manager agent named server01.contoso.com and disables it from acting as a proxy agent.

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

### Example 2: Disable agents that act as a proxy agent

This command gets all Operations Manager agents that have the **ProxyingEnabled** property set to \$True, and then disables the agents from acting as a proxy agent.

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

## Related topics

[Enable-SCOMAgentProxy](#)

[Get-SCOMAgent](#)

[Get-SCOMAgentlessManagedComputer](#)

[Get-SCOMAgent](#)

---

# Disable-SCOMDiscovery

---

## Disable-SCOMDiscovery

Disables Operations Manager 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 System Center 2012 – Operations Manager discoveries. The cmdlet creates and saves overrides for specified discoveries that disable those discoveries.

Specify groups, instances, or classes, along with the discoveries to disable. The cmdlet saves overrides to a specified unsealed management pack.

### Parameters

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

Specifies an array of management pack class objects. To obtain a class object, use the **Get-SCOMClass** cmdlet. The cmdlet disables discoveries for these classes.

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

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

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

Specifies an array of **ManagementPackDiscovery** objects. To obtain a **ManagementPackDiscovery** object, use the **Get-SCOMDiscovery** cmdlet.

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

### **-Enforce**

Indicates that the cmdlet sets the **Enforce** property to \$True on the override.

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

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

Specifies an array of monitoring objects that represent groups. To obtain a group, use the **Get-SCOMGroup** cmdlet. The cmdlet disables discoveries for these groups.

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

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

Specifies an array of monitoring objects that represent instances. To obtain instances, use the **Get-SCOMClassInstance** cmdlet. The cmdlet disables discoveries for these instances.

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. To obtain a management pack object, use the **Get-SCOMManagementPack** cmdlet. The cmdlet saves overrides into the specified management pack.

If the discovery is in an unsealed management pack, you must save the override into the same management pack.

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

---

## -PassThru

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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: Disable discoveries for a class

This example disables discoveries for a specified class. The **Disable-SCOMDiscovery** cmdlet saves an override in a specified management pack.

The first command uses the **Get-SCOMManagementPack** cmdlet to get management pack objects that have the specified display name, and passes them to the **Where-Object** cmdlet by using the pipeline operator. That cmdlet drops any sealed management packs. For more information, type `Get-Help Where-Object`. The command stores all unsealed management packs in the `$MP` variable.

The second command uses the **Get-SCOMClass** cmdlet to get classes that have the specified display name, and then stores them in the `$Class` variable.

The third command uses the **Get-SCOMDiscovery** cmdlet to get discovery objects that have display names that contain the string rule, and then stores them in the `$Discovery` variable.

The fourth command disables the discoveries represented by the objects stored in the `$Discovery` variable. The command specifies the class object stored in the `$Class` variable. The cmdlet saves the override in the management pack represented by the object in the `$MP` variable. The command uses the *Enforce* parameter; therefore the cmdlet sets the **Enforce** property to `$True`.

```
PS C:\> $MP = Get-SCOMManagementPack -displayname "My SQL MP Customization" | Where-Object
{$_ .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 in Operations Manager.

### 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 in System Center 2012 – Operations Manager. In Operations Manager, monitors define logic for determining the health of an object. The cmdlet disables monitors by creating and saving overrides that modify the default behavior of a monitor.

### Parameters

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

Specifies an array of class objects. 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 cmdlet sets the **Enforce** property on the override to \$True.

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

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

Specifies an array of one or more group objects. 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 an array of class instance objects. The *Instance* parameter also accepts group objects as input. 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 where you can save the override. If the monitor is in an unsealed management pack, you must save the overrides into the same management pack. 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 an array of one or more monitor objects. 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

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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: Disable a monitor

This example disables an Operations Manager monitor.

The first command uses the **Get-SCOMManagementPack** cmdlet to get a management pack object, and it stores the result in the variable named \$MP.

The second command uses the **Get-SCOMClass** cmdlet to get a class object, and it stores the result in the variable named \$Class.

The third command uses the **Get-SCOMMonitor** cmdlet to get a monitor object, and it stores the result in the variable named \$Monitor.

The fourth command uses the **Disable-SCOMMonitor** cmdlet to disable the monitor by using the *Enforce* parameter. The cmdlet stores the override in the variable named \$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

Disables a notification subscription.

### Syntax

Parameter Set: Default

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

### Detailed Description

The **Disable-SCOMNotificationSubscription** cmdlet disables a notification in System Center 2012 – Operations Manager.

### Parameters

#### -PassThru

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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 an array of subscriptions to disable.

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: Disable all enabled notifications

This command disables all enabled notifications.

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

### Example 2: Disable notifications to a notification subscriber

This command disables all notifications where the TO line of the message includes the user account CONTOSO\SarahJones.

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

## Related topics

[Add-SCOMNotificationSubscription](#)

[Enable-SCOMNotificationSubscription](#)

[Get-SCOMNotificationSubscription](#)

[Remove-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 in System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Disable data reporting**

This command disables operational data reporting.

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

## **Related topics**

[Enable-SCOMOperationalDataReporting](#)

[Test-SCOMOperationalDataReporting](#)

---

# Disable-SCOMRule

---

## Disable-SCOMRule

Creates and saves overrides that disable monitoring 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 creates and saves overrides that disable System Center 2012 – Operations Manager monitoring rules. After you disable a monitoring rule, Operations Manager no longer raises alerts for the systems specified in the rules.

## Parameters

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

Specifies an array of management pack objects that represent classes for which the cmdlet disables rules. To obtain a **ManagementPackClass** object, use the **Get-SCOMClass** cmdlet.

Aliases	none
Required?	true

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

## **-Enforce**

Indicates that Operations Manager enforces the override that disables the monitoring rules.

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

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

Specifies an array of monitoring objects that represent groups. To obtain a group object, use the **Get-SCOMGroup** cmdlet. The cmdlet disables rules for these groups.

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

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

Specifies an array of monitoring objects that represent instances. To obtain instances, use the **Get-SCOMClassInstance** cmdlet. The cmdlet disables rules for these instances. This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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

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

Specifies a management pack object that stores overrides. To obtain a management pack object, use the **Get-SCOMManagementPack** cmdlet. If the rule is in an unsealed management pack, you must save the override into the same management pack.

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

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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 an array of rules as **ManagementPackRule** objects. To obtain a **ManagementPackRule** object, use the **Get-SCOMRule** cmdlet.

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: Disable a monitoring rule

This example disables a monitoring rule for a management pack.

The first three commands get an unsealed 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 class object stored in \$Class. The command stores the override in the management pack stored in \$MP. The command specifies the *Enforce* parameter.

```
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

[Enable-SCOMRule](#)

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementPack](#)

[Get-SCOMRule](#)

---

# Enable-SCOMAgentProxy

---

## Enable-SCOMAgentProxy

Enables agents to act as proxy agents for other computers.

### Syntax

Parameter Set: FromAgent

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

### Detailed Description

The **Enable-SCOMAgentProxy** cmdlet enables one or more agents to act as a proxy agents and discover managed objects on other computers.

### Parameters

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

Specifies an array of **AgentManagedComputer** objects. This parameter specifies the Operations Manager agents to enable to act as proxy agents. To obtain an **AgentManagedComputer** object, use the **Get-SCOMADAgent** cmdlet.

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

#### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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: Enable an agent to act as a proxy agent

This command gets the Operations Manager agent named server01.contoso.com and enables the agent to act as a proxy agent for other agents and agentless managed computers. The command passes the name of the agent to the *DNSHostName* parameter of the **Get-SCOMAgent** cmdlet.

```
PS C:\> "server01.contoso.com" | Get-SCOMAgent | Enable-SCOMAgentProxy -PassThru
```

### Example 2: Enable agents to act as a proxy agent

This command gets all Operations Manager agents that have the **ProxyingEnabled** property set to `$False`, and then enables the agents to act as a proxy after the user confirms the action.

```
PS C:\> Get-SCOMAgent | Where-Object {$_.ProxyingEnabled.Value -eq $False} | Enable-SCOMAgentProxy -Confirm
```

### Example 3: Enable an agent to act as a proxy agent by using a property value

This example enables an agent to act as an agent proxy by setting a property of the agent.

The first command gets the Operations Manager agent named server01.contoso.com, and stores it in the `$Agent` variable.

The second command enables the agent stored in the `$Agent` variable to act as a proxy agent for other computers. The command sets the **ProxyingEnabled** property to `$True` for the agent stored in the `$Agent` variable.

```
PS C:\> $Agent = Get-SCOMAgent -DNSHostName "server01.contoso.com"  
PS C:\> $Agent.ProxyingEnabled
```

## Related topics

[Disable-SCOMAgentProxy](#)

[Get-SCOMAgent](#)

---

# Enable-SCOMDiscovery

---

## Enable-SCOMDiscovery

Enables Operations Manager 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-SCOMDiscovery** cmdlet enables System Center 2012 – Operations Manager discoveries. The cmdlet creates and saves overrides for specified discoveries that enable those discoveries.

## Parameters

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

Specifies an array of management pack class objects. To obtain a class object, use the **Get-SCOMClass** cmdlet. The cmdlet enables discoveries for these classes.

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

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

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

Specifies an array of **ManagementPackDiscovery** objects. To obtain a **ManagementPackDiscovery** object, use the **Get-SCOMDiscovery** cmdlet.

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

### **-Enforce**

Indicates that the cmdlet sets the **Enforce** property to \$True on the override.

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

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

Specifies an array of monitoring objects that represent groups. To obtain a group, use the **Get-SCOMGroup** cmdlet. The cmdlet enables discoveries for these groups.

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

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

Specifies an array of monitoring objects that represent instances. To obtain instances, use the **Get-SCOMClassInstance** cmdlet. The cmdlet enables discoveries for these instances.

This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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. To obtain a management pack object, use the **Get-SCOMManagementPack** cmdlet. The cmdlet saves the override into the specified management pack.

If the discovery is in an unsealed management pack, you must save the override into the same management pack.

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

---

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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: Enable discoveries for a class

This example enables discoveries for a specified class. The **Enable-SCOMDiscovery** cmdlet saves an override in a specified management pack.

The first command uses the **Get-SCOMManagementPack** cmdlet to get management pack objects that have the specified display name, and passes them to the **Where-Object** cmdlet by using the pipeline operator. That cmdlet drops any sealed management packs. For more information, type `Get-Help Where-Object`. The command stores all unsealed management packs in the `$MP` variable.

The second command uses the **Get-SCOMClass** cmdlet to get classes that have the specified display name, and then stores them in the `$Class` variable.

The third command uses the **Get-SCOMDiscovery** cmdlet to get discovery objects that have display names that contain the string rule, and then stores them in the `$Discovery` variable.

The fourth command enables the discoveries. The `$Discovery` variable contains objects that represented discoveries. The command specifies the class object stored in the `$Class` variable. The cmdlet saves the override in the management pack represented by the object in the `$MP` variable. The command uses the *Enforce* parameter; therefore the cmdlet sets the **Enforce** property to `$True`.

```
PS C:\> $MP = Get-SCOMManagementPack -DisplayName "My SQL MP Customization" | Where-Object
{$_ .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 -ManagementPack $MP -Discovery $Discovery -
Enforce
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMDiscovery](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementPack](#)

[Disable-SCOMDiscovery](#)

---

# Enable-SCOMMonitor

---

## Enable-SCOMMonitor

Enables monitors in Operations Manager.

### 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 in System Center 2012 – Operations Manager. In Operations Manager, monitors define logic for determining the health of an object.

## Parameters

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

Specifies an array of class objects. 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 cmdlet sets the **Enforce** property on the override to \$True.

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

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

Specifies an array of group objects. 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 an array of class instance objects. The *Instance* parameter also accepts group objects as input. 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 where you can save the override. If the monitor is in an unsealed management pack, you must save the overrides into the same management pack. For information about how to get a management pack object, type "Get-He1p Get-SCOMManagementPack".

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

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

Specifies an array of monitor objects. For information about how to get a monitor object, type "Get-He1p Get-SCOMMonitor".

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

---

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in a pipeline. 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: Enable a monitor

This example enables an Operations Manager monitor.

The first command uses the **Get-SCOMManagementPack** cmdlet to get a management pack object, and it stores the result in the variable named \$MP.

The second command uses the **Get-SCOMClass** cmdlet to get a class object, and it stores the result in the variable named \$Class.

The third command uses the **Get-SCOMMonitor** cmdlet to get a monitor object, and it stores the result in the variable named \$Monitor.

The fourth command uses the **Enable-SCOMMonitor** cmdlet to enable the monitor by using the *Enforce* parameter.

```
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 in System Center 2012 – Operations Manager.

### Parameters

#### -PassThru

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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 an array of subscriptions to enable. To obtain a notification subscription object, use the **Get-SCOMNotificationSubscription** cmdlet.

---

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

## <CommonParameters>

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

## Examples

### Example 1: Enable all notification subscriptions

This command enables all disabled notifications.

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

### Example 2: Enable notification subscriptions for a specific user

This command enables all notifications where the TO line includes a specific user.

```
PS C:\> Get-SCOMNotificationSubscription | where-object{$_ .ToRecipients -contains "Contoso\SarahJones"} | Enable-SCOMNotificationSubscription
```

## Related topics

[Add-SCOMNotificationSubscription](#)

[Disable-SCOMNotificationSubscription](#)

[Get-SCOMNotificationSubscription](#)

[Remove-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 for the management group in System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Enable data reporting**

This command enables operational data reporting.

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

## **Related topics**

[Disable-SCOMOperationalDataReporting](#)

[Test-SCOMOperationalDataReporting](#)

---

# Enable-SCOMRule

---

## Enable-SCOMRule

Creates and saves overrides that enable monitoring 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** cmdlet creates and saves overrides that enable System Center 2012 – Operations Manager monitoring rules.

## Parameters

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

Specifies an array of management pack objects that represent classes for which the cmdlet disables rules. To obtain a **ManagementPackClass** object, use the **Get-SCOMClass** cmdlet.

Aliases	none
Required?	true
Position?	1

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

## **-Enforce**

Indicates that Operations Manager enforces the override that enables the monitoring rules.

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

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

Specifies an array of monitoring objects that represent groups. To obtain a group object, use the **Get-SCOMGroup** cmdlet. The cmdlet enables rules for these groups.

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

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

Specifies an array of monitoring objects that represent instances. To obtain instances, use the **Get-SCOMClassInstance** cmdlet. The cmdlet enable rules for these instances. This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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

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

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

Specifies a management pack object that stores overrides. To obtain a management pack object, use the **Get-SCOMManagementPack** cmdlet. If the rule is in an unsealed management pack, you must save the override into the same management pack.

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

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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 an array of rules as **ManagementPackRule** objects. To obtain a **ManagementPackRule** object, use the **Get-SCOMRule** cmdlet.

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: Enable a monitoring rule for a management pack

This example enables a monitoring rule for a management pack.

The first three commands get an unsealed 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 monitoring rule stored in the \$Rule variable for the class stored in the \$Class variable. The command stores the override in the management pack stored in the \$MP variable. The *Enforce* parameter specifies that Operations Manager enforces the override that enables the monitoring rules.

```
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

[Disable-SCOMRule](#)

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementPack](#)

[Get-SCOMRule](#)

---

# Exit-SCOMCEIP

---

## Exit-SCOMCEIP

Removes the local computer from Operations Manager 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 Microsoft Customer Experience Improvement Program (CEIP) data collection in System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Remove a computer from CEIP participation**

This command removes the local computer from participating in Operations Manager CEIP data collection.

```
PS C:\> Exit-SCOMCEIP
```

## **Related topics**

[Join-SCOMCEIP](#)

[Test-SCOMCEIP](#)

---

# Export-SCOMEffectiveMonitoringConfiguration

---

## Export-SCOMEffectiveMonitoringConfiguration

Exports configuration applicable to a monitoring object.

### Syntax

Parameter Set: Empty

```
Export-SCOMEffectiveMonitoringConfiguration [-Instance] <MonitoringObject> [-Path] <String>
[-ComputerName <String[]> ] [-Credential <PSCredential> ] [-Encoding <Encoding> ] [-
RecurseContainedObjects] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Export-SCOMEffectiveMonitoringConfiguration** 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 saves the results to a .csv file. This file uses the pipe symbol (|) as a separator.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 an array of monitoring objects that represent instances. To obtain a class instance object, use the **Get-SCOMClassInstance** cmdlet.

This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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

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

Specifies the path and file name for the exported .csv file. The cmdlet does not add a file name extension.

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

### **-RecurseContainedObjects**

Indicates that the cmdlet exports data for objects included in the monitoring object that the *Instance* parameter specifies. For example, in the case of a computer, the cmdlet exports 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Export monitoring configuration for a group of computers

This example exports the monitoring configuration for all the computers in a specified group of computers.

The first command gets the group members of the All Windows Computers group and stores the objects in the \$Members variable.

The second command uses the pipeline operator to pass each object stored in the \$Members variable to the **ForEach-Object** cmdlet, which includes the **Export-SCOMEffektivMonitoringConfiguration** command that exports the data to a .csv file in the specified location. The command includes the name of the computer in the file name, and appends a .csv extension. For more information about **ForEach-Object**, type Get-Help ForEach-Object.

```
PS C:\> $Members = (Get-SCOMGroup -DisplayName "All Windows  
Computers").GetRelatedMonitoringObjects()  
PS C:\> $Members | ForEach-Object { Export-SCOMEffektivMonitoringConfiguration -Instance  
$_ -Path "C:\temp\$($_.DisplayName).csv" } -RecurseContainedObjects }
```

---

## **Related topics**

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementGroupConnection](#)

---

# Export-SCOMManagementPack

---

## Export-SCOMManagementPack

Exports a management pack.

### 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 System Center 2012 – Operations Manager. You can use this cmdlet to save or archive management pack information.

### Parameters

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

Specifies an array of **ManagementPack** objects. To obtain a **ManagementPack** object, use the **Get-SCOMManagementPack** cmdlet.

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

#### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

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

Specifies the folder for the exported management pack files. 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 this 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, this cmdlet returns a **ManagementPack** object.

## Examples

### Example 1: Export management packs by using names

This command uses the **Get-SCOMManagementPack** cmdlet to get all management packs that have a name that contains the string snmp and then uses the pipeline operator to pass them to the **Export-SCOMManagementPack** cmdlet. The command exports management packs to the C:\MPArchive directory.

```
PS C:\> Get-SCOMManagementPack -Name *snmp* | Export-SCOMManagementPack -Path "C:\MPArchive"
```

## Related topics

[Import-SCOMManagementPack](#)

[Get-SCOMManagementPack](#)

[Remove-SCOMManagementPack](#)

---

# Get-SCOMAccessLicense

---

## Get-SCOMAccessLicense

Gets information about licenses for 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

The **Get-SCOMAccessLicense** cmdlet gets a list of System Center 2012 – Operations Manager license types or a list of computers and their license information. Use this cmdlet as part of a central management system that correlates and deduplicates the license information with Access License data from other System Center 2012 products.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of license names.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCManagementGroupConnection** cmdlet.

---

A connection object represents a connection to a management server. The default is the current management group connection.

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

## **-ShowLicense**

Indicates that the cmdlet returns 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: Get all licenses on the local computer**

This command gets information about all licenses for System Center 2012 – Operations Manager and Windows for the local computer.

```
PS C:\> Get-SCOMAccessLicense
```

---

## Example 2: Get all licenses for Operations Manager

This command gets all the licenses for System Center 2012 – Operations Manager for the local computer.

```
PS C:\> Get-SCOMAccessLicense -LicenseName
```

## Example 3: Get licenses by using a name

This command gets all licenses that have the name System Center Operations Manager 2012 Management Server.

```
PS C:\> Get-SCOMAccessLicense -LicenseName "System Center Operations Manager 2012 Management Server"
```

## Example 4: Get licenses by using a name

This command gets all licenses that have the name SML.

```
PS C:\> Get-SCOMAccessLicense -LicenseName "SML"
```

## Related topics

[Get-SCOMLicense](#)

[Set-SCOMLicense](#)

---

# Get-SCOMADAgentAssignment

---

## Get-SCOMADAgentAssignment

Gets AD DS 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 Domain Services (AD DS) agent assignments for the management group.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies the name of the domain or domain controller in which the target agents reside.

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

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

Specifies an array of **ManagementServer** objects. This parameter specifies the primary management servers for the target agent-managed computer. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

## **<CommonParameters>**

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

## **Examples**

### **Example 1: Get AD DS agent assignments by using a domain name**

This command gets all AD DS agent assignments for the domain named cdomain01.contoso.com.

```
PS C:\> Get-SCOMADAgentAssignment -Domain "cdomain01.contoso.com"
```

### **Example 2: Get AD DS agent assignments by using a primary server**

This command gets AD DS agent assignments for a domain that have the same primary management server. The command uses the **Get-SCOMManagementServer** cmdlet to get the management server

---

named OMServer01, and passes the result to the **Get-SCOMADAgentAssignment** cmdlet by using the pipeline operator. The command cmdlet gets all AD DS agent assignments for the domain named contoso.com that have the primary server named OMServer01.

```
PS C:\> Get-SCOMManagementServer "OMServer01*" | Get-SCOMADAgentAssignment -Domain "contoso.com"
```

## Related topics

[Add-SCOMADAgentAssignment](#)

[Update-SCOMADAgentAssignment](#)

[Remove-SCOMADAgentAssignment](#)

---

# 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. You can specify the *DNSHostName* parameter to get the agent-managed computers on a Domain Name System (DNS) host, or you can specify the *ManagementServer* parameter to get all the child agent-managed computers for a primary management server.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false

Position?	named
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 of the agent-managed computer.

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

---

## **-ManagementServer<ManagementServer>**

Specifies a **ManagementServer** object. This parameter specifies the primary management server from which to retrieve all child agents. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -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: Get all agents in the management group

This command establishes a temporary connection with the computer named Server01 and gets all agents in the management group.

```
PS C:\> Get-SCOMAgent -ComputerName "Server01.Contoso.Com"
```

### Example 2: Get the agents managed by a management server

This example gets agent-managed computers that are managed by a management server.

The first command gets the management server object named MgmtServer01.Contoso.com and stores the object in the \$MgmtServer variable.

The second command gets the agents that are managed by the management server stored in \$MgmtServer.

```
PS C:\> $MgmtServer = Get-SCOMManagementServer "MgmtServer01.Contoso.com"  
PS C:\> Get-SCOMAgent -ManagementServer $MgmtServer
```

### Example 3: Get agents in a management group by using the agent name

This command gets agents that are named server01.contoso.com, that have a name that begins with Server02, and that are in the Contoso.com domain.

```
PS C:\> Get-SCOMAgent -DNSHostName "server01.contoso.com", "Server02*", "*.Contoso.com"
```

### Example 4: Get all agents in a domain

This command establishes a temporary connection with the computer named Server01.Contoso.com and gets all agents in the Contoso.com domain.

```
PS C:\> Get-SCOMAgent -DNSHostName "*.Contoso.com" -ComputerName "Server01.Contoso.com"
```

## Related topics

[Install-SCOMAgent](#)

[Uninstall-SCOMAgent](#)

[Repair-SCOMAgent](#)

[Get-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. You can use the **Set-SCOMAgentApprovalSetting** cmdlet to change the manual agent approval setting for the management group.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

## **<CommonParameters>**

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

---

## Examples

### Example 1: Get the manual approval setting

This command gets the manual agent approval setting for the management group.

```
PS C:\> Get-SCOMAgentApprovalSetting
```

### Related topics

[Set-SCOMAgentApprovalSetting](#)

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMAgentlessManagedComputer

---

## Get-SCOMAgentlessManagedComputer

Gets managed computers that do not have Operations Manager 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 managed computers that do not have System Center 2012 – Operations Manager agents.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 an array of **AgentManagedComputer** objects. This parameter specifies the Operations Manager agent that performs agentless monitoring. The action account of the agent that performs the monitoring must have local administrative rights on the computer that it monitors.

To obtain an **AgentManagedComputer** object, use the **Get-SCOMAgent** cmdlet.

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

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

Specifies a **ManagementServer** object. This parameter specifies the primary management server that performs agentless monitoring of the agentless managed computers. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

Aliases	none
Required?	false

---

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

## <CommonParameters>

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

## Examples

### Example 1: Get agentless managed computers by using a name

This command gets the agentless managed computer named server01 and agentless managed computers that have a name that begins with server0.

```
PS C:\> Get-SCOMAgentlessManagedComputer -DNSHostName "server01","server0*"
```

### Example 2: Get agentless managed computers managed by an agent

This command gets a list of agentless computers managed by Operations Manager agent. The command uses the **Get-SCOMAgent** cmdlet to get the Operations Manager agent named contoso01, and passed the result to the **Foreach-Object** cmdlet. The command gets all agentless managed computers managed by the Operations Manager agent named contoso01.

```
PS C:\> Get-SCOMAgent -DNSHostName "contoso01" | foreach {Get-SCAgentlessManagedComputer -ManagedByAgent $_}
```

## Related topics

[Add-SCOMAgentlessManagedComputer](#)

[Set-SCOMAgentlessManagedComputer](#)

[Get-SCOMAgentlessManagedComputer](#)

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

---

# Get-SCOMAlert

---

## Get-SCOMAlert

Gets Operations Manager 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 one or more alerts. An alert is an indication of a significant event that requires your attention. Rules and monitors can generate alerts.

## Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-Criteria<String>**

Specifies the criteria XML that indicates how to filter alerts. Criteria can filter on particular rules or monitors, or properties of the alert, but cannot filter on classes or groups.

Aliases	none
Required?	false
Position?	10
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

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

Specifies an array of health states. Valid values are: critical (red), warning (yellow), healthy (green).

Aliases	none
Required?	false
Position?	7
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Id<Guid[]>**

Specifies an array of GUIDs of alert objects. 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[]>**

Specifies an array of class instance objects. This parameter also accepts group objects. To obtain a class instance object, use the **Get-SCOMClassInstance** cmdlet. For more information, 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[]>**

Specifies an array of user names. The cmdlet get the alerts if the last user that edited the alert matches a user name that you specify.

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

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

Specifies an array of alert names. The cmdlet get the alerts that match the alert names that you specify.

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

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

Specifies an array of user names. The cmdlet gets an alert if the owner of an alert matches a user name that you specify.

Aliases	none
Required?	false
Position?	4

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

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

Specifies an array of alert priority levels. Valid values are:

- Low
- Medium
- High

Aliases	none
Required?	false
Position?	8
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ResolutionState<Int32[]>**

Specifies an array of resolution state IDs.

When an alert is generated, its resolution state is New. Operators can change the resolution state for a new alert to Closed or to a custom resolution state that an administrator has created for the management group. The ID for New is 0 and the ID for Closed is 255. You can assign custom resolution states any value from 2 through 254.

Aliases	none
Required?	false
Position?	5
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

---

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

Specifies an array of user names. The cmdlet get the alerts if the user that resolved the alert matches a user name that you specify.

Aliases	none
Required?	false
Position?	6
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	true

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

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

Specifies an array of severity values of alerts. Valid values are:

- Information
- Warning
- Critical

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: Get all new alerts

This command gets all alerts that have a resolution state of zero (new alerts).

```
PS C:\> Get-SCOMAlert -ResolutionState 0
```

### Example 2: Get alerts by using a custom field

This command gets all alerts that have a value of TestServer in the **CustomField1** property.

```
PS C:\> Get-SCOMAlert | where {$_.CustomField1 -eq "TestServer"}
```

### Example 3: Get alerts and suppress error messages

This example gets alerts and suppresses error messages for the class instances that do not match the criteria of the command.

The first command stores the value of the \$ErrorActionPreference variable in the \$OriginalErrorAction variable.

The second command sets the value of the \$ErrorActionPreference variable to SilentlyContinue. By setting the value of the \$ErrorActionPreference variable to SilentlyContinue, class instances that do not have matching task results continue to run and not show an error.

The third command gets all classes with health in their name and passes the class objects to the **Get-SCOMClassInstance** cmdlet by using the pipeline operator. The **Get-SCOMClassInstance** gets the class instances for each class object and passes each of the class instance objects to the **Get-SCOMAlert** cmdlet. The **Get-SCOMAlert** cmdlet returns the alerts that have a resolution state from 5 through 200, inclusive, for each class instance.

The fourth command sets the value for the \$ErrorActionPreference variable 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: Get an alert by using the ID

This command gets the alert that has the Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\> Get-SCOMAlert -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Get-SCOMClassInstance](#)

[Set-SCOMAlert](#)

[Resolve-SCOMAlert](#)

---

# Get-SCOMAlertHistory

---

## Get-SCOMAlertHistory

Gets history entries for alerts.

### Syntax

Parameter Set: FromAlertDefault

```
Get-SCOMAlertHistory [-Alert] <MonitoringAlert[]> [ <CommonParameters>]
```

### Detailed Description

The **Get-SCOMAlertHistory** cmdlet gets history entries for one or many alerts.

### Parameters

#### **-Alert<MonitoringAlert[]>**

Specifies an array of **MonitoringAlert** objects. To obtain a **MonitoringAlert** object, use the **Get-SCOMAlert** cmdlet.

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

#### **<CommonParameters>**

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

---

## Examples

### Example 1: Get the history of alerts

This command gets all alerts with heartbeat in their name and passes the alerts to the **Get-SCOMAlertHistory** cmdlet by using the pipe operator. The **Get-SCOMAlertHistory** cmdlet returns the history for each alert. The **Format-Table** cmdlet 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 an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

## **<CommonParameters>**

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

---

## Examples

### Example 1: Get the automatic alert resolution setting

This command gets the automatic alert resolution for the management group.

```
PS C:\> Get-SCOMAlertResolutionSetting
```

## Related topics

[Set-SCOMAlertResolutionSetting](#)

---

# Get-SCOMAlertResolutionState

---

## Get-SCOMAlertResolutionState

Gets the alert resolution states 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 in the management group. Each resolution state is assigned an ID, a code number which uniquely identifies that resolution state. The ID for New is 0 and the ID for Closed is 255. You can assign custom resolution states any value from 2 through 254.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of names of alert resolution states.

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

### **-ResolutionStateCode<Byte[]>**

Specifies a resolution state ID.

---

Operations Manager defines two resolution states: New (0) and Closed (255). You can assign custom resolution states any value from 2 through 254.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

### **<CommonParameters>**

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

## **Examples**

### **Example 1: Get all resolution states**

This command gets all resolution states in the management group.

```
PS C:\> Get-SCOMAlertResolutionState
```

---

## Example 2: Get resolution states by using a resolution state code

This command gets information about the resolution state that has the code 42.

```
PS C:\> Get-SCOMAlertResolutionState -ResolutionStateCode 42
```

### Related topics

[Add-SCOMAlertResolutionState](#)

[Remove-SCOMAlertResolutionState](#)

---

# Get-SCOMClass

---

## Get-SCOMClass

Gets classes in Operations Manager.

### 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 gets one or more classes defined by System Center 2012 – Operations Manager or an imported management pack.

## Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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.

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

---

## **-Id<Guid[]>**

Specifies an array of GUIDs of classes. If you specify an Id as a string, the cmdlet converts the string 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 array of **ClassInstance** objects. To obtain a **ClassInstance** object, use the **Get-SCOMClassInstance** object.

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

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

Specifies an array of **ManagementPack** objects. To obtain a **ManagementPack** object, use 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 an array of names of classes.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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 pass an instance of a management pack to the *Instance* parameter of the **Get-SCOMClass** cmdlet by using

---

the pipe operator. The `Microsoft.EnterpriseManagement.Common.EnterpriseManagementObject` object is one the properties of the output object of the **Get-SCOMClassInstance** cmdlet.

- **System.Guid**You can pass a GUID to the *Id* parameter of the **Get-SCOMClass** cmdlets by using the pipe operator.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPackManagementPack**You can pass a management pack to the *ManagementPack* parameter of the **Get-SCOMClass** cmdlet by using the pipe operator. This management pack object contains the class object.
- **System.String**You can pass a name to the *Name* parameter of the **Get-SCOMClass** cmdlet by using the pipe operator.

## 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: Get a class by using a name

This command gets all classes that have a name that ends with user.

```
PS C:\> Get-SCOMClass -Name "*user"
```

### Example 2: Get a class by using a display name

This command gets the class that has the display name User.

```
PS C:\> Get-SCOMClass -DisplayName "User"
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMManagementPack](#)

---

# Get-SCOMClassInstance

---

## Get-SCOMClassInstance

Gets 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 one or more class instances. A class represents a kind of object, and every object in System Center 2012 – Operations Manager is considered an instance of a particular class. All instances of a class share a common set of properties.

---

## Parameters

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

Specifies an array of **ManagementPackClass** objects. To obtain a **ManagementPackClass** object, use the **Get-SCOMClass** cmdlet.

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

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names of objects. Values for this parameter depend on which localized management packs you import and the locale of the user that runs Windows PowerShell.

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

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

Specifies an array of **EnterpriseManagementObject** objects. To obtain a **EnterpriseManagementObject** object, use the **Get-SCOMGroup** cmdlet. For more information, 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[]>**

Specifies an array of GUIDs of class instances. 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 an array of names of objects.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

---

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

false
-------

## <CommonParameters>

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

## Examples

### Example 1: Gets class instances by using a display name

This command gets the class instance that has the display name Server01.Contoso.Com and the class instances in the contoso.com domain.

```
PS C:\> Get-SCOMClassInstance -Displayname "Server01.Contoso.Com", "*.contoso.com"
```

### Example 2: Gets class instances by using a name

This command gets class instances of classes. The command uses the **Get-SCOMClass** to get all classes that have "Windows" in their name, and passes the result to the **Get-SCOMClassInstance** cmdlet by using the pipeline operator. The command gets the class instances for the classes that have Windows in their name.

```
PS C:\> Get-SCOMClass -Name "*Windows*" | Get-SCOMClassInstance
```

### Example 3: Gets class instances by using an Id

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 System Center 2012 – Operations Manager commands in the current session. If you do not specify any parameters for the cmdlet, the cmdlet returns all commands in the current session.

### Parameters

#### **-CommandType<CommandTypes>**

Specifies a Operations Manager command type.

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

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

Specifies an array of names of Operations Manager commands.

---

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

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

Specifies an array of nouns in Operations Manager cmdlet names.

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

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

Specifies an array of verbs in Operations Manager cmdlet names.

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

---

## <CommonParameters>

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

## Examples

### Example 1: Get commands by using a name

This command gets all commands in the Operations Manager module that have agent in the name of the command.

```
PS C:\> Get-SCOMCommand "*agent*"
```

### Example 2: Get commands by using a cmdlet verb

This command gets all commands in the Operations Manager module that use the verb Get.

```
PS C:\> Get-SCOMCommand -Verb "Get"
```

## Related topics

[Write-SCOMCommand](#)

---

# Get-SCOMConnector

---

## Get-SCOMConnector

Gets Operations Manager 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 one or more configuration item (CI) connectors for System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 connector. Values for this parameter depend on which localized management packs you import and the locale of the user that runs Windows PowerShell.

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

---

## **-Id<Guid[]>**

Specifies an array of GUIDs of connectors. 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 an array of names of connectors.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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

---

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

false
-------

## <CommonParameters>

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

## Examples

### Example 1: Get connectors by using a display name

This command gets all connectors that have a display name that begins with Connector.

```
PS C:\> Get-SCOMConnector -DisplayName "Connector*"
```

### Example 2: Get connectors by using a name

This command gets all connectors that have a name that begins with Operations Manager.

```
PS C:\> Get-SCOMConnector -Name "Operations Manager*"
```

### Example 3: Get a connector by using an Id

This command gets the connector that has the Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\> Get-SCOMConnector -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Add-SCOMConnector](#)

[Remove-SCOMConnector](#)

---

# Get-SCOMDatabaseGroomingSetting

---

## Get-SCOMDatabaseGroomingSetting

Gets the database grooming settings for a 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 a management group. Database grooming automatically removes unnecessary data from the System Center 2012 – Operations Manager database in order to maintain performance. Use the **Set-SCOMDatabaseGroomingSetting** cmdlet to make changes to grooming settings.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get database grooming settings for the current management group

This command gets database grooming settings. This command does not specify values for the *ComputerName*, *Credential*, or *SCSession* parameters; therefore, the command uses the active persistent connection to determine the management group.

```
PS C:\> Get-SCOMDatabaseGroomingSetting
```

### Related topics

[Set-SCOMDatabaseGroomingSetting](#)

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMDataWarehouseSetting

---

## Get-SCOMDataWarehouseSetting

Gets data warehouse settings for a 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 a management group. Use the **Set-SCOMDataWarehouseSetting** to modify data warehouse settings for a management group.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get data warehouse settings

This command gets the data warehouse settings for the current management group. This command does not specify values for the *ComputerName*, *Credential*, or *SCSession* parameters; therefore the cmdlet uses the current active persistent management group connection.

```
PS C:\> Get-SCOMDataWarehouseSetting
```

### Related topics

[Set-SCOMDataWarehouseSetting](#)

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMDiagnostic

---

## Get-SCOMDiagnostic

Gets 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. A diagnostic task attempts to discover the cause of a problem or provide you with additional information.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary

---

connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.). The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

---

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

Specifies an array of display names. Values for this parameter depend on which localized management packs you import and the locale of the user that runs Windows PowerShell.

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

## **-Id<Guid[]>**

Specifies an array of GUIDs of diagnostics.

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

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

Specifies an array of management pack objects. To obtain a management pack object, use the **Get-SCOMManagementPack** cmdlet.

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

---

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

Specifies an array of **ManagementPackMonitor** objects. To obtain a monitor, use the **Get-SCOMMonitor** cmdlet.

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

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

Specifies an array of names of a diagnostics.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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 an array of **ManagementPackClass** objects. To obtain a **ManagementPackClass** object, use the **Get-SCClass** cmdlet. These classes are the targets of diagnostics.

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: Get diagnostics**

This command gets diagnostics related to network management. The command searches for diagnostics that have names that contain the string NetworkManagement.

```
PS C:\> Get-SCOMDiagnostic -Name "*NetworkManagement*"
```

## **Related topics**

[Get-SCOMManagementPack](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMMonitor](#)

[Get-SCClass](#)

---

# Get-SCOMDiscovery

---

## Get-SCOMDiscovery

Gets Operations Manager discoveries.

### 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 gets System Center 2012 – Operations Manager discoveries. Specify a display name, ID, name, management pack, or class for a discovery. For the name and display name, you can use wildcards.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, Localhost , or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names. Values for this parameter depend on which localized management packs you import and the locale of the user that runs Windows PowerShell.

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

### **-Id<Guid[]>**

Specifies an array of IDs for discoveries. Specify a GUID or a string to convert to a GUID.

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

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

Specifies an array of management pack objects. To obtain a management pack object, use the **Get-SCOMManagementPack** cmdlet. Management packs contain discoveries.

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

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

Specifies an array of names of discoveries.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

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

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

Specifies an array of **ManagementPackClass** objects. To obtain a target, use the **Get-SCOMClass** cmdlet.

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

---

## <CommonParameters>

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

## 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 this cmdlet.
- **System.String**You can pipe a discovery name to the *Name* parameter of this cmdlet.
- **System.Guid**You can pipe a GUID of a discovery object to the *Id* parameter of this cmdlet.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPackClass**You can pipe a management pack class to the *Target* parameter of this 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: Get discoveries

This command gets the discoveries defined in Operations Manager.

```
PS C:\> Get-SCOMDiscovery
```

### Example 2: Get discoveries by using wildcards

This command gets discoveries. The command specifies names that contain PopulateRootM.

```
PS C:\> Get-SCOMDiscovery -Name "*PopulateRootM*"
```

## Related topics

[Get-SCOMManagementPack](#)

[Get-SCOMClass](#)

[Get-SCOMManagementGroupConnection](#)

[Disable-SCOMDiscovery](#)

[Enable-SCOMDiscovery](#)

[Get-SCOMDiscovery](#)

---

# Get-SCOMErrorReportingSetting

---

## Get-SCOMErrorReportingSetting

Gets the error reporting settings for a 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 a management group. You can change the setting by using the **Set-SCOMErrorReportingSetting** cmdlet.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get error reporting setting

This command gets the error reporting setting for the current management group.

```
PS C:\> Get-SCOMErrorReportingSetting
```

### Related topics

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMEvent

---

## Get-SCOMEvent

Gets Operations Manager events.

### 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 System Center 2012 – Operations Manager events. Rules collect events.

You can specify events by ID or by the managed object. You can also specify the rule that the event triggers.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.). The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`. If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-EventId<Int32[]>**

Specifies an array of event IDs. An **SCOMEvent** object contains an event ID as its **Number** property.

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

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

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

Specifies an array of names of event logs. An **SCOMEEvent** object contains an event log name as its **Channel** property.

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

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

Specifies an array of event sources. An **SCOMEEvent** object contains an event source as its **PublisherName** property.

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

### **-Id<Guid[]>**

Specifies an array of GUIDs of events. An **SCOMEEvent** object contains GUID as its **Id** property.

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

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

Specifies an array of monitoring objects that represent instances. To obtain monitoring objects, use the **Get-SCOMClassInstance** cmdlet.

This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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

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

Specifies an array of rules as **ManagementPackRule** objects. To obtain a **ManagementPackRule** object, use the **Get-SCOMRule** cmdlet.

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

---

## -SCSession<Connection[]>

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get events for WMI rules

This example gets all the health events matching rules that contain WMI in their display names. The first command gets all monitoring rules that have WMI in their display names and stores them in the \$Rules variable.

The second command gets all classes with health in the display name and uses the pipeline operator (|) to pass the class objects to the **Get-SCOMClassInstance** cmdlet. That cmdlet 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 a value of 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: Get an event by using its ID

This command gets the event that has an ID of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\> Get-SCOMEvent -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

---

## Example 3: View events grouped by event IDs

This command gets all events and then passes them to the **Group-Object** cmdlet by using the pipeline operator. That cmdlet displays the events grouped by the **Number** property of the **SCOMEvent** object, which corresponds to an integer event ID. For more information, type `Get-Help Group-Object`.

```
PS C:\> Get-SCOMEvent | Group-Object -Property Number
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMRule](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMGroup](#)

---

# Get-SCOMGatewayManagementServer

---

## Get-SCOMGatewayManagementServer

Gets the gateway management servers in a management group.

### 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 in a management group.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named

Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of names for gateway management servers. You can use wildcards.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get gateway management servers by using names

This command gets the gateway management server named Server01.ContosoPartner.com and all gateway management servers that are in the Contoso.com domain.

```
PS C:\> Get-SCOMGatewayManagementServer -Name "Server01.ContosoPartner.com", "*.Contoso.com"
```

#### Example 2: Get gateway management servers for a different management group

This command gets the gateway management servers for a management group that the server Server01.Contoso.com belongs to. Because the command does not include the *Credential* parameter, the current user must have access rights for Server01.Contoso.com.

```
PS C:\> Get-SCOMGatewayManagementServer -ComputerName "Server01.Contoso.com"
```

### Related topics

[Get-SCOMManagementServer](#)

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMGroup

---

## Get-SCOMGroup

Gets Operations Manager groups.

### Syntax

Parameter Set: Empty

```
Get-SCOMGroup [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Parameter Set: FromGroupDisplayName

```
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 System Center 2012 – Operations Manager groups. You can specify which groups to get by name or ID.

Because a group object is a type of class instance object, it can be passed to the *Instance* parameter of another cmdlet, such as the **Enable-SCOMDiscovery** cmdlet.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names. Values for this parameter depend on which localized management packs you import and the locale of the user that runs Windows PowerShell.

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

---

## **-Id<Guid[]>**

Specifies an array of GUIDs of groups. An **SCOMGroup** object contains a GUID as its **Id** property.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get groups by using display names**

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: Get a group by using an ID

This command gets the group that has an Id of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\> Get-SCOMGroup -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMManagementGroupConnection](#)

[Enable-SCOMDiscovery](#)

---

# Get-SCOMHeartbeatSetting

---

## Get-SCOMHeartbeatSetting

Gets heartbeat settings for a 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 agents and servers in an System Center 2012 – Operations Manager management group. Use the **Set-SCOMHeartbeatSetting** cmdlet to modify the heartbeat interval or the missing heartbeat threshold.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get heartbeat settings

This command gets heartbeat settings for the management group.

```
PS C:\> Get-SCOMHeartbeatSetting
```

### Related topics

[Get-SCOMManagementGroupConnection](#)

[Set-SCOMHeartbeatSetting](#)

---

# Get-SCOMLicense

---

## Get-SCOMLicense

Displays Microsoft Software License Terms.

### 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 current System Center 2012 – Operations Manager product license. You can also view the license information in the Operations Console Help About dialog box.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get the current license terms

This command displays the Microsoft Software License Terms for the current product license.

```
PS C:\> Get-SCOMLicense
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Set-SCOMLicense](#)

---

# Get-SCOMLocation

---

## Get-SCOMLocation

Gets locations.

### 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 locations. Specify locations to get by display name or ID, or by specifying associated agents, management servers, or resource pools.

You can create a location by using the **New-SCOMLocation** cmdlet.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

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

Specifies an array of **AgentManagedComputer** objects. To obtain an **AgentManagedComputer** object, use the **Get-SCOMAgent** cmdlet.

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

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names. Values for this parameter depend on which localized management packs you import and the locale of the user that runs Windows PowerShell.

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

### **-Id<Guid[]>**

Specifies an array of unique IDs of locations.

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

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

Specifies an array of management server objects. To obtain a management server object, use the **Get-SCOMManagementServer** cmdlet.

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

### **-Pool<ManagementServicePool[]>**

Specifies an array of resource pool objects. To obtain a resource pool object, use the **Get-SCOMResourcePool** cmdlet.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get a location by display name

This command gets the location that has the display name Seattle, WA.

```
PS C:\> Get-SCOMLocation -DisplayName "Seattle, WA"
```

### Example 2: Get the location of an agent-managed computer

This example gets a location for an agent-managed computer. The first command gets the agent object for an agent-managed computer named Server01.Contoso.com and then stores the object in the \$Agent variable.

The second command gets the location for the agent object stored in the \$Agent variable.

```
PS C:\> $Agent = Get-SCOMAgent -Name "Server01.Contoso.com"
PS C:\> Get-SCOMLocation -Agent $Agent
```

### Example 3: Get the location of a management server

This example gets the location for a management server named MgmtServer01.Contoso.com. The first command gets the management server object for the management server named MgmtServer01.Contoso.com, and then stores the object in the \$MgmtSvr variable.

The second command gets the location for the management server object stored in the \$MgmtSvr variable.

```
PS C:\> $MgmtSvr = Get-SCOMManagementServer -Name "MgmtServer01.Contoso.com"
PS C:\> Get-SCOMLocation -ManagementServer $MgmtSvr
```

### Example 4: Get the location of a resource pool

This example gets a location for the resource pool named Pool01. The first command gets the resource pool object for the resource pool named Pool01 and stores the object in the \$Pool variable.

The second command gets the location for the pool stored in the \$Pool variable.

```
PS C:\> $Pool = Get-SCOMResourcePool -Name "Pool01"
PS C:\> Get-SCOMLocation -Pool $Pool
```

---

## Related topics

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Get-SCOMResourcePool](#)

[Get-SCOMManagementGroupConnection](#)

[New-SCOMLocation](#)

[Remove-SCOMLocation](#)

[Set-SCOMLocation](#)

[Update-SCOMLocation](#)

---

# Get-SCOMMaintenanceMode

---

## Get-SCOMMaintenanceMode

Gets 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 maintenance mode entries. You can use this cmdlet to provide objects to the **Set-SCOMMaintenanceMode** cmdlet, or by itself to see details about maintenance mode entries.

When a resource is in maintenance mode, System Center 2012 – Operations Manager suppresses alerts, notifications, rules, monitors, automatic responses, state changes, and new alerts.

When you use this cmdlet by itself, it presents information for the local time. If you pipe its results to the **Format-Table** cmdlet, the cmdlet provides information in Coordinated Universal Time (UTC). For more information about **Format-Table**, type, `Get-Help Format-Table`.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-History**

Indicates that the cmdlet gets all maintenance mode entries, including inactive entries. Without this parameter, the cmdlet gets only active maintenance mode entries.

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

---

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

Specifies an array of monitoring objects that represent instances. To obtain monitoring objects, use the **Get-SCOMClassInstance** cmdlet.

This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get all active maintenance mode entries**

This command gets all active maintenance mode entries. In order to include inactive entries, specify the *History* parameter.

```
PS C:\> Get-SCOMMaintenanceMode
```

---

## Example 2: Get all active maintenance mode entries for a specified domain

This command gets all active maintenance mode entries for class instances in the Contoso.com domain. The command uses the **Get-SCOMClassInstance** cmdlet to get the instances that contain Contoso.com.

If the command encounters a class instance that lacks active maintenance mode entries, 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: Get all maintenance mode entries for a specified domain

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. This example uses the *History* parameter; therefore, the command gets all maintenance mode entries, including inactive entries.

```
PS C:\> Get-SCOMClassInstance -Name "*.Contoso.com" | Get-SCOMMaintenanceMode -History
```

## Related topics

[Get-SCOMClassInstance](#)

[Set-SCOMMaintenanceMode](#)

[Start-SCOMMaintenanceMode](#)

[Get-SCOMGroup](#)

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMManagementGroup

---

## Get-SCOMManagementGroup

Gets a management group.

### Syntax

Parameter Set: Empty

```
Get-SCOMManagementGroup [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-SCOMManagementGroup** cmdlet gets a **ManagementGroup** object that represents a System Center 2012 – Operations Manager management group. You can get a management group for a specific computer or session by specifying values for the *ComputerName* parameter or *SCSession* parameter. You may need to provide credentials for a computer. If you do not specify a computer or session, the cmdlet gets the current management group.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get the current management group

This command gets a **ManagementGroup** object for the current management group.

```
PS C:\> Get-SCOMManagementGroup
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMManagementGroupConnection

---

## Get-SCOMManagementGroupConnection

Gets management group connections.

### 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 gets objects that represent persistent connections to System Center 2012 – Operations Manager management groups. You can get all connections, or get connections for specific computers or management groups. For more information about persistent connections, type `Get-Help about_OpsMgr_Connections`.

You can use the **New-SCOMManagementGroupConnection** to create a connection and you can use the **Remove-SCOMManagementGroupConnection** to remove a connection.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 an array of IDs of management groups.

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

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

Specifies an array of names of the management groups.

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 this 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: Get all persistent connections

This command gets all management group connections.

```
PS C:\> Get-SCOMManagementGroupConnection
```

## Related topics

[New-SCOMManagementGroupConnection](#)

[Remove-SCOMManagementGroupConnection](#)

[Set-SCOMManagementGroupConnection](#)

---

# Get-SCOMManagementPack

---

## Get-SCOMManagementPack

Gets management packs.

### 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 gets System Center 2012 – Operations Manager management packs. You can get management packs from a management group, from a management pack file with an .mp or .xml file name extension, or from a management pack bundle with an .mpb file name extension.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

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

Specifies an array of names of management pack bundle files. Bundle files use an .mpb file name extension.

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

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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 an array of display names. Values for this parameter depend on which localized management packs you import and the locale of the user that runs Windows PowerShell. You can use regular expressions.

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

### **-Id<Guid[]>**

Specifies an array of GUIDs.

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

---

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

Specifies an array of management pack files. Management pack files use .xml or .mp file name extensions.

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

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

Specifies an array of names of management pack files. You can use regular expressions.

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

## **-Recurse**

Indicates that the cmdlet recursively gets all management packs that depend on specified management packs.

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

---

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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 GUIDs of management packs to the *Id* parameter of this cmdlet.
- **System.String**You can pipe names of management packs to the *Name* parameter of this cmdlet.

## **Outputs**

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

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack**A management pack object contains a management pack and its properties.

## **Examples**

### **Example 1: Get management packs from a management group**

This command gets all the management packs for the current management group.

```
PS C:\> Get-SCOMManagementPack
```

---

## Example 2: Get a management pack from a management group by using its name

This command gets a management pack named System.Library from the current management group.

```
PS C:\> Get-SCOMManagementPack -Name "System.Library"
```

### Related topics

[Import-SCOMManagementPack](#)

[New-SCOMManagementPackBundle](#)

[Export-SCOMManagementPack](#)

[New-SCOMManagementPack](#)

[Protect-SCOMManagementPack](#)

[Remove-SCOMManagementPack](#)

[Test-SCOMManagementPack](#)

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMManagementServer

---

## Get-SCOMManagementServer

Gets the management servers in a 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 a management group.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	localhost

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of names for management servers. You can use wildcards.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get management servers by using names

This command gets the management servers named Server01.ContosoPartner.com or that have a domain of Contoso.com. This command refers to the default management group, which is the management group for the active persistent connection.

```
PS C:\> Get-SCOMManagementServer -Name "Server01.ContosoPartner.com","*.Contoso.com"
```

### Example 2: Get management servers for a management group

This command gets the management servers for a management group that the server Server01.Contoso.com belongs to. In order for this command to work, the current user must have rights for that server.

```
PS C:\> Get-SCOMManagementServer -ComputerName "Server01.Contoso.com"
```

## Related topics

[Get-SCOMGatewayManagementServer](#)

[Get-SCOMManagementGroupConnection](#)

---

# Get-SCOMMonitor

---

## Get-SCOMMonitor

Retrieves monitors in Operations Manager.

### 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 retrieves monitors in System Center 2012 – Operations Manager. In Operations Manager, monitors define logic for determining the health of an object.

---

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the computer or computers that the *ComputerName* parameter specifies. The management group connection runs under this credential. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of display name objects. Values of the *DisplayName* parameter may vary depending on which localized management packs a user imports into the management group and the locale of the user who is running Windows PowerShell.

Aliases	none
Required?	true

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

### **-Id<Guid[]>**

Specifies an array of unique identifiers for monitors.

The **SCOMMonitor** object stores the GUID in the **Id** property of the object. For more information, 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[]>**

Specifies an array of class instance objects. The *Instance* parameter also accepts group objects as input. 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[]>**

Specifies one or more management pack objects where you can save the override. If an unsealed management pack contains the monitor, save the override in the same management pack. 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 an array of names of an object.

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

## **-Recurse**

Indicates that the cmdlet searches recursively.

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

---

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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 an array of monitors for one or more target class objects. 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: Retrieve a monitor by using a name**

This command retrieves all monitors whose names begin with the string "System.Health".

```
PS C:\> Get-SCOMMonitor -Name "System.Health*"
```

---

## Example 2: Retrieve a monitor by using a display name

This command retrieves all monitors whose display names contain the string "Performance".

```
PS C:\> Get-SCOMMonitor -DisplayName "*Performance*"
```

## Example 3: Retrieve monitors for a management pack

This command retrieves the management pack that has the display name of System Center Core Library, and then passes the management pack object to the **Get-SCOMMonitor** cmdlet by using the pipeline operator (|). The **Get-SCOMMonitor** cmdlet gets all monitors for the management pack object.

```
PS C:\> Get-SCOMManagementPack -DisplayName "System Center Core Library" | Get-SCOMMonitor
```

## Example 4: Retrieve monitors from a management pack

This example retrieves monitors in a management pack.

The first command gets the path to a management pack file and stores the value in the variable named \$MPFile.

The second command uses the **Get-SCOMMonitor** cmdlet with the *ManagementPack* parameter to get the monitors from the management pack.

```
PS C:\> $MPFile = "C:\Program Files\System Center Operations Manager  
2007\Microsoft.SystemCenter.2007.mp"  
PS C:\> Get-SCOMMonitor -ManagementPack $MPFile
```

## Example 5: Retrieve monitors by display name

This example retrieves monitors by display name.

The first command gets all classes that contain "health" in their display names, and then uses the pipeline operator (|) to pass the class objects to the **Get-SCOMClassInstance** cmdlet. The **Get-SCOMClassInstance** cmdlet 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: Retrieve monitors by using a target class

This example retrieves monitors by using a target class.

This command gets the monitors that have target classes that include "health" in their display names.

```
PS C:\> Get-SCOMMonitor -Target (Get-SCOMClass -DisplayName *health*)
```

---

## Example 7: Retrieve a monitor by using an ID

This command retrieves the monitor that has the ID 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

Retrieves notification channels for 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 retrieves the notification channels for the management group. Notification channels inform administrators of an alert, or they run automation in response to an alert. A notification channel uses a delivery mechanism in System Center 2012 – Operations Manager, such as email, instant message, Short Message Service, or command, to deliver notifications.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 a channel.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Get all notification channels

This command gets all notification channels.

```
PS C:\> Get-SCOMNotificationChannel
```

### Example 2: Get notification channels by using a display name

This command gets the notification channel named Email channel.

```
PS C:\> Get-SCOMNotificationChannel -DisplayName "Email channel"
```

### Example 3: Get notification channels by using a name

This command gets the notification channel named MyCustomChannel.

```
PS C:\> Get-SCOMNotificationChannel -DisplayName "MyCustomChannel"
```

## Related topics

[Add-SCOMNotificationChannel](#)

[Remove-SCOMNotificationChannel](#)

---

# Get-SCOMNotificationSubscriber

---

## Get-SCOMNotificationSubscriber

Retrieves a list of notification subscribers.

### 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 retrieves a list of notification subscribers in System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential".

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-Id<Guid[]>**

Specifies an array of GUIDs of subscribers.

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

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

Specifies an array of names of subscribers.

Aliases	none
Required?	true
Position?	1

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Retrieve all notification subscribers**

This command retrieves all notification subscribers.

```
PS C:\> Get-SCOMNotificationSubscriber
```

### **Example 2: Retrieve a notification subscriber by using a name**

This command retrieves the notification subscriber named SarahJones.

```
PS C:\> Get-SCOMNotificationSubscriber -Name "SarahJones"
```

## **Related topics**

[Add-SCOMNotificationSubscriber](#)

[Remove-SCOMNotificationSubscriber](#)

---

# Get-SCOMNotificationSubscription

---

## Get-SCOMNotificationSubscription

Retrieves 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 retrieves a list of notification subscriptions in System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of display names of an object. Values for this parameter depend on the localized management packs you import and the locale of the user that runs Windows PowerShell.

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

### **-Id<Guid[]>**

Specifies an array of GUIDs.

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

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

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

Specifies an array of names to match against the **Name** property of the given objects.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Retrieve notification subscriptions by display name

This command retrieves the notification subscription that has the display name Subscription01 and displays information about the subscription to the user.

```
PS C:\> Get-SCOMNotificationSubscription -DisplayName "Subscription01"
```

## Related topics

[Add-SCOMNotificationSubscription](#)

[Disable-SCOMNotificationSubscription](#)

[Enable-SCOMNotificationSubscription](#)

[Remove-SCOMNotificationSubscription](#)

---

# Get-SCOMOverride

---

## Get-SCOMOverride

Retrieves 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 retrieves a list of overrides, or a resulting set of overrides. In System Center 2012 – Operations Manager, overrides represent changes to parameters in monitors.

## Parameters

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

Specifies an array of class objects. For more information, 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 an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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[]>**

Specifies an array of diagnostic objects. For more 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[]>**

Specifies an array of discovery objects. For more 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[]>**

Specifies an array of group objects. For more 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[]>**

Specifies an array of class instance objects. This parameter also accepts group objects. For more  
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[]>**

Specifies an array of monitor objects. For more 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[]>**

Specifies an array of recovery objects. For more 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[]>**

Specifies an array of monitoring rule objects. For more 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 an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCManagementGroupConnection** cmdlet.

Aliases	none
Required?	false

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

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

Specifies an array of task objects. For more 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: Retrieve all overrides for a set of rules**

This example gets all monitoring rules that contain the word 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: Retrieve all overrides for a specific class**

This command gets the override for monitors that contain the word health in their name and are in a class with a display name that contains the word computer.

---

```
PS C:\> Get-SCOMOverride -Monitor (Get-SCOMMonitor -Name "*health*") -Class (Get-SCOMClass -
DisplayName "*computer*")
```

### Example 3: Retrieve all overrides for a set of discovery objects

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

Retrieves 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 retrieves existing overrides associated with specified workflows that are constrained to a specified class or class instance.

---

## Parameters

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

Specifies an array of management pack class objects. For more 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 an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of discovery workflow objects to retrieve. For more information about how to get discovery workflow objects, 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 array of instances of a class to retrieve.

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

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

Specifies an array of monitor workflow objects to retrieve. For more information, 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 an array of monitoring rule workflow objects to retrieve. For information about how to get a rule object, type "Get-He1p Get-SCOMRule".

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Retrieve a list of overrides for a specific class

This example gets the override result for a specific class.

The first command uses the **Get-SCOMClass** cmdlet to get the class object named Memory and stores the object in the `$Class` variable.

The second command uses the **Get-SCOMDiscovery** cmdlet to get the discovery object with the display name Discover Windows Server Computers, and stores the object in the `$Discovery` variable.

The third command uses the **Get-SCOMManagementGroupConnection** cmdlet to get the management group connection object for Server01 and stores the object in the `$Session` variable.

The last command uses the **Get-SCOMOverrideResult** cmdlet to get 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 -SCSession $Session
```

### Example 2: Retrieve override results by class instance

This example gets the override result for a specific class instance.

The first command uses the **Get-SCOMClassInstance** cmdlet to get the class instance object named Operations Manager Agents and stores the object in the `$Instance` variable.

The second command uses the **Get-SCOMMonitor** cmdlet to get the monitor object with the display name Management Service connectivity state and stores the object in the `$Monitor` variable.

The last command uses the **Get-SCOMOverrideResult** cmdlet to get 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: Retrieve override results by monitoring rule object

This example gets the override result for a monitoring rule object.

The first command uses the **Get-SCOMRule** cmdlet to get 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 uses the **Get-SCOMClass** cmdlet to get the class object named Memory and uses the pipeline operator to pass the object to the **Get-SCOMOverrideResult** cmdlet, 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: Retrieve override results for a display name

This example gets the override result for a specific display name.

The first command uses the **Get-SCOMDiscovery** cmdlet to get the discovery object with the display name Discover Windows Server Computers and stores the object in the \$Discovery variable.

The second command uses the **Get-SCOMClassInstance** cmdlet to get 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
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMDiscovery](#)

[Get-SCOMMonitor](#)

[Get-SCOMRule](#)

---

# Get-SCOMPparentManagementServer

---

## Get-SCOMPparentManagementServer

Gets the management servers to which an agent reports.

### Syntax

Parameter Set: FromAgent

```
Get-SCOMPparentManagementServer [-Agent] <AgentManagedComputer[]> [ <CommonParameters>]
```

Parameter Set: FromGatewayManagementServer

```
Get-SCOMPparentManagementServer [-GatewayServer] <ManagementServer[]> [ <CommonParameters>]
```

### Detailed Description

The **Get-SCOMPparentManagementServer** cmdlet retrieves the management servers to which an agent reports. In System Center 2012 – Operations Manager, primary and failover management servers provide redundancy for agents and gateway management servers.

### Parameters

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

Specifies an array of one or more agent objects. For more 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 an array of gateway management servers. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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: Retrieve a list of management servers

This command retrieves the management servers to which the agent named Server01 reports.

The command in parentheses executes first and retrieves the agent named Server01. The cmdlet then uses the results of the command in parentheses as input to the *Agent* parameter.

```
PS C:\> Get-SCOMParentManagementServer -Agent (Get-SCAgent -Name "Server01.Contoso.com")
```

### Example 2: Retrieve a list of management servers by wildcard

This example retrieves a list of management server for agent names that match a specific string.

The first command uses the **Get-SCAgent** cmdlet to get agents that begin with the string Server with a wildcard character. The cmdlet stores the agents in the \$Agents variable.

The second command uses the **Get-SCOMParentManagementServer** cmdlet to get the management servers in the \$Agents variable.

```
PS C:\> $Agents = "Server*.Contoso.com" | Get-SCAgent
```

```
PS C:\> Get-SCOMParentManagementServer -Agent $Agents
```

---

### Example 3: Retrieve a list of management server using pipes

This command passes the agent named Server01.Contoso.com to the **Get-SCAgent** cmdlet by using the pipeline operator, and then passes the output to the **Get-SCOMPparentManagementServer** cmdlet.

```
PS C:\> "Server01.Contoso.com" | Get-SCAgent | Get-SCOMPparentManagementServer
```

### Related topics

[Set-SCOMPparentManagementServer](#)

---

# Get-SCOMPendingManagement

---

## Get-SCOMPendingManagement

Retrieves pending agent management actions.

### Syntax

Parameter Set: Empty

```
Get-SCOMPendingManagement [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-SCOMPendingManagement** cmdlet retrieves pending agent management actions in System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Retrieve all pending management entries**

This command uses the **Get-SCOMPendingManagement** cmdlet to retrieve all pending management entries and sort them by the AgentName property.

```
PS C:\> Get-SCOMPendingManagement | Sort AgentName
```

---

## Example 2: Retrieve pending management entries for a specific action

This command uses the **Get-SCOMPendingManagement** cmdlet to get the agent management entries that are pending with an action of ManualApproval, and sort the returned entries by the AgentName property.

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

## Example 3: Retrieve pending management entries and group by name

This command uses the **Get-SCOMPendingManagement** cmdlet to get all pending management entries, group them by their pending action type, and then sort them in descending order by the number of entries there are per action type group. The command 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 in Operations Manager.

### 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 in System Center 2012 – Operations Manager.

---

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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. Values of the *DisplayName* parameter may vary depending on which localized management packs a user imports into the management group and the locale of the user who is running Windows PowerShell.

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

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

### **-Id<Guid[]>**

Specifies an array of GUIDs.

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

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

Specifies an array of management pack. To obtain a management pack object, use 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 an array of monitors.

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

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

Specifies an array of names

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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 an array of target classes to use to limit the results of this cmdlet. To obtain a management pack class object, use 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: Retrieve all recoveries by name

This command retrieves all recoveries related to the Health Service.

```
PS C:\> Get-SCOMRecovery -Name Microsoft.SystemCenter.HealthService.Recovery.*
```

## Related topics

[Get-SCClass](#)

[Get-SCManagementGroupConnection](#)

[Get-SCManagementPack](#)

---

# 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 System Center 2012 – Operations Manager.

---

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names 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 an array of GUIDs 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 an array of management packs containing the relationships to retrieve. To obtain a **ManagementPack** object, use 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 an array of names 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 an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCManagementGroupConnection** cmdlet.

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

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

Specifies an array of classes that represent the source of the relationship. If you specify more than one class, the cmdlet returns any relationship that includes one of those classes as a source. The source class of the relationship must match the specified class type. To obtain a **ManagementPackClass** object, use the **Get-SCOMClass** cmdlet.

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 an array of classes that represent the target of the relationship. If you specify more than one class, the cmdlet returns any relationship that includes one of those classes as a target. The target class of the relationship must match the specified class type. To obtain a **ManagementPackClass** object, use the **Get-SCOMClass** cmdlet.

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 pass a source class to the *Source* parameter of the Get-SCOMRelationship cmdlet by using the pipeline operator.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPackClass** You can pass a target class to the *Target* parameter of the Get-SCOMRelationship cmdlet by using the pipeline operator.
- **System.Guid** You can pass a GUID to the *Id* parameter of the Get-SCOMRelationship cmdlet by using the pipeline operator.
- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** You can pass a management pack to the *ManagementPack* parameter of the Get-SCOMRelationship cmdlet by using the pipeline operator.
- **System.String** You can pipe a name to the *Name* parameter of the Get-SCOMRelationship cmdlet by using the pipeline operator.

---

## Outputs

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

## Examples

### Example 1: Retrieve relationship information

This command retrieves information about the first three relationship objects.

```
PS C:\> Get-SCOMRelationship | Select-Object -first 3
```

### Example 2: Retrieve relationship information for configuration items

This example retrieves relationships that target configuration items.

The first command uses the **Get-SCOMClass** cmdlet to retrieve the class for an item by name, and stores the result in the variable named X.

The second command uses the **Get-SCOMRelationship** cmdlet to get the relationship.

```
PS C:\> $X = Get-SCOMClass -Name "system.configitem"  
PS C:\> Get-SCOMRelationship -Target $X
```

### Example 3: Retrieve relationship information for derived types

This example retrieves relationships that target configuration items, as well as any types that derive from these configuration items.

The first command uses the **Get-SCOMClass** cmdlet to retrieve the class for an item by name, and stores the result in the variable named X.

The second command uses the `GetDerivedTypes` method and stores the result in the variable named Y.

The third command uses the **Get-SCOMRelationship** cmdlet to get the relationship for the derived types.

```
PS C:\> $X = Get-SCOMClass -Name "system.configitem"  
PS C:\> $Y = @($X; $X.GetDerivedTypes())  
PS C:\> Get-SCOMRelationship -Target $Y
```

## Related topics

[Get-SCOMClass](#)

[Get-SCOMManagementPack](#)

---

# 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 System Center 2012 – Operations Manager. These relationships describe the relationship of one class instance to another class instance.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-Id<Guid[]>**

Specifies an array of GUIDs 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 an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCManagementGroupConnection** cmdlet.

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

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

### **-SourceInstance<EnterpriseManagementObject[]>**

Specifies an array of instances that represent the source class of the relationships to retrieve. To obtain a class instance object, use the **Get-SCOMClassInstance** cmdlet. For more information type `Get-Help Get-SCOMClassInstance`.

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

### **-TargetInstance<EnterpriseManagementObject[]>**

Specifies an array of instances that represent the target class of the relationships to retrieve.

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: Retrieve all relationship instances

This example retrieves all relationship instances for which the source instance and the target instance are the Health Service class.

The first command uses the **Get-SCOMClassInstance** cmdlet to get the class instance. The output of the command in parentheses becomes input to the `Class` parameter. The cmdlet stores the result in the variable named `HealthService`.

The second command uses the **Get-SCOMRelationshipInstance** cmdlet to retrieve the relationship instance with the variable named `HealthService`.

```
PS C:\> $HealthService = Get-SCOMClassInstance -Class (Get-SCOMClass -Name
Microsoft.SystemCenter.HealthService)
PS C:\> Get-SCOMRelationshipInstance -SourceInstance $HealthService -TargetInstance
$HealthService | format-Table
```

## Related topics

[Get-SCOMClassInstance](#)

---

# Get-SCOMReportingSetting

---

## Get-SCOMReportingSetting

Retrieves 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 retrieves the reporting server settings for the management group.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you

---

specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Retrieve reporting server settings**

This command retrieves the reporting server settings for the management group.

```
PS C:\> Get-SCOMReportingSetting
```

---

## Related topics

[Set-SCOMReportingSetting](#)

---

# Get-SCOMResourcePool

---

## Get-SCOMResourcePool

Retrieves resource pools in Operations Manager.

### 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 retrieves resource pools in System Center 2012 – Operations Manager. A resource pool enables a collection of management servers to distribute the workload.

---

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names of an object. Values of the *DisplayName* parameter may vary depending on which localized management packs a user imports into the management group and the locale of the user who is running Windows PowerShell.

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

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

### **-EnableAutomaticMembership<Boolean>**

Indicates that the cmdlet returns only those 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[]>**

Specifies an array of GUIDs.

An object stores a GUID in the **Id** property for a resource pool. To obtain 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 array of objects to include in the resource pool.

---

Valid objects 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".

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

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

Specifies an array of names of an object.

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

### **-Observer<ComputerHealthService[]>**

Specifies an array of 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Retrieve all resource pools by name**

This command uses the **Get-SCOMResourcePool** cmdlet to get the resource pools with Management in their display name and resource pool names that begin with the string All.

```
PS C:\> Get-SCOMResourcePool -DisplayName "*Management*", "All"
```

### **Example 2: Retrieve resource pool by IDs**

This command uses the **Get-SCOMResourcePool** cmdlet to get the resource pools with the IDs 7413b06b-a95b-4ae3-98f2-dac9ff76dabd and 2ef74789-f9f5-46b0-af70-16d01d4f4577.

```
PS C:\> Get-SCOMResourcePool -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd, 2ef74789-f9f5-46b0-af70-16d01d4f4577
```

### **Example 3: Retrieve resource pools by member name**

This example retrieves resource pools by display name.

The first command uses the **Get-SCOMManagementServer** cmdlet to get the member with the display name Member01, and stores the result in the \$Member variable.

---

The second command uses the **Get-SCOMResourcePool** cmdlet to return all resource pools for which the value stored in the \$Member variable is a member.

```
PS C:\> $Member = Get-SCOMManagementServer -Name "Member01"  
PS C:\> Get-SCOMResourcePool -Member $Member
```

## Example 4: Retrieve resource pools by observer name

This example retrieves resource pools by display name.

The first command uses the **Get-SCOMManagementServer** cmdlet to get the observer with the display name Observer01, and stores the result in the \$Observer variable.

The second command uses the **Get-SCOMResourcePool** cmdlet to return all resource pools for which the value 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 hosts the root management server (RMS) Emulator role. The RMS Emulator gives legacy management packs that depend on RMS the ability to continue working in System Center 2012 – Operations Manager, which does not support a root management server.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

## **<CommonParameters>**

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

---

## Examples

### Example 1: Get management servers that have RMS Emulator role

This command establishes a temporary connection to the server named Server01.Contoso.com and gets management servers that host the RMS Emulator role.

```
PS C:\> Get-SCOMRMSEmulator -ComputerName "Server01.Contoso.com"
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Remove-SCOMRMSEmulator](#)

[Set-SCOMRMSEmulator](#)

---

# Get-SCOMRule

---

## Get-SCOMRule

Gets Operations Manager 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 monitoring rules for System Center 2012 – Operations Manager.

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

---

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names for objects. This cmdlet takes the strings in this array and looks for matches among the **DisplayName** properties of the objects that the cmdlet works with. *DisplayName* values depend on the localized, imported management packs that are part of the management group and the locale of the user who runs Windows PowerShell

If this parameter does not appear, the default description is the one in the *Name* parameter.

Aliases	none
Required?	true

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

### **-Id<Guid[]>**

Specifies an array of IDs for monitoring rules. The cmdlet gets monitoring rules that match the GUIDs.

An **SCOMGroup** object stores the GUID in its Id property. 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[]>**

Specifies a management pack object that stores overrides. To obtain a management pack object, use the **Get-SCOMManagementPack** cmdlet. If the rule is in an unsealed management pack, you must save the override into the same management pack.

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

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

Specifies an array of object names.

---

This parameter takes each string in the array and looks for matches among the Name properties of the objects that the cmdlet works with.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

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

Specifies an array of **ManagementPackClass** objects. To obtain a **ManagementPackClass** object, use the **Get-SCClass** cmdlet. The cmdlet gets monitoring rules for these classes.

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: Get monitoring rules by specifying names with wildcard

This command gets all monitoring rules that have health in their name.

```
PS C:\> Get-SCOMRule -Name "*health*"
```

### Example 2: Get monitoring rules by specifying display names with a wildcard

This command gets all monitoring rules with performance in their display name.

```
PS C:\> Get-SCOMRule -DisplayName "*performance*"
```

### Example 3: Get a monitoring rule by specifying a management pack

This command gets all management packs with System Center Core Monitoring in their display names. 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: Get monitoring rules by specifying target classes with a wildcard

This command gets the monitoring rules for each target class that has health in its display name.

```
PS C:\> Get-SCOMRule -Target (Get-SCOMClass -DisplayName "*health*")
```

### Example 5: Get monitoring rules by specifying a management pack with path:

This example gets monitoring rules for a management pack in a specified path.

---

The first command gets the path to a management pack file and stores it in the \$MPFile variable. The second command uses the *ManagementPack* parameter to get the monitoring rules from the management pack that is 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: Get a monitoring rule by specifying a GUID

This command gets the monitoring rule that has an ID of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\> Get-SCOMRule -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Related topics

[Disable-SCOMRule](#)

[Enable-SCOMRule](#)

[Get-SCOMClass](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMManagementPack](#)

---

# Get-SCOMRunAsAccount

---

## Get-SCOMRunAsAccount

Gets a Run As account for a management group.

### 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-SCOMRunAsAccount** cmdlet gets Run As accounts for a System Center 2012 – Operations Manager management group. A Run As account gives users the ability to specify the necessary permissions for use with rules, tasks, monitors, and discoveries targeted to specific computers on an as-needed basis.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, specify 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 the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type **Get-Help Get-Credential**.

This account must have access to the server that is specified in the *ComputerName* parameter if that parameter appears.

If you do not specify this parameter, the default is the account for the current user.

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

### **-Id<Guid[]>**

Specifies an array of account IDs.

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

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

Specifies an array of account names.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

## **<CommonParameters>**

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

## **Examples**

### **Example 1: Get all action accounts**

This command gets all action accounts.

```
PS C:\> Get-SCOMRunAsAccount
```

---

## Example 2: Get all data warehouse accounts

This command gets all accounts whose names begin with Data Warehouse.

```
PS C:\> Get-SCOMRunAsAccount -Name "Data Warehouse*"
```

### Related topics

[Add-SCOMRunAsAccount](#)

[Get-SCOMManagementGroupConnection](#)

[New-SCOMRunAsAccount](#)

[Remove-SCOMRunAsAccount](#)

[Update-SCOMRunAsAccount](#)

---

# 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 a System Center 2012 – Operations Manager Run As account. Distribution policies determine which computers receive a Run As account credential.

Due to the default formatting behavior of Windows PowerShell, the console does not always 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** property of the **SCOMRunAsDistribution** object that is stored in the variable.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-RunAsAccount<SecureData[]>**

Specifies an array of **SecureData** objects that represent Run As accounts. To obtain a **SecureData** object, use the **Get-SCOMRunAsAccount** cmdlet. This account cannot be part of a Run As profile.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a

---

temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

## <CommonParameters>

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

## Examples

### Example 1: Get the distribution policy for an account

This command displays the distribution policy for the `Contoso\Administrator` account.

```
PS C:\> Get-SCOMRunAsAccount "Contoso\Administrator" | Get-SCOMRunAsDistribution
```

### Example 2: Get systems approved for distribution in a distribution policy

This example displays the distribution policy for an account and lists the systems that are approved for distribution.

The first command gets the `Contoso\Administrator` account and passes that account to the **Get-SCOMRunAsDistribution** by using the pipeline operator. The command uses the pipeline operator to pass the distribution policy to the **Tee-Object** cmdlet, which displays it to the console, as well as saving it in the `$Distribution` variable. For more information, type `Get-Help Tee-Object`.

The second command displays the systems approved for distribution.

```
PS C:\> Get-SCOMRunAsAccount "Contoso\Administrator" | Get-SCOMRunAsDistribution | Tee-Object -Variable Distribution
PS C:\> $Distribution.SecureDistribution
```

---

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMRunAsAccount](#)

[Set-SCOMRunAsDistribution](#)

---

# Get-SCOMRunAsProfile

---

## Get-SCOMRunAsProfile

Gets 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 Run As profiles. A Run As profile is a group of associated Run As accounts that manages credentials and their distribution to different computers.

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

---

If you do not specify this parameter, the default is the computer for the current management group connection.

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 runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, or a user name such as User01, Domain01\User01, or User@Domain.com. If you type a user name, the cmdlet prompts you for a password. For more information about credential objects, type "Get-Help Get-Credential".

This account must have access to the server that is specified in the *ComputerName* parameter if that parameter appears.

If you do not specify this parameter, the default is the account for the current user.

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

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

Specifies an array of display names for Run As profiles. The cmdlet takes each string in the array and looks for matches among the display names of the Run As profiles that this cmdlet works with. Values for this parameter vary depending on which localized management packs are imported into the management group and the locale of the user that runs Windows PowerShell.

Aliases	none
Required?	true

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

### **-Id<Guid[]>**

Specifies an array of IDs of Run As profiles. The cmdlet gets Run As profiles that have these IDs.

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

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

Specifies an array of management pack objects. The cmdlet gets the Run As profiles for the management pack objects in the array. To obtain management pack objects, use 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 an array of names of Run As profiles. The cmdlet takes each string in the array and looks for matches among the names of the Run As profiles 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 an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type "Get-Help about\_OpsMgr\_Connections".

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

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -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: Get Run As profiles by using a name

This command gets the Run As profiles that have names that begin with Microsoft.SystemCenter.

```
PS C:\> Get-SCOMRunAsProfile -Name "Microsoft.SystemCenter*"
```

### Example 2: Get a Run As profile by using a display name

This command gets the Run As profile that has the display name Automatic Agent Management Account.

```
PS C:\> Get-SCOMRunAsProfile -DisplayName "Automatic Agent Management Account"
```

### Example 3: Get a Run As profile by using an ID

This command gets the Run As profile that has the Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\> Get-SCOMRunAsProfile -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

### Example 4: Get all Run As profiles for a management pack

This command gets the management pack named System Center Core Library and uses the pipeline operator to pass that management pack object to the **Get-SCOMRunAsProfile** cmdlet. The cmdlet gets all Run As profiles for the management pack.

```
PS C:\> Get-SCOMManagementPack -DisplayName "System Center Core Library" | Get-SCOMRunAsProfile
```

## Related topics

[Add-SCOMRunAsProfile](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMManagementPack](#)

[Get-SCOMRunAsProfile](#)

[Remove-SCOMRunAsProfile](#)

[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. Use this cmdlet to get tasks that have a specific name or ID as well as tasks that are associated with specified user roles, class instances, management packs, or target classes.

---

## Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type "Get-Help Get-Credential".

This account must have access to the server that is specified in the *ComputerName* parameter if that parameter appears.

If you do not specify this parameter, the default is the account for the current user.

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

---

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

Specifies an array of display names for task objects. This parameter takes one or more strings, and the cmdlet looks for matches among the display names of the task objects that the cmdlet works with.

Values for this parameter vary depending on which localized management packs are imported into the management group and the locale of the user that runs Windows PowerShell.

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

## **-Id<Guid[]>**

Specifies an array of task IDs. The cmdlet gets the tasks that have these IDs. To obtain a task, use the **Get-SCOMTask** cmdlet. The ID is the **ID** property of a task object.

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

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

Specifies an array of **EnterpriseManagementObject** objects that represent class instances. The cmdlet retrieves tasks for the class instances that the array stores. Specify a variable that represents the class instances or use a cmdlet such as **Get-SCOMClassInstance** that gets the class instances.

This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

Aliases	none
Required?	true
Position?	1

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

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

Specifies an array of management pack objects. The cmdlet gets tasks for the management pack objects in the array. Specify a variable that contains management pack objects, or use a cmdlet such as **Get-SCOMManagementPack** that gets management pack objects.

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

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

Specifies an array of task names.

The cmdlet takes each string in the array and matches it with the **Name** properties of the task 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 an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

---

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type "Get-Help about\_OpsMgr\_Connections".

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

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

Specifies an array of management pack class objects that represent target classes. Specify a variable that contains the target class objects, or use a cmdlet such as **Get-SCOMClass** that gets the target class objects.

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

### **<CommonParameters>**

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

## **Examples**

### **Example 1: Get tasks by specifying a partial name**

This command gets all tasks that have health in their names.

```
PS C:\> Get-SCOMTask -Name "*health*"
```

---

## Example 2: Get a task by using a partial display name

This command gets all tasks that have display names that begin with Stop.

```
PS C:\> Get-SCOMTask -DisplayName "Stop*"
```

## Example 3: Get a task by specifying a partial display name and an error action

This command gets all tasks for the management packs that have System Center Core in their display names. The command uses the **Get-SCOMManagementPack** cmdlet to get management packs based on display names, and passes them to the **Get-SCOMTask** cmdlet. Because the command specifies *SilentlyContinue* for the *ErrorAction* parameter, if the cmdlet finds a management pack with no associated tasks, it continues to run and does not display error messages.

```
PS C:\> Get-SCOMManagementPack -DisplayName "System Center Core*" | Get-SCOMTask -  
ErrorAction SilentlyContinue
```

## Example 4: Get tasks by using a partial class name

This command gets all tasks for classes that have health in their names.

```
PS C:\> Get-SCOMClass -Name "*health*" | Get-SCOMTask
```

## Example 5: Get tasks by specifying a partial class instance name

This command gets all tasks for class instances that have .Contoso.com in their names.

```
PS C:\> Get-SCOMClassInstance -Name "*.Contoso.com" | Get-SCOMTask
```

## Example 6: Get a task by specifying an ID

This command gets the task that has 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-SCOMManagementGroupConnection](#)

[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. Use this cmdlet to get results by task name or ID as well as for tasks that are associated with specified class instances or batches.

### Parameters

#### **-BatchID<Guid[]>**

Specifies an array that contains the IDs of batches in which tasks run. The cmdlet gets task results for each batch.

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

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

Specifies an array that contains the name of the computer with which to establish a connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, specify the computer name, **localhost**, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

This account must have access to the server that is specified in the *ComputerName* parameter if that parameter appears.

If you do not specify this parameter, the default is the account for the current user.

Aliases	none
Required?	false
Position?	named

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

### **-Id<Guid[]>**

Specifies an array of task IDs. The cmdlet gets the results of tasks that have these IDs.

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

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

Specifies an array of class instances for which to start a task. Specify a variable that stores class instances, or use a cmdlet such as **Get-SCOMClassInstance** that gets the class instances. This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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

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

Specifies an array of connections to management servers. To obtain management group connection objects, use the **Get-SCOMManagementGroupConnection** cmdlet. If this parameter does not appear, the default is the current management group connection.

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

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

Specifies an array of management pack tasks. The cmdlet gets results for the task objects that the array contains. Specify a variable that contains task objects or use a cmdlet such as **Get-SCOMTask** cmdlet that gets tasks.

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: Get task results by specifying a partial display name and an error action**

This command gets results for tasks that have SystemCenter in their names. The command uses the **Get-SCOMTask** cmdlet to get tasks based on name and passes them to the **Get-SCOMTaskResult** cmdlet by using the pipeline operator. Because the command specifies SilentlyContinue for the *ErrorAction* parameter, if the cmdlet finds a task with no matching results, the command continues to run and does not display error messages.

```
PS C:\> Get-SCOMTask -Name "*SystemCenter*" | Get-SCOMTaskResult -ErrorAction SilentlyContinue
```

---

## Example 2: Get task results by specifying class instances

This command gets all class instances in the Contoso.com domain and then returns the task results for each class instance object. The command uses the **Get-SCOMClassInstance** cmdlet to get all the instances that have a display name that contains .Contoso.com and passes them to the **Get-SCOMTaskResult** cmdlet by using the pipeline operator. Because the command specifies `SilentlyContinue` for the *ErrorAction* parameter, if the cmdlet finds a class instance with no matching task results, the command continues to run and does not display error messages.

```
PS C:\> Get-SCOMClassInstance -DisplayName "*.Contoso.com" | Get-SCOMTaskResult -ErrorAction SilentlyContinue
```

## Example 3: Get task results by specifying a task ID

This command gets the results of the task that has an ID of 7413b06b-a95b-4ae3-98f2-dac9ff76dabd.

```
PS C:\> Get-SCOMTaskResult -Id 7413b06b-a95b-4ae3-98f2-dac9ff76dabd
```

## Example 4: Get task results by specifying a batch ID

This command gets the results of a task that runs in a batch that has an ID of 2ef74789-f9f5-46b0-af70-16d01d4f4577.

```
PS C:\> Get-SCOMTaskResult -BatchId 2ef74789-f9f5-46b0-af70-16d01d4f4577
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[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. A tiered management group is part of a connected management group that has peer-to-peer connections between its members and that shares data in a single System Center 2012 – Operations Manager console.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

---

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of display names for connectors.

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

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

### **-Id<Guid[]>**

Specifies an array of connector IDs.

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

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

Specifies an array of connector names.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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: Get connectors for all tiered management groups**

This command gets all connectors for all tiered management groups. The command uses the **Get-SCOMTieredManagementGroup** cmdlet to get management groups and pass them to the **Get-SCOMTierConnector** cmdlet by using the pipeline operator.

```
PS C:\> Get-SCOMTieredManagementGroup | Get-SCOMTierConnector
```

### **Example 2: Get connectors for a tiered management group**

This command gets all connectors that have MyConnector in their names for the management group named ContosoTier. The command uses the **Get-SCOMTieredManagementGroup** cmdlet to get the management group named ContosoTier and passes that management group to the **Get-SCOMTierConnector** cmdlet by using the pipeline operator.

---

```
PS C:\> Get-SCOMTieredManagementGroup -Name "ContosoTier" | Get-SCOMTierConnector -Name  
"*MyConnector*"
```

## Related topics

[Add-SCOMTierConnector](#)

[Add-SCOMTieredManagementGroup](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMTierConnector](#)

[Get-SCOMTieredManagementGroup](#)

[Remove-SCOMTieredManagementGroup](#)

[Remove-SCOMTierConnector](#)

---

# 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 System Center 2012 – Operations Manager. A tiered management group is part of a connected management group that has peer-to-peer connections between its members and that shares data in a single Operations Manager console.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-Id<Guid[]>**

Specifies an array of IDs for tiered management groups.

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

---

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

Specifies an array of names for tiered management groups.

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

## **-OnlyForConnector**

Indicates that the cmdlet returns only tiered management groups that are available to connectors.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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: Get all tiered management groups**

This command gets all tiered management groups.

```
PS C:\> Get-SCOMTieredManagementGroup
```

---

## Example 2: Get a specific tiered management group

This command gets the tiered management group named Fabrikam.

```
PS C:\> Get-SCOMTieredManagementGroup -Name "Fabrikam"
```

## Example 3: Get tiered management groups that are available for connectors

This command gets only tiered management groups that are marked as available for connectors.

```
PS C:\> Get-SCOMTieredManagementGroup -OnlyForConnector
```

## Related topics

[Add-SCOMTieredManagementGroup](#)

[Get-SCOMManagementGroupConnection](#)

[Remove-SCOMTieredManagementGroup](#)

---

# 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: FromUserRoleId

```
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. A user role combines two elements:

- Profile. Defines the collection of operations to which the user role has access.
- Scope. Defines the boundaries for profile operations; for example, tasks and groups.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array that contains the display names of user role objects. The cmdlet takes each string in the array and looks for matches among the display names of the user role objects that this cmdlet works with. Values for this parameter vary depending on which localized management packs are imported into the management group and the locale of the user that runs Windows PowerShell.

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

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

### **-Id<Guid[]>**

Specifies an array of IDs for user role objects. The cmdlet takes each globally unique identifier (GUID) in the array and looks for matches with the **ID** properties of the user role objects that the cmdlet gets.

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

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

Specifies an array that contains the names of user role objects. The cmdlet takes each string in the array and looks for matches with the **Name** properties of the user role objects that the cmdlet gets.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

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

### <CommonParameters>

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

## Examples

### Example 1: Get user role information for a role

This command gets the information for the user role named OperationsManagerReadOnlyOperators.

```
PS C:\> Get-SCOMUserRole -Name "OperationsManagerReadOnlyOperators"
```

## Related topics

[Add-SCOMUserRole](#)

[Get-SCOMManagementGroupConnection](#)

[Set-SCOMUserRole](#)

---

# Get-SCOMWebAddressSetting

---

## Get-SCOMWebAddressSetting

Gets URLs for the web console and for online product knowledge.

### Syntax

Parameter Set: Empty

```
Get-SCOMWebAddressSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-SCOMWebAddressSetting** cmdlet gets the URLs of a System Center 2012 – Operations Manager management group for the web console and for online product knowledge.

The web console is a browser-based application that gives users the ability to monitor management group data from the Internet.

Product knowledge is the collection of notes that are attached to objects in a management pack. These notes document issues and suggested resolutions for those objects.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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

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

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

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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

---

## <CommonParameters>

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

## Examples

### Example 1: Get Web console and product knowledge URLs for a management group

This command gets the web console and online product knowledge URLs for the management group.

```
PS C:\> Get-SCOMWebAddressSetting
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Set-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 System Center 2012 – Operations Manager management packs. You can export a management pack to an XML-formatted file by using the **Export-SCOMManagementPack** cmdlet.

Before the cmdlet imports a management pack, System Center 2012 – Operations Manager attempts to validate the management pack. If the management pack contains invalid XML, the cmdlet does not import the management pack and raises an error.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 an array of full paths and file names of management packs to import. Include management packs (\*.xml, \*.mp) or management pack bundles (\*.mpb). If you specify a management pack bundle, the cmdlet imports all management packs from that bundle.

If you specify multiple management packs, the cmdlet imports them in the order that satisfies management pack dependencies. As a result, the cmdlet may import management packs in a different order from the one you specify.

Aliases	none
Required?	true
Position?	1
Default Value	none

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

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

Specifies an array of **ManagementPack** objects. To obtain a **ManagementPack** object, use the **Get-SCOMManagementPack** cmdlet.

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

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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 an array of **Connection** objects. To obtain a **Connection** object, use 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 object to the *ManagementPack* parameter of this cmdlet.

---

## Outputs

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

- **Microsoft.EnterpriseManagement.Configuration.ManagementPack** When you specify the *PassThru* parameter, this cmdlet returns an object that represents the management pack.

## Examples

### Example 1: Import a management pack from a file

This command imports the management pack file C:\MPArchive\MPWesternOffice.xml.

```
PS C:\> Import-SCOMManagementPack -Fullname "C:\MPArchive\MPWesternOffice.xml"
```

### Example 2: Import a management pack from a management pack bundle

This command imports only the Woodgrove.ManagementPack management pack that is included in the C:\MPArchive\MPWesternOfficeBundle.mpb management pack bundle file.

```
PS C:\> Import-SCOMManagementPack -Fullname "C:\MPArchive\MPWesternOfficeBundle.mpb" -Name Woodgrove.ManagementPack
```

## Related topics

[Export-SCOMManagementPack](#)

[Get-SCOMManagementPack](#)

[Remove-SCOMManagementPack](#)

[Get-SCOMManagementGroupConnection](#)

---

# Install-SCOMAgent

---

## Install-SCOMAgent

Deploys Operations Manager agents.

### 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 2012 – Operations Manager agents by using client push installation.

### Parameters

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

Specifies a **PSCredential** object. This parameter specifies the credentials that Operations Manager uses to run the deployment task. If you do not specify this parameter or you specify a null value, Operations Manager uses the default action account of the management server for the agent.

This parameter does not specify the action account that the agent uses after you install it on the computer. By default, the action account that the agent uses is set to Local System. You can use the **AgentActionAccount** parameter to change the action account that the agent uses.

To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

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

---

## **-AgentActionAccount<PSCredential>**

Specifies a **PSCredential** object. This parameter specifies the action account that the agent uses after it is installed on the computer. By default, the action account that the agent uses is set to Local System. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`. You can use the **Get-SCOMRunAsProfile** cmdlet to get a Run As accounts that you can use for the action account.

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**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

### **-PrimaryManagementServer<ManagementServer>**

Specifies a **ManagementServer** object. This parameter specifies the primary management server for the agent. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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

### **-Confirm**

Prompts you for confirmation before executing the command.

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

### **-WhatIf**

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

Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

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

## Examples

### Example 1: Install an agent on a server

This example installs an agent on a server and sets the primary management server for the agent.

The first command gets the management server object named MgmtServer01.Contoso.com and stores the object 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 \$PrimaryMgmtServer.

```
PS C:\> $PrimaryMgmtServer = Get-SCOMManagementServer -ComputerName
"MgmtServer01.Contoso.com"
PS C:\> Install-SCOMAgent -DNSHostName "server01.contoso.com" -PrimaryManagementServer
$PrimaryMgmtServer
```

### Example 2: Install an agent on a server by using the action account credentials

This example prompts the user to enter credentials and then uses the credentials to install an agent.

The first command prompts the user for credentials and store credentials in the \$InstallAccount variable.

The second command gets the management server object named MgmtServer01.Contoso.com and stores the object in the \$PrimaryMgmtServer variable.

The third command installs an agent on Server01.Contoso.com by using the credentials stored in \$InstallAccount, and sets the primary management server for the agent to the management server stored in \$PrimaryMgmtServer.

```
PS C:\> $InstallAccount = Get-Credential
PS C:\> $PrimaryMgmtServer = Get-SCOMManagementServer -ComputerName
"MgmtServer01.Contoso.com"
PS C:\> Install-SCOMAgent -DNSHostName "Server01.Contoso.com" -PrimaryManagementServer
$PrimaryMgmtServer -ActionAccount $InstallAccount
```

---

## Example 3: Install an agent on servers and set the agent action account

This example installs agents on three computers and sets the agent action account for the agents.

The first command stores the name of three computers in the \$Agents variable.

The second command prompts the user for credentials and stores the credentials 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 object named MgmtServer01.Contoso.com and stores the object in the \$PrimaryMgmtServer variable.

The fifth command installs an agent on the three computers stored in the \$Agents variable by using the credentials stored in \$InstallAccount. The command sets the primary management server for the agent to the management server stored in \$PrimaryMgmtServer, and sets the agent action account to the ActionAccount user stored in \$AgentActionAccount.

```
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 -ComputerName
```

```
MgmtServer01.contoso.com
```

```
PS C:\> $Install-SCOMAgent -DNSHostName $Agents -PrimaryManagementServer $PrimaryMgmtServer  
-ActionAccount $InstallAccount -AgentActionAccount $AgentActionAccount -Confirm
```

## Related topics

[Get-SCOMRunAsAccount](#)

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Uninstall-SCOMAgent](#)

[Repair-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 System Center 2012 – Operations Manager Customer Experience Improvement Program (CEIP) data collection on the local computer. CEIP collects usage and performance data from System Center 2012 – Operations Manager and System Center 2012 – Operations Manager cmdlets.

Clients send CEIP reports to a System Center 2012 – Operations Manager management server. Management servers are configured to forward these reports to Microsoft.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Enable CEIP data collection on a computer**

This command enables Operations Manager CEIP data collection on the local computer.

```
PS C:\> Join-SCOMCEIP
```

## **Related topics**

[Exit-SCOMCEIP](#)

[Test-SCOMCEIP](#)

---

# New-SCOMLocation

---

## New-SCOMLocation

Creates a location to which you can assign agent-managed computers, management servers, or resource pools.

### 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. You can associate agent-managed computers, management servers, or resource pools with a location by using the **Set-SCOMLocation** cmdlet. The Web Application Availability Monitoring Summary Map Dashboard displays the items that you associate with a location.

Provide a display name, a latitude, and a longitude for your location. Use the **Update-SCOMLocation** cmdlet to make changes to a location, or use the **Remove-SCOMLocation** to remove a location.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies a display name for the location.

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

### **-Latitude<String>**

Specifies a latitude value for the 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 a longitude value for the 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Create a location

This command creates a location that has a display name of Seattle, WA. The command specifies the latitude and longitude of Seattle.

```
PS C:\> New-SCOMLocation -DisplayName "Seattle, WA" -Latitude 47.6063889 -Longitude -  
122.330833
```

### Related topics

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Get-SCOMResourcePool](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMLocation](#)

[Remove-SCOMLocation](#)

[Set-SCOMLocation](#)

[Update-SCOMLocation](#)

---

# New-SCOMManagementGroupConnection

---

## New-SCOMManagementGroupConnection

Creates a persistent connection to a management group.

### Syntax

Parameter Set: FromComputerNames

```
New-SCOMManagementGroupConnection [[-ComputerName] <String[]> ] [[-Credential] <PSCredential> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The **New-SCOMManagementGroupConnection** cmdlet creates a persistent connection to a System Center 2012 – Operations Manager management group. A variety of cmdlets require a connection to a management group. By default, cmdlets use the current active persistent connection to a management group.

The most recent connection that this cmdlet creates becomes the active connection. Only one active persistent connection can exist at a time. If you attempt to create a connection that already exists, the cmdlet does not create another instance of the connection.

Specify a computer that belongs to the management group, along with required credentials. The System Center Data Access service must be running on the computer. You can use the **Get-SCOMManagementGroupConnection** cmdlet to get existing persistent connections, and you can use the **Set-SCOMManagementGroupConnection** cmdlet to make any persistent connection become the active connection. For more information about persistent connections, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes persistent connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.). The System Center Data Access service must be running on the computer.

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

Required?	false
Position?	1
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

Use an account that has access to the computer specified in the *ComputerName* parameter. The default is the current user.

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

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

---

## <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 this cmdlet.

## Outputs

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

- **Connection object** If you specify the *PassThru* parameter, this cmdlet generates a connection object.

## Examples

### Example 1: Create a connection

This command creates a persistent connection for the local computer. The command employs the **Get-Credential** cmdlet to create a **PSCredential** object for the user WOODGROVE\Administrator. That cmdlet prompts the user for a password.

```
PS C:\> New-SCOMManagementGroupConnection -ComputerName "localhost" -Credential (Get-Credential WOODGROVE\Administrator)
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Remove-SCOMManagementGroupConnection](#)

[Set-SCOMManagementGroupConnection](#)

---

# New-SCOMResourcePool

---

## New-SCOMResourcePool

Creates a resource pool in Operations Manager.

### 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 in System Center 2012 – Operations Manager. A resource pool enables a collection of management servers to distribute the workload.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 parameter accepts a maximum character length of 4,000 characters.

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

## **-DisplayName<String>**

Specifies a display name of an object. Values of the *DisplayName* parameter may vary depending on which localized management packs a user imports into the management group and the locale of the user who is 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 array of objects 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 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 an array of management servers or gateway management servers not currently in a 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Create a resource pool

This example creates a resource pool named Pool01 that contains all management servers.

The command in parentheses, which is executed first, uses the **Get-SCOMmanagementServer** cmdlet to get all management servers. The cmdlet then passes the results of the command in parentheses to the **New-SCOMResourcePool** cmdlet, which 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 you must add all future members manually.

```
PS C:\> New-SCOMResourcePool -DisplayName "Pool01" -Member (Get-SCOMManagementServer) -PassThru
```

## Related topics

[New-SCOMResourcePool](#)

[Remove-SCOMResourcePool](#)

[Set-SCOMResourcePool](#)

---

# Remove-SCOMADAgentAssignment

---

## Remove-SCOMADAgentAssignment

Removes AD DS agent assignments 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 removes Active Directory Domain Services (AD DS) agent assignments from the management group.

After you remove an agent assignment, the agent managed computers are not monitored by the management group. The state of these computers changes to critical, because the computers no longer send heartbeats to the management group. You can remove these computers from the management group and, if the computer is not assigned to other management groups, you can uninstall the System Center 2012 – Operations Manager agent.

### Parameters

#### **-AgentAssignment<AgentAssignment>**

Specifies an **AgentAssignment** object. To obtain an **AgentAssignment** object, use the **Get-SCOMADAgentAssignment** cmdlet.

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

---

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

## **-PrimaryServer<ManagementServer>**

Specifies a **ManagementServer** object. This parameter specifies the primary management server for the target agent-managed computers. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

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

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Remove an agent assignment by using the primary server

This command removes all AD DS agent assignments for a primary management server. The command uses the **Get-SCOMManagementServer** cmdlet to get the management server named OMServer01, and passes the result to the **Get-SCOMADAgentAssignment** cmdlet by using the pipeline operator.

The command gets all AD DS agent assignments that have the primary server named OMServer01, and passes the results to the **Remove-SCOMADAgentAssignment** cmdlet by using the pipeline operator. The command removes the agent assignments that have the primary server named OMServer01.

```
PS C:\> Get-SCOMManagementServer "OMServer01*" | Get-SCOMADAgentAssignment | Remove-SCOMADAgentAssignment
```

### Example 2: Remove an agent assignment by using the domain

This command gets all AD DS agent assignments for a domain. The command uses the **Get-SCOMADAgentAssignment** cmdlet to get the AD DS agent assignment named contoso.com, and passes the result to the **Remove-SCOMADAgentAssignment** cmdlet by using the pipeline operator. The command cmdlet removes the AD DS agent assignments for the domain named contoso.com.

```
PS C:\> Get-SCOMADAgentAssignment -Domain "contoso.com" | Remove-SCOMADAgentAssignment
```

---

## Related topics

[Get-SCOMADAgentAssignment](#)

[Add-SCOMADAgentAssignment](#)

[Update-SCOMADAgentAssignment](#)

---

# Remove-SCOMAgentlessManagedComputer

---

## Remove-SCOMAgentlessManagedComputer

Removes agentless managed computers from a management group.

### Syntax

Parameter Set: Default

```
Remove-SCOMAgentlessManagedComputer [-Computer] <RemotelyManagedComputer[]> [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Remove-SCOMAgentlessManagedComputer** cmdlet removes one or more agentless managed computers from a management group.

### Parameters

#### **-Computer<RemotelyManagedComputer[]>**

Specifies an array of agentless managed computers. You can use the **Get-SCOMAgentlessManagedComputer** cmdlet to get managed computers that do not have Operations Manager agents.

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: Remove agentless managed computers from the management group

This command removes managed computers that do not have Operations Manager agent. The command uses the **Get-SCOMAgentlesslyManagedComputer** cmdlet to get all agentless managed computers with names that begin with server01, and passes the results to the **Remove-SCOMAgentlessManagedComputer** cmdlet by using the pipeline operator. The command removes the agentless managed computers returned by the **Get-SCOMAgentlesslyManagedComputer** cmdlet from the management group.

```
PS C:\> Get-SCOMAgentlesslyManagedComputer -DNSHostName "server01*" | Remove-SCOMAgentlessManagedComputer -Confirm
```

---

## Related topics

[Get-SCOMAgentlessManagedComputer](#)

[Add-SCOMAgentlessManagedComputer](#)

[Set-SCOMAgentlessManagedComputer](#)

---

# 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** cmdlet removes a custom alert resolution state from the management group.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

## **-ResolutionState<MonitoringAlertResolutionState[]>**

Specifies a resolution state ID.

Operations Manager defines two resolution states: New (0) and Closed (255). You can assign custom resolution states any value from 2 through 254.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Remove a custom alert resolution state**

This example adds a new custom alert resolution state and then removes it.

---

The first command adds a custom alert resolution state named Investigating that has the resolution state code 10.

The second command gets the alert resolution state named Investigating and removes it.

```
PS C:\> Add-SCOMAlertResolutionState -Name "Investigating" -ResolutionStateCode 10
PS C:\> Get-SCOMAlertResolutionState -Name "Investigating" | Remove-
SCOMAlertResolutionState
```

## Related topics

[Get-SCOMAlertResolutionState](#)

[Remove-SCOMAlertResolutionState](#)

---

# Remove-SCOMConnector

---

## Remove-SCOMConnector

Removes connectors 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 one or more configuration item (CI) connectors from the management group. Removing a connector disables any transmission of information to that connector from System Center 2012 – Operations Manager.

### Parameters

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

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

---

## **-Connector<MonitoringConnector[]>**

Specifies an array of **MonitoringConnector** objects. To obtain a **MonitoringConnector** object, use the **Get-MonitoringConnector** cmdlet. For more information, type `Get-Help Get-SCOMConnector`.

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

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Remove a connector by using a name**

This command removes the connector named CustomConnector01.

```
PS C:\> Get-SCOMConnector -Connector "CustomConnector01" | Remove-SCOMConnector
```

---

# Remove-SCOMDisabledClassInstance

---

## Remove-SCOMDisabledClassInstance

Deletes class instances associated with disabled discovery.

### Syntax

Parameter Set: Empty

```
Remove-SCOMDisabledClassInstance [-ComputerName <String[]> ] [-Credential <PSCredential> ]  
[-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Remove-SCOMDisabledClassInstance** cmdlet deletes class instances for which you previously disabled discovery. The cmdlet also deletes all relationships that involve these class instances. You can disable discovery for an instance by using the **Disable-SCOMDiscovery** cmdlet.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

**Caution** This operation permanently deletes class instances and relationships. This operation may cause significant activity in the operational database.

### Parameters

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Remove class instances

This command removes all class instances class instances for which you previously disabled discovery.

```
PS C:\> Remove-SCOMDisabledClassInstance
```

### Related topics

[Disable-SCOMDiscovery](#)

[Get-SCOMManagementGroupConnection](#)

---

# Remove-SCOMLocation

---

## Remove-SCOMLocation

Removes associations with a location or deletes 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 the association of a location with an agent-managed computer, management server, or resource pool, or it deletes a location. To remove an association, specify agent-managed computers, management servers, or resource pools. To delete a location, specify the location by using the **Get-SCOMLocation** cmdlet.

You can change a location display name, a latitude, or a longitude by using the **Update-SCOMLocation** cmdlet.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

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

Specifies an array of **AgentManagedComputer** objects. To obtain an **AgentManagedComputer** object, use the **Get-SCOMAgent** cmdlet.

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

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

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

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

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

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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

### **-Location<EnterpriseManagementObject[]>**

Specifies an array of locations as **EnterpriseManagementObject** objects. To obtain a location, use the **Get-SCOMLocation** cmdlet.

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

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

Specifies an array of management server objects. To obtain a management server object, use the **Get-SCOMManagementServer** cmdlet.

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

---

## **-Pool<ManagementServicePool[]>**

Specifies an array of resource pool objects. To obtain a resource pool object, use the **Get-SCOMResourcePool** cmdlet.

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

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

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Delete a location

This example deletes a location that has the display name Seattle, WA. The first command gets the location object that has the display name Seattle, WA, and then stores the object in the \$Location variable.

The second command deletes the location object stored in the \$Location variable.

```
PS C:\> $Location = Get-SCOMLocation -DisplayName "Seattle, WA"
PS C:\> Remove-SCOMLocation -Location $Location
```

### Example 2: Remove an agent-managed computer from a location

This example removes an agent-managed computer named Server73.Contoso.com from its location. The first command gets the agent object for the computer named Server73.Contoso.com and then stores the object in the \$Agent variable.

The second command removes the location association from the agent-managed computer object stored in the \$Agent variable. This command does not delete the location itself.

```
PS C:\> $Agent = Get-SCOMAgent -Name "Server73.Contoso.com"
PS C:\> Remove-SCOMLocation -Agent $Agent
```

---

## Example 3: Remove a management server from a location

This example removes a management server from its location. The first command gets the management server object named MgmtServer01.Contoso.com, and then stores the object in the \$MgmtServer variable.

The second command removes the location association from the management server object stored in the \$MgmtServer variable. This command does not delete the location itself.

```
PS C:\> $MgmtServer = Get-SCOMManagementServer -Name "MgmtServer01.Contoso.com"
PS C:\> Remove-SCOMLocation -ManagementServer $MgmtServer
```

## Example 4: Remove a resource pool from a location

This example removes a resource pool named Notifications Resource Pool from its location. The first command gets the resource pool object for a resource pool named Notifications Resource Pool, and then stores the object in the \$Pool variable.

The second command removes the location association from the resource pool object stored in the \$Pool variable. This command does not delete the location itself.

```
PS C:\> $Pool = Get-SCOMResourcePool -Name "Notifications Resource Pool"
PS C:\> Remove-SCOMLocation -Pool $Pool
```

## Related topics

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Get-SCOMResourcePool](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMLocation](#)

[New-SCOMLocation](#)

[Set-SCOMLocation](#)

[Update-SCOMLocation](#)

---

# Remove-SCOMManagementGroupConnection

---

## Remove-SCOMManagementGroupConnection

Deletes persistent connections to management groups.

### Syntax

Parameter Set: FromConnection

```
Remove-SCOMManagementGroupConnection [-Connection] <Connection[]> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The **Remove-SCOMManagementGroupConnection** cmdlet deletes persistent connections to System Center 2012 – Operations Manager management groups. To obtain a connection, use the **Get-SCOMManagementGroupConnection** cmdlet. For more information about persistent connections, type `Get-Help about_OpsMgr_Connections`.

### Parameters

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

Specifies an array of **Connection** objects. To obtain a connection object, use the **Get-SCOMManagementGroupConnection** 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.SystemCenter.Core.Connection.Connection** You can pipe a management group connection to the *Connection* parameter of this cmdlet.

## Outputs

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

## Examples

### Example 1: Remove the active connection

This command deletes the currently active management group connection. The command uses the **Get-SCOMManagementGroupConnection** cmdlet to get all connections and passes them to a

---

conditional that drops any inactive connections from the pipeline. The command passes any active connection to the **Remove-SCOMManagementGroupConnection** cmdlet by using the pipeline operator.

```
PS C:\> Get-SCOMManagementGroupConnection |?{$_.IsActive } | Remove-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 System Center 2012 – Operations Manager management packs from a management group. When the cmdlet removes a management pack, it also removes all instances of types included in that management pack. You cannot remove a management pack on which other management packs depend.

### Parameters

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

Specifies an array of **ManagementPack** objects. To obtain a **ManagementPack** object, use 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 this 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: Remove management packs by using names

This example removes all management packs that have names that contain the string Woodgrove. The first command uses the **Get-SCOMManagementPack** cmdlet to get management packs, and then uses the pipeline operator to pass them to the **Where-Object** cmdlet. The **Where-Object** cmdlet drops all management packs that do not have names that contain Woodgrove. For more information, type `Get-Help Where-Object`. The command then stores any matching management packs in the \$MPS variable.

The second command removes the management packs that the objects stored in the \$MPS variable specify.

```
PS C:\> $MPS = Get-SCOMManagementPack | Where-Object { $_.name -match "Woodgrove" }
PS C:\> Remove-SCOMManagementPack -ManagementPack $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. Notification channels inform administrators of an alert, or they run automation in response to an alert. A notification channel uses a delivery mechanism in System Center 2012 – Operations Manager, such as email, instant message, Short Message Service, or command, to deliver notifications.

### Parameters

#### -Action<NotificationAction>

Specifies a notification action to take in the channel that you are removing. If you pipe input from the **Get-SCOMNotificationChannel** cmdlet, the cmdlet automatically populates this parameter.

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

---

## **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Endpoint<NotificationEndpoint>**

Specifies a notification endpoint in the channel. If you pipe input from the **Get-SCOMNotificationChannel** cmdlet, the cmdlet automatically populates this parameter.

Aliases	none
Required?	false
Position?	2
Default Value	none

Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Remove a notification channel

This command removes the notification channel named EmailMeWhenAnythingHappens.

```
PS C:\> Get-SCOMNotificationChannel "EmailMeWhenAnythingHappens" | Remove-SCOMNotificationChannel
```

## Related topics

[Add-SCOMNotificationChannel](#)

[Get-SCOMNotificationChannel](#)

---

# Remove-SCOMNotificationSubscriber

---

## Remove-SCOMNotificationSubscriber

Removes a notification subscriber.

### 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 in System Center 2012 – Operations Manager.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential".

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Recipient<NotificationRecipient[]>**

Specifies an array of notification subscribers to remove.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Remove a notification subscriber by using a name**

This command removes the notification subscriber named SarahJones.

```
PS C:\> Get-SCOMNotificationSubscriber -Name "SarahJones" | Remove-SCOMNotificationSubscriber
```

---

## Related topics

[Add-SCOMNotificationSubscriber](#)

[Get-SCOMNotificationSubscriber](#)

---

# Remove-SCOMNotificationSubscription

---

## Remove-SCOMNotificationSubscription

Removes notification subscriptions.

### Syntax

Parameter Set: FromNotificationSubscription

```
Remove-SCOMNotificationSubscription [-Subscription] <NotificationSubscription[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Remove-SCOMNotificationSubscription** cmdlet removes one or many notification subscriptions in System Center 2012 – Operations Manager.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCManagementGroupConnection** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Subscription<NotificationSubscription[]>**

Specifies an array of notification subscriptions 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: Remove a notification subscription**

This command removes the notification subscription for subscriber named Subscription01.

```
PS C:\> Get-SCOMNotificationSubscription -Name "Subscription01" | Remove-SCOMNotificationSubscription
```

---

## Related topics

[Add-SCOMNotificationSubscription](#)

[Disable-SCOMNotificationSubscription](#)

[Enable-SCOMNotificationSubscription](#)

[Remove-SCOMNotificationSubscription](#)

---

# Remove-SCOMResourcePool

---

## Remove-SCOMResourcePool

Removes one or more resource pools in Operations Manager.

### 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 in System Center 2012 – Operations Manager. A resource pool enables a collection of management servers to distribute the workload.

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 an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential".

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ResourcePool<ManagementServicePool[]>**

Specifies an array of resource pool objects. 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Remove a resource pool**

This command retrieves all resource pools that have a display name that begins with the string Pool01. The command uses the pipeline operator to pass the resource pool objects to the **Remove-SCOMResourcePool** cmdlet and 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 from a management group.

### Syntax

Parameter Set: Default

```
Remove-SCOMRMSEmulator [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Remove-SCOMRMSEmulator** cmdlet removes the root management server (RMS) Emulator role from a management group.

The RMS Emulator role gives legacy management packs that depend on an RMS the ability to continue working in System Center 2012 – Operations Manager, which does not support RMS. If you remove the RMS Emulator role, these management packs no longer work in System Center 2012 – Operations Manager.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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: Remove the RMS Emulator role for a management pack**

This command removes the RMS Emulator role from Server01.Contoso.com. The command prompts for confirmation before it performs the operation.

```
PS C:\> Remove-SCOMRMSEmulator -ComputerName "Server01.Contoso.com" -Confirm
```

## **Related topics**

[Get-SCOMManagementGroupConnection](#)

[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. A Run As account gives users the ability to specify permissions for rules, tasks, monitors, and discoveries that target specific computers on an as-needed basis.

The account cannot be part of any existing Run As profiles, or the command fails.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RunAsAccount<SecureData[]>**

Specifies an array of **SecureData** objects that represent Run As accounts. To obtain a **SecureData** object, use the **Get-SCOMRunAsAccount** cmdlet. This account cannot be part of a Run As profile.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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: Remove a Run As account

This command removes the Contoso\Administrator Run As account. The command uses the **Get-SCOMRunAsAccount** cmdlet to get the Run As account and passes it to the **Remove-SCOMRunAsAccount** cmdlet by using the pipeline operator.

```
PS C:\> Get-SCOMRunAsAccount -Name "Contoso\Administrator" | Remove-SCOMRunAsAccount
```

### Related topics

[Add-SCOMRunAsAccount](#)

[Get-SCOMManagementGroupConnection](#)

[Add-SCOMRunAsAccount](#)

[Get-SCOMRunAsAccount](#)

[Update-SCOMRunAsAccount](#)

---

# Remove-SCOMRunAsProfile

---

## Remove-SCOMRunAsProfile

Removes a Run As profile from a 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 a management group. A Run As profile is a group of associated Run As accounts that manage credentials and their distribution to different computers.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array that contains the name of the computer with which to establish a connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, specify the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type "Get-Help Get-Credential".

This account must have access to the server that is specified in the *ComputerName* parameter if that parameter appears.

If you do not specify this parameter, the default is the account for the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-RunAsProfile<ManagementPackSecureReference[]>**

Specifies an array that contains the Run As profile to remove.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter does not appear, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type "Get-Help about\_OpsMgr\_Connections".

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: Remove a Run As profile

This command removes the Profile01 profile. The command uses the **Get-SCOMRunAsProfile** cmdlet to get the specified profile and passes it to the **Remove-SCOMRunAsProfile** cmdlet by using the pipeline operator.

```
PS C:\> Get-SCOMRunAsProfile "Profile01" | Remove-SCOMRunAsProfile
```

### Related topics

[Add-SCOMRunAsProfile](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMRunAsProfile](#)

[Set-SCOMRunAsProfile](#)

---

# Remove-SCOMSubscriberSchedule

---

## Remove-SCOMSubscriberSchedule

Removes entries 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 entries from a notification subscriber schedule. Notification subscribers are users who receive notifications when System Center 2012 – Operations Manager raises an alert on a monitored system.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. This account must have access to the server that is specified in the *ComputerName* parameter if that parameter is used. For more information about credential objects, type "Get-Help Get-Credential".

If you do not specify this parameter, the default is the account for the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Entry<NotificationRecipientScheduleEntry>**

Specifies the schedule entry that the cmdlet removes.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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 an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type "Get-Help about\_OpsMgr\_Connections".

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Subscriber<NotificationRecipient>**

Specifies the notification subscriber for whom the cmdlet removes schedule entries.

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: Remove a schedule entry from a subscriber

This example removes the last schedule entry from a subscriber.

The first command gets the notification subscriber object named Katarina and stores the object in the \$Subscriber variable.

The second command selects the last schedule entry for the subscriber that is stored in the \$Subscriber variable. For more information, type "Get-Help Select-Object". The command passes the subscriber object to the **Remove-SCOMSubscriberSchedule** cmdlet by using the pipeline operator.

```
PS C:\> $Subscriber = Get-SCOMNotificationSubscriber "Katarina"
PS C:\> $Subscriber.ScheduleEntries | Select-Object -Last 1 | Remove-ScomSubscriberSchedule
-Subscriber $Subscriber
```

---

## Related topics

[Add-SCOMSubscriberSchedule](#)

[Clear-SCOMSubscriberSchedule](#)

[Get-SCOMManagementGroupConnection](#)

---

# Remove-SCOMTierConnector

---

## Remove-SCOMTierConnector

Removes 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 removes a connector from a tiered management group. A tiered management group is part of a connected management group that has peer-to-peer connections between its members and that shares data in a single System Center 2012 – Operations Manager console. Connectors communicate monitoring information between systems in a tiered management group. Removing a connector from a tier stops the tier from transmitting monitoring information to the connector.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array that contains the name of the computer with which to establish a connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, specify the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

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 the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies an array of management group connection objects that contains a connection to a management server. To get management group connection objects, specify the **Get-SCOMManagementGroupConnection** cmdlet. If you do not specify a value for this parameter, the default is the current management group connection.

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. To obtain a tiered management group object, use the **Get-SCOMTieredManagementGroup** 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: Remove connectors from a management group

This example clears all connectors from a tiered management group.

The first command uses the **Get-SCOMTieredManagementGroup** cmdlet to get tiered management groups from the server, and passes them to the **Select-Object** cmdlet by using the pipeline operator. The **Select-Object** cmdlet selects the first tiered management group object, and stores that object in the \$Tier variable. For more information, type `Get-Help Select-Object`.

The second command gets the connectors from the tiered management group in the \$Tier variable. It then passes these connectors to the **Remove-SCOMTierConnector** cmdlet by using the pipeline operator. The cmdlet removes the connectors from the tiered management group in \$Tier.

```
PS C:\> $Tier = Get-SCOMTieredManagementGroup | Select-Object -First 1
PS C:\> Get-SCOMTierConnector -Tier $Tier | Remove-SCOMTierConnector -Tier $Tier
```

## Related topics

[Add-SCOMTierConnector](#)

[Add-SCOMTieredManagementGroup](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMTierConnector](#)

[Get-SCOMTieredManagementGroup](#)

[Remove-SCOMTieredManagementGroup](#)

---

# 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. A tiered management group is part of a connected management group that has peer-to-peer connections between its members and that shares data in a single System Center 2012 – Operations Manager console.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Tier<TieredManagementGroup[]>**

Specifies an array of **TieredManagementGroup** objects that represent connections to management servers. Enter a management group connection object, such as one that the **Get-**

---

**SCOMManagementGroupConnection** cmdlet returns. The default is the current management group connection.

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: Remove all tiered management groups

This command removes all tiered management groups. The command uses the **Get-SCOMTieredManagementGroup** cmdlet to get all tiered management groups, and passes them to the **Remove-SCOMTieredManagementGroup** cmdlet by using the pipeline operator.

```
PS C:\> Get-SCOMTieredManagementGroup | Remove-SCOMTieredManagementGroup
```

### Related topics

[Add-SCOMTieredManagementGroup](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMTieredManagementGroup](#)

---

# Repair-SCOMAgent

---

## Repair-SCOMAgent

Repairs Operations Manager agents.

### Syntax

Parameter Set: FromAgent

```
Repair-SCOMAgent -Agent <AgentManagedComputer[]> [-Actionaccount <PSCredential> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The **Repair-SCOMAgent** cmdlet repairs one or more System Center 2012 – Operations Manager agent installations.

### Parameters

#### **-Actionaccount<PSCredential>**

Specifies a **PSCredential** object. This parameter specifies the credentials that Operations Manager uses to run the deployment task. If you do not specify this parameter or you specify a null value, Operations Manager uses the default action account of the management server for the agent.

To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Agent<AgentManagedComputer[]>**

Specifies an array of **AgentManagedComputer** objects. This parameter specifies the Operations Manager agents to repair. To obtain an **AgentManagedComputer** object, use the **Get-SCOMADAgent** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Repair an Operations Manager agent**

This command repairs an agent installation. The command uses the **Get-SCOMAgent** cmdlet to get the Operations Manager agent named server01.contoso.com, and passes the result to the **Repair-SCOMAgent** cmdlet by using the pipe operator. The second command repairs the agent named server01.contoso.com.

---

PS C:\> Get-SCOMAgent -DNSHostName "server01.contoso.com" | Repair-SCOMAgent

## Related topics

[Get-SCOMRunAsAccount](#)

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Uninstall-SCOMAgent](#)

[Install-SCOMAgent](#)

---

# Resolve-SCOMAlert

---

## Resolve-SCOMAlert

Resolves an alert.

### 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 System Center 2012 – Operations Manager. The cmdlet sets the **ResolutionState** property of an alert to 255 (Closed). You can also resolve an alert by using the **Set-SCOMAlert** cmdlet and setting the *ResolutionState* parameter to 255 (Closed).

### Parameters

#### **-Alert<MonitoringAlert[]>**

Specifies an array of **MonitoringAlert** objects. To obtain a **MonitoringAlert** object, use the **Get-SCOMAlert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-Comment<String>**

Specifies a comment to add to the resolved alert.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-Connector<MonitoringConnector>**

Specifies a **MonitoringConnector** object. This parameter specifies the connector that generated the connection string. To obtain a **MonitoringConnector** object, use the **Get-SCOMConnector** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField1<String>**

Specifies information to add to the **CustomField1** property of the alert.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-CustomField10<String>**

Specifies information to add to the **CustomField10** property of the alert.

Aliases	none
Required?	false
Position?	11
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField2<String>**

Specifies information to add to the **CustomField2** property of the alert.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField3<String>**

Specifies information to add to the **CustomField3** property of the alert.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-CustomField4<String>**

Specifies information to add to the **CustomField4** property of the alert.

Aliases	none
Required?	false
Position?	5
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField5<String>**

Specifies information to add to the **CustomField5** property of the alert.

Aliases	none
Required?	false
Position?	6
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField6<String>**

Specifies information to add to the **CustomField6** property of the alert.

Aliases	none
Required?	false
Position?	7
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-CustomField7<String>**

Specifies information to add to the **CustomField7** property of the alert.

Aliases	none
Required?	false
Position?	8
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField8<String>**

Specifies information to add to the **CustomField8** property of the alert.

Aliases	none
Required?	false
Position?	9
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField9<String>**

Specifies information to add to the **CustomField9** property of the alert.

Aliases	none
Required?	false
Position?	10
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-Owner<String>**

Specifies the user name of the owner of the alert, in the format Domain\Account.

Aliases	none
Required?	false
Position?	13
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

## **-TicketId<String>**

Specifies the ticket ID of the alert.

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: Resolve error alerts by using the alert severity

This command resolves all Error alerts and adds a comment to the resolved alerts.

```
PS C:\> Get-SCOMAlert -Severity 2 | Resolve-SCOMAlert -Comment "All alerts are resolved."
```

## Related topics

[Get-SCOMAlert](#)

[Set-SCOMAlert](#)

---

# Set-SCOMAgentApprovalSetting

---

## Set-SCOMAgentApprovalSetting

Changes 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 changes the manual agent approval setting for the management group.

Specify the *AutoApprove* parameter to automatically approve any new manually installed agents.

Specify the *AutoReject* parameter to automatically reject any new manually installed agents.

Specify the *Pending* parameter to review the request from any new manually installed agents.

### Parameters

#### -AutoApprove

Indicates that Operations Manager automatically approves any manually installed agent that contacts the management server and joins the agent to the management group.

Aliases	none
Required?	true
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AutoReject**

Indicates that Operations Manager automatically rejects any manually installed agent that contacts the management server and does not join the agent to the management group.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

---

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

### **-Pending**

Indicates that Operations Manager directs all requests from manually installed agents that contact the management server to the Pending Management list. An administrator must review the request and manually approve the agent requests.

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 an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **Example 1: Change the agent approval setting to AutoApprove**

This command sets the manual agent approval setting for the management group to automatically approve any manually installed agent that contacts the management server and join the agent to the management group.

```
PS C:\> Set-SCOMAgentApprovalSetting -AutoApprove
```

### **Example 2: Change the agent approval setting to AutoReject**

This command changes the manual agent approval setting for the management group to automatically reject any manually installed agent that contacts the management server.

```
PS C:\> Set-SCOMAgentApprovalSetting -AutoReject
```

---

## Example 3: Change the agent approval setting to Pending

This command changes the manual agent approval setting for the management group to pending. An administrator must review the requests and manually approve the agent requests.

```
PS C:\> Set-SCOMAgentApprovalSetting -Pending
```

### Related topics

[Get-SCOMAgentApprovalSetting](#)

[Get-SCOMPendingManagement](#)

[Approve-SCOMPendingManagement](#)

[Deny-SCOMPendingManagement](#)

---

# Set-SCOMAgentlessManagedComputer

---

## Set-SCOMAgentlessManagedComputer

Changes the settings for agentless managed computers.

### 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 changes settings for agentless managed computers.

### Parameters

#### **-Computer<RemotelyManagedComputer[]>**

Specifies an array of agentless managed computers. You can use the **Get-SCOMAgentlessManagedComputer** cmdlet to get managed computers that do not have agents.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-ManagedByAgent<AgentManagedComputer>**

Specifies an **AgentManagedComputer** object. This parameter specifies the Operations Manager agent that performs agentless monitoring. The action account of the agent that performs the monitoring must have local administrative rights on the computer that it monitors.

To obtain an **AgentManagedComputer** object, use the **Get-SCOMAgent** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ManagedByManagementServer<ManagementServer>**

Specifies a **ManagementServer** object. This parameter specifies the primary management server that performs agentless monitoring of the agentless managed computers. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Set the Operations Manager agent to perform agentless monitoring

This command gets the agentless managed computer named server02 and passes it to the **Set-SCOMAgentlessManagedComputer** cmdlet by using a pipe operator. The command sets the agent-managed computer named OMAgent01 as the Operations Manager agent that performs agentless monitoring for server02.

```
PS C:\> Get-SCOMAgentlessManagedComputer -DNSHostName "server02.contoso.com" | Set-SCOMAgentlessManagedComputer -ManagedByAgent (Get-SCOMAgent -DNSHostName "OMAgent01.contoso.com") -PassThru
```

## Related topics

[Add-SCOMAgentlessManagedComputer](#)

[Set-SCOMAgentlessManagedComputer](#)

[Get-SCOMAgentlessManagedComputer](#)

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

---

# Set-SCOMAlert

---

## Set-SCOMAlert

Changes the properties of alerts.

### 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 one or more alerts. You can use the cmdlet to resolve an alert by setting the *ResolutionState* parameter to 255 (Closed).

### Parameters

#### **-Alert<MonitoringAlert[]>**

Specifies an array of **MonitoringAlert** objects. To obtain a **MonitoringAlert** object, use the **Get-SCOMAlert** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-Comment<String>**

Specifies a comment to add to the resolved alert.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-Connector<MonitoringConnector>**

Specifies a **MonitoringConnector** object. This parameter specifies the connector that generated the connection string. To obtain a **MonitoringConnector** object, use the **Get-SCOMConnector** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField1<String>**

Specifies information to add to the **CustomField1** property of the alert.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-CustomField10<String>**

Specifies information to add to the **CustomField10** property of the alert.

Aliases	none
Required?	false
Position?	11
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField2<String>**

Specifies information to add to the **CustomField2** property of the alert.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField3<String>**

Specifies information to add to the **CustomField3** property of the alert.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-CustomField4<String>**

Specifies information to add to the **CustomField4** property of the alert.

Aliases	none
Required?	false
Position?	5
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField5<String>**

Specifies information to add to the **CustomField5** property of the alert.

Aliases	none
Required?	false
Position?	6
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

## **-CustomField6<String>**

Specifies information to add to the **CustomField6** property of the alert.

Aliases	none
Required?	false
Position?	7
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

### **-CustomField7<String>**

Specifies information to add to the **CustomField7** property of the alert.

Aliases	none
Required?	false
Position?	8
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField8<String>**

Specifies information to add to the **CustomField8** property of the alert.

Aliases	none
Required?	false
Position?	9
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-CustomField9<String>**

Specifies information to add to the **CustomField9** property of the alert.

Aliases	none
Required?	false
Position?	10
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-Owner<String>**

Specifies the user name of 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**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## **-ResolutionState<Byte>**

Specifies a resolution state Id.

When Operations Manager generates an alert, its resolution state is New. You can change the resolution state for a new alert to Closed or to a custom resolution state that an administrator has created for the management group. The ID for New is 0 and the ID for Closed is 255. You can assign custom resolution states any value from 2 through 254.

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: Close alerts by using the resolution state**

This command gets all alerts with a resolution state of 15 and then passes the alert objects to the **Set-SCOMAlert** cmdlet by using the pipe operator. The **Set-SCOMAlert** cmdlet closes the alert by setting the resolution state to 255.

```
PS C:\> Get-SCOMAlert -ResolutionState 15 | Set-SCOMAlert -ResolutionState 255
```

### **Example 2: Change the properties of alerts**

This command gets all alerts named "Failed Accessing Windows Event Log" and then passes the alert objects to the **Set-SCOMAlert** cmdlet by using the pipe operator. The **Set-SCOMAlert** cmdlet changes the owner of the alert 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](#)

[Resolve-SCOMAlert](#)

[Get-SCOMConnector](#)

---

# Set-SCOMAlertResolutionSetting

---

## Set-SCOMAlertResolutionSetting

Changes 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 changes the alert automatic resolution settings for the management group. You can configure monitors that create alerts to automatically resolve the alert when the monitor returns to a healthy state. This means that any unresolved alert for the monitor represents a problem that still exists.

### Parameters

#### **-AlertAutoResolveDays<Int32>**

Specifies the number of days until Operations Manager automatically resolves active alerts that have a monitoring state of New (0).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

---

## **-ComputerName<String[]>**

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-HealthyAlertAutoResolveDays<Int32>**

Specifies the number of days after the alert source for active alerts is healthy that Operations Manager automatically resolves the alerts.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Change the automatic resolution setting for alerts**

This command changes the alert resolution setting to automatically resolve active alerts in 10 days, and to automatically resolve active alerts after their alert source is healthy for 5 days.

```
PS C:\> Set-SCOMAlertResolutionSetting -AlertAutoResolveDays 10 -HealthyAlertAutoResolveDays 5
```

## **Related topics**

[Get-SCOMAlertResolutionSetting](#)

---

# Set-SCOMDatabaseGroomingSetting

---

## Set-SCOMDatabaseGroomingSetting

Modifies database grooming settings for a 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 modifies database grooming settings for a management group. Database grooming automatically removes unnecessary data from the System Center 2012 – Operations Manager database in order to maintain performance.

You can specify how long, in days, to keep the following items:

- Resolved alerts
- Availability history
- Event data
- Task history
- Maintenance mode history
- Monitoring job data
- Performance data
- State change data

Use the **Get-SCOMDatabaseGroomingSetting** cmdlet to see the current values for these settings.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

### **-AlertDaysToKeep<Byte>**

Specifies the number of days to keep resolved alerts.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-AvailabilityHistoryDaysToKeep<Byte>**

Specifies the number of days to keep availability history.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-EventDaysToKeep<Byte>**

Specifies the number of days to keep event data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-JobStatusDaysToKeep<Byte>**

Specifies the number of days to keep task history.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-MaintenanceModeHistoryDaysToKeep<Byte>**

Specifies the number of days to keep maintenance mode history.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-MonitoringJobDaysToKeep<Byte>**

Specifies the number of days to keep monitoring job data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

### **-PerformanceDataDaysToKeep<Byte>**

Specifies the number of days to keep performance data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-StateChangeEventDaysToKeep<Byte>**

Specifies the number of days to keep 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: Set days to keep resolved alerts

This command sets the number of days to keep resolved alerts to 21.

```
PS C:\> Set-SCOMDatabaseGroomingSetting -AlertDaysToKeep 21
```

### Example 2: Set days to keep availability history

This command sets the number of days to keep availability history to 10.

```
PS C:\> Set-SCOMDatabaseGroomingSetting -AvailabilityHistoryDaysToKeep 10
```

## Related topics

[Get-SCOMDatabaseGroomingSetting](#)

[Get-SCOMManagementGroupConnection](#)

---

# Set-SCOMDataWarehouseSetting

---

## Set-SCOMDataWarehouseSetting

Modifies data warehouse settings for a 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 modifies data warehouse settings for a management group. You can modify the database name or the server name. Use the **Get-SCOMDataWarehouseSetting** cmdlet to see the current settings.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DatabaseName<String>**

Specifies the name of a data warehouse database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ServerName<String>**

Specifies the name of a data warehouse server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Modify data warehouse settings**

This command modifies the settings for the name of the data warehouse database and the name of the data warehouse server.

```
PS C:\> Set-SCOMDataWarehouseSetting -DatabaseName "SCOMDW" -ServerName  
"SCOMServer\INSTANCE1"
```

## **Related topics**

[Get-SCOMDataWarehouseSetting](#)

[Get-SCOMManagementGroupConnection](#)

---

# Set-SCOMErrorReportingSetting

---

## Set-SCOMErrorReportingSetting

Modifies the error reporting setting for a 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 modifies the System Center 2012 – Operations Manager error reporting setting for a management group. You can see the current setting by using the **Get-SCOMErrorReportingSetting** cmdlet.

Specify one of the following parameters to choose your reporting setting:

- *AutomaticallySend*. Send error reports.
- *DoNotSend*. Do not send error reports.
- *PromptBeforeSending*. Prompt user before sending.

To opt out of error reporting, use the cmdlet with none of these parameters.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

### **-AutomaticallySend**

Indicates that Operations Manager sends error reports.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DoNotSend**

Indicates that Operations Manager does not send error reports.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

### **-PromptBeforeSending**

Indicates that Operations Manager prompts for approval before it sends error reports.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Change setting to always send error reports

This command changes the error reporting setting for the current management group to always send error reports.

```
PS C:\> Set-SCOMErrorReportingSetting -AutomaticallySend
```

## Related topics

[Get-SCOMErrorReportingSetting](#)

[Get-SCOMManagementGroupConnection](#)

---

# Set-SCOMHeartbeatSetting

---

## Set-SCOMHeartbeatSetting

Modifies the agent and server heartbeat settings for a management group.

### Syntax

Parameter Set: Empty

```
Set-SCOMHeartbeatSetting [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-HeartbeatInterval <TimeSpan> ] [-MissingHeartbeatThreshold <Int32> ] [-PassThru] [-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Set-SCOMHeartbeatSetting** cmdlet modifies the agent and server heartbeat settings for an System Center 2012 – Operations Manager management group. You can set the heartbeat interval and the missing heartbeat threshold.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-HeartbeatInterval<TimeSpan>**

Specifies an interval as a **TimeSpan** object. To obtain a **TimeSpan** object, use the **Get-TimeSpan** cmdlet. For more information, type `Get-Help Get-TimeSpan`. This interval is the number of minutes between heartbeats.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-MissingHeartbeatThreshold<Int32>**

Specifies an integer threshold. A management server ignores this many missing heartbeats before it raises an alert.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Modify the missing heartbeat threshold

This command sets the missing heartbeat threshold to five.

```
PS C:\> Set-SCOMHeartbeatSetting -MissingHeartbeatThreshold 5
```

### Example 2: Modify the heartbeat interval

This command sets the heartbeat interval to 1 minute.

```
PS C:\> Set-SCOMHeartbeatSetting -HeartbeatInterval "0:01:00"
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMHeartbeatSetting](#)

---

# Set-SCOMLicense

---

## Set-SCOMLicense

Sets the product license level and removes evaluation timeout.

### 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 System Center 2012 – Operations Manager product license level and removes the evaluation expiration timeout. To view the current license terms, use the **Get-SCOMLicense** cmdlet. You can also view the license information in the Operations Console Help About dialog box.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

After you enter a license key, restart the System Center Data Access Service on all of your Operations Manager management servers.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Set a product license

This command sets the product license level for the product with the specified ID.

```
PS C:\> Set-SCOMLicense -ProductId 'C97A1C5E-6429-4F71-8B2D-3525E237BF62'
```

### Related topics

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMLicense](#)

---

# Set-SCOMLocation

---

## Set-SCOMLocation

Associates agent-managed computers, management servers, or resource pools 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 agent-managed computers, management servers, or resource pools with a location. This cmdlet removes a current association, if one exists. The Web Application Availability Monitoring Summary Map Dashboard displays state information for agents, management servers, and resource pools associated with a location.

You can remove an association by using the **Remove-SCOMLocation** cmdlet. You can create a location by using the **New-SCOMLocation** cmdlet.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

---

## Parameters

### **-Agent<AgentManagedComputer[]>**

Specifies an array of **AgentManagedComputer** objects. To obtain an **AgentManagedComputer** object, use the **Get-SCOMAgent** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Location<EnterpriseManagementObject>**

Specifies a location as an **EnterpriseManagementObject** object. To obtain a location, use the **Get-SCOMLocation** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-ManagementServer<ManagementServer[]>**

Specifies an array of management server objects. To obtain a management server object, use the **Get-SCOMManagementServer** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## **-Pool<ManagementServicePool[]>**

Specifies an array of resource pool objects. To obtain a resource pool object, use the **Get-SCOMResourcePool** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Associate an agent with a location

This example associates an agent-managed computer named Sever01.Contoso.com with the location Seattle, WA. The first command gets the location object that has the display name of Seattle, WA, and then stores the object in the \$Location variable.

The second command gets the agent object named Server01.Contoso.com, and then stores the object in the \$Agent variable.

The third command associates the location stored in the \$Location variable with the agent-managed computer stored in the \$Agent variable. If you previously associated this agent-managed computer with a location, the command removes that association.

```
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: Associate a management server with a location

This example associates a management server named MgmtServer01.Contoso.com with the location New York, NY. The first command gets the location object that has the display name of New York, NY, and then stores the object in the \$Location variable.

The second command gets the management server object named MgmtServer01.Contoso.com, and stores the object in the \$MgmtServer variable.

The third command associates the location stored in the \$Location variable with the management server stored in the \$MgmtServer variable. If you previously associated this management server with a location, the command removes that association.

```
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: Associate a resource pool to a location

This example associates a resource pool named Notifications Resource Pool with the location Paris, FR. The first command gets the location object that has the display name Paris, FR, and then stores the object in the \$Location variable.

---

The second command gets the resource pool named Notifications Resource Pool, and then stores the object in the \$Pool variable.

The third command associates the location stored in the \$Location variable with the resource pool stored in the \$Pool variable. If you previously associated the resource pool with a location, the command removes that association.

```
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-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Get-SCOMResourcePool](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMLocation](#)

[New-SCOMLocation](#)

[Remove-SCOMLocation](#)

[Update-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. You can use this cmdlet to update only active entries.

When a resource is in maintenance mode, System Center 2012 – Operations Manager suppresses alerts, notifications, rules, monitors, automatic responses, state changes, and new alerts.

You can change the comment or reason for the maintenance mode. Use the **Get-SCOMMaintenanceMode** cmdlet to get a maintenance mode entry to update. Specify an end for the maintenance window.

### Parameters

#### **-Comment<String>**

Specifies a comment for the maintenance mode entry.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-EndTime<DateTime>**

Specifies when maintenance mode ends as a **DateTime** object. A resource cannot be in maintenance mode for fewer than five minutes. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-MaintenanceModeEntry<MaintenanceWindow[]>**

Specifies an array of **MaintenanceWindow** objects. To obtain a **MaintenanceWindow** object, use the **Get-SCOMMaintenanceMode** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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 a reason for maintenance mode. The acceptable values for this parameter 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: Update active maintenance mode entries for resources in a domain

This example updates all active maintenance mode entries for a specified domain. The first command creates a **DateTime** object for one day in the future and then stores it 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-SCOMMaintenanceMode** cmdlet, which gets maintenance mode entry objects. The command uses the pipeline operator to pass these objects to the **Set-SCOMMaintenanceMode** cmdlet. This cmdlet updates the end time for each object to the **DateTime** object stored in the `$NewEndTime` variable. The command also includes a comment for each updated maintenance mode entry.

```
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: Update maintenance mode entry for a specified resource

This example extends maintenance mode for a specified server. The first command gets the class instance named Server01.Contoso.com and then 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 creates a **DateTime** object for 30 minutes in the future and then stores it in the `$NewEndTime` variable.

The fourth command updates the maintenance mode session for the maintenance mode entry stored in the `$MMEEntry` variable to the **DateTime** object stored in the `$NewEndTime` variable and adds a comment.

```
PS C:\> $Instance = Get-SCOMClassInstance -Name "Server01.Contoso.com"
PS C:\> $MMEEntry = Get-SCOMMaintenanceMode -Instance $Instance
PS C:\> $NewEndTime = (Get-Date).addMinutes(30)
PS C:\> Set-SCOMMaintenanceMode -MaintenanceModeEntry $MMEEntry -EndTime $NewEndTime -Comment "Adding 30 minutes to the end time."
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMMaintenanceMode](#)

[Start-SCOMMaintenanceMode](#)

---

# Set-SCOMManagementGroupConnection

---

## Set-SCOMManagementGroupConnection

Makes a management group connection become the active connection.

### Syntax

Parameter Set: FromConnection

```
Set-SCOMManagementGroupConnection [-Connection] <Connection> [ <CommonParameters>]
```

### Detailed Description

The **Set-SCOMManagementGroupConnection** cmdlet makes a persistent connection for an System Center 2012 – Operations Manager management group become the active connection. Only one active persistent connection can exist at a time. When you create a connection by using the **New-SCOMManagementGroupConnection** cmdlet, that connection becomes the active connection.

To determine the current active connection, use the **Get-SCOMManagementGroupConnection** cmdlet to get connection objects. A connection object includes the **IsActive** property. If this value is \$True, the connection is the active management group connection.

For more information about persistent connections, type `Get-Help about_OpsMgr_Connections`.

### Parameters

#### **-Connection<Connection>**

Specifies the connection to become the active 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 this 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: Make a connection active

This command uses the **Get-SCOMManagementGroupConnection** cmdlet to get all the management group connections. The command passes connections to the **Set-SCOMManagementGroupConnection** cmdlet by using the pipeline operator. The command makes each connection passed to it become the active connection.

```
PS C:\> Get-SCOMManagementGroupConnection | Set-SCOMManagementGroupConnection
```

## Related topics

[Get-SCOMManagementGroupConnection](#)

[New-SCOMManagementGroupConnection](#)

[Remove-SCOMManagementGroupConnection](#)

---

# Set-SCOMPparentManagementServer

---

## Set-SCOMPparentManagementServer

Modifies 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 modifies the primary and failover management servers for an agent or gateway management server. In System Center 2012 – Operations Manager, primary and failover management servers provide redundancy for agents and gateway management servers.

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 gateway management server object, type `"Get-Help Get-SCOMGatewayManagementServer"`.

### Parameters

#### **-Agent<AgentManagedComputer[]>**

Specifies an array of agent objects. To obtain an **AgentManagedComputer** object, use the **Get-SCOMADAgent** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

### **-FailoverServer<ManagementServer[]>**

Specifies an array of management server objects for the agent to use as 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 an array of gateway management server objects. For information about how to get a management server object, type "Get-Help Get-SCOMGatewayManagementServer".

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue, ByPropertyName)
Accept Wildcard Characters?	false

---

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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 an array of management server object for the agent to use as its primary 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: Set primary and failover management servers

This example sets the primary and failover parent management servers.

The first two commands use the **Get-SCManagementServer** cmdlet to 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 uses the **Get-SCAgent** cmdlet to get the agent named Server01.Contoso.com and passes the result to the **Set-SCOMPparentManagementServer** cmdlet by using the pipeline operator. The command 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: Set gateway primary server and failover server

This example sets the gateway, primary, and failover parent management servers.

The first two commands use the **Get-SCManagementServer** cmdlet to 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 uses the **Get-SCOMGatewayManagementServer** cmdlet to get the gateway management server named GatewayMgmtServer01.Contoso.com and pipes the result to the **Set-SCOMPARENTManagementServer** cmdlet to set 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-SCOMGatewayManagementServer | Set-
SCOMPARENTManagementServer -PrimaryServer $PrimaryMgmtServer -FailoverServer
$FailoverMgmtServer
```

### Example 3: Set primary management server

This example sets the primary management server 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 the **Set-SCOMPARENTManagementServer** cmdlet, which 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")
```

### Related topics

[Get-SCOMPARENTManagementServer](#)

---

# Set-SCOMReportingSetting

---

## Set-SCOMReportingSetting

Changes the URL for a 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 changes the URL for a reporting server for the management group.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential". If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

## **-ReportingServerUrl<Uri>**

Specifies a URL that identifies a reporting server.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use 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: Change the reporting server URL

This command changes the reporting server URL for the management group to `http://Reporting.Contoso.com`.

```
PS C:\> Set-SCOMReportingSetting -ReportingServerUrl "http://Reporting.Contoso.com"
```

## Related topics

[Get-SCOMReportingSetting](#)

---

# Set-SCOMResourcePool

---

## Set-SCOMResourcePool

Changes the properties of a resource pool in Operations Manager.

### 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 resource pool in System Center 2012 – Operations Manager. A resource pool enables a collection of management servers to distribute the workload.

### Parameters

#### **-Action<UpdateAction>**

Specifies an action to take. When updating the members of a resource pool, specify either Add or Remove.

Aliases	none
Required?	true
Position?	3
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type "Get-Help Get-Credential".

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-EnableAutomaticMembership<Boolean>**

Indicates that the cmdlet 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 you change the value from \$True to \$False, the membership of the pool remains the same until you manually update it.

If the value is changed from \$False to \$True, all members of the pool are removed, and then the pool is automatically repopulated 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 array of objects to include in the resource pool.

Valid 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".

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-Observer<ComputerHealthService[]>**

Specifies an array of management servers or gateway management servers that are not members 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**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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 an array of resource pool objects. 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Add objects to a resource pool

This example changes the properties of a resource pool by adding objects to the resource pool.

The first command uses the **Get-SCOMManagementServer** cmdlet to get all management servers and stores them in the \$Members variable.

The second command uses the **Get-SCOMResourcePool** cmdlet to get the resource pool with a display name of Pool01, and then uses the **Set-SCOMResourcePool** cmdlet to add 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: Remove objects from a resource pool

This example changes the properties of a resource pool by removing objects from the resource pool.

The first command uses the **Get-SCOMManagementServer** cmdlet to get the management server with the display name of Member01 and stores it in the \$Member variable.

The second command uses the **Get-SCOMResourcePool** cmdlet to get the resource pool with the display name of Pool01 and then uses the **Set-SCOMResourcePool** cmdlet to remove 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: Add an observer object to a resource pool

This example changes the properties of a resource pool by adding an observer object to the resource pool.

The first command uses the **Get-SCOMManagementServer** cmdlet to get the management server named Observer01, and stores the object in the variable named \$Observer.

---

The second command uses the **Get-SCOMResourcePool** cmdlet to get the resource pool with the display name of Pool01 and then uses the **Set-SCOMResourcePool** cmdlet to add 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: Remove an observer object from a resource pool

This example changes the properties of a resource pool by removing an observer object from the resource pool.

The first command uses the **Get-SCOMManagementServer** cmdlet to get the management server named Observer01, and stores the object in the variable named \$Observer.

The second command uses the **Get-SCOMResourcePool** cmdlet to get the resource pool with the display name of Pool01 and then uses the **Set-SCOMResourcePool** cmdlet to remove 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: Make a display name property change to a resource pool

This example changes the properties of a resource pool by changing a property of the resource pool.

The first command uses the **Get-SCOMResourcePool** cmdlet to get the resource pool with a display name of Pool01, and stores the object in the variable named \$Pool.

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: Make a description property change to a resource pool

This example makes a change to a description property for a resource pool.

The first command uses the **Get-SCOMResourcePool** cmdlet to get the resource pool with a display name of New Pool 02, and stores the object in the variable named \$Pool.

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 a 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 root management server (RMS) Emulator role on the specified management server or moves it to this server if the role already exists in the management group.

The RMS Emulator role gives legacy management packs that depend on an RMS the ability to continue working in System Center 2012 – Operations Manager, which does not support RMS.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	localhost
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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 an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

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. To obtain a management server object, use the **Get-SCOMManagementServer** cmdlet.

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: Create the RMS Emulator role on a management server**

This command creates the RMS Emulator role on the management server named Server01.Contoso.com. The command uses the **Get-SCOMManagementServer** cmdlet to get the specified management server object and passes the object to the **Set-SCOMRMSEmulator** cmdlet.

```
PS C:\> Get-SCOMManagementServer -Name "Server01.Contoso.com" | Set-SCOMRMSEmulator
```

---

## **Related topics**

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMManagementServer](#)

[Get-SCOMRMSEmulator](#)

[Remove-SCOMRMSEmulator](#)

---

# Set-SCOMRunAsDistribution

---

## Set-SCOMRunAsDistribution

Sets the distribution policy of a Run As account for Operations Manager.

### 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 a Run As account for System Center 2012 – Operations Manager. Distribution policies determine which computers receive a credential for a Run As account. By default, new accounts have the more secure distribution with no approved systems.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LessSecure**

Indicates that Operations Manager distributes the credential automatically to all managed computers.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-MoreSecure**

Indicates that Operations Manager distributes the credential only to systems that the *SecureDistribution* parameter specifies.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## **-RunAsAccount<SecureData>**

Specifies an array of **SecureData** objects that represent Run As accounts. To obtain a **SecureData** object, use the **Get-SCOMRunAsAccount** cmdlet. This account cannot be part of a Run As profile.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-SecureDistribution<Object[]>**

Specifies an array of objects that represent systems that the cmdlet authorizes for distribution.

This parameter list can contain only the following types of items:

-- Agents. Objects that the **Get-SCOMAgent** cmdlet returns. The cmdlet authorizes this account to the agent for distribution.

-- Management servers. Objects that the **Get-SCOMManagementServer** cmdlet returns. The cmdlet authorizes this account to the agent for distribution.

-- Pools. Objects that the **Get-SCOMResourcePool** cmdlet returns. The cmdlet authorizes this account to the agent for distribution.

-- Health service instances. Objects that the **Get-SCOMClassInstance** cmdlet returns and that have a managed type of **HealthService**. The cmdlet authorizes this health service to the agent for distribution.

Passing output from the **Get-SCOMRunAsDistribution** cmdlet as input to **Set-SCOMRunAsDistribution** by using the pipeline operator automatically populates this parameter.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Security<String>**

Specifies the security level. The acceptable values for this parameter are:

-- MoreSecure

-- LessSecure

Passing output from the **Get-SCOMRunAsDistribution** cmdlet as input to **Set-SCOMRunAsDistribution** by using the pipeline operator automatically populates this parameter.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### Example 1: Specify a less secure distribution

This command sets the Contoso\LowPriv account for less secure distribution, so that Operations Manager distributes the credential automatically to all managed computers. The command uses the **Get-SCOMRunAsAccount** cmdlet to get the specified account and passes it to the **Set-SCOMRunAsDistribution** cmdlet by using the pipeline operator. The command specifies the *LessSecure* parameter.

```
PS C:\> Get-SCOMRunAsAccount -Name "Contoso\LowPriv" | Set-SCOMRunAsDistribution -LessSecure
```

### Example 2: Specify a more secure distribution without approved systems

This command sets the Contoso\LowPriv account for more secure distribution, with no approved systems. The command uses the **Get-SCOMRunAsAccount** cmdlet to get the specified account and passes it to the **Set-SCOMRunAsDistribution** cmdlet by using the pipeline operator. The command specifies the *MoreSecure* parameter.

```
PS C:\> Get-SCOMRunAsAccount -Name "Contoso\LowPriv" | Set-SCOMRunAsDistribution -MoreSecure
```

### Example 3: Specify a more secure distribution to a collection

This example sets the Contoso\LowPriv account for more secure distribution to a collection of pools, agents, and servers, so that only specified pools, agents, and servers get the distribution.

The first command gets the pools, agents, and servers to receive more secure distribution and stores them in the \$Distribution variable .

The second command gets pools, agents, and servers that have less secure distribution and passes them to the **Set-SCOMRunAsDistribution** cmdlet by using the pipeline operator. That cmdlet assigns them more secure distribution.

```
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: Specify less secure distribution for a new Run As account

This command creates a Run As account for Windows and approves it for distribution to all agents. The command uses the **Add-SCOMRunAsAccount** cmdlet to add the account NewAccount with the credential that the **Get-Credential** cmdlet creates. It then passes the result to the **Set-SCOMRunAsDistribution** cmdlet by using the pipeline operator.

```
PS C:\> Add-SCOMRunAsAccount -Windows -Name "NewAccount" -Credential (Get-Credential) | Set-SCOMRunAsDistribution -MoreSecure -SecureDistribution (Get-SCOMAgent)
```

## Example 5: Copy a more secure distribution policy to a different account

This example copies the secure distribution policy from the Contoso\Administrator account to the Contoso\Monitoring account.

The first command uses the **Get-SCOMRunAsAccount** cmdlet to get the Contoso\Monitoring account and stores it in the \$MonitoringAcct variable.

The second command uses the **Get-SCOMRunAsAccount** cmdlet to get the Contoso\Administrator account and passes it to the **Get-SCOMRunAsDistribution** cmdlet by using the pipeline operator. The command passes the result to the **Set-SCOMRunAsDistribution** cmdlet to copy the result to the Contoso\Monitoring account.

```
PS C:\> $MonitoringAcct = Get-SCOMRunAsAccount "Contoso\Monitoring"
PS C:\> Get-SCOMRunAsAccount "Contoso\Administrator" | Get-SCOMRunAsDistribution | Set-SCOMRunAsDistribution -RunAsAccount $MonitoringAccount
```

## Related topics

[Get-SCOMAgent](#)

[Get-SCOMClassInstance](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMManagementServer](#)

[Get-SCOMResourcePool](#)

[Get-SCOMRunAsAccount](#)

[Get-SCOMRunAsDistribution](#)

---

# 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 an array of **Microsoft.EnterpriseManagement.Security.SecureData** objects that represent Run As accounts. To obtain a **SecureData** object, use the **Get-SCOMRunAsAccount** cmdlet.

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 to update the Run As profile. The acceptable values for this parameter are: Add or Remove.

Aliases	none
Required?	true
Position?	4
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Class<ManagementPackClass[]>**

Specifies an array of **ManagementPackClass** objects that represent the classes that the Run As account can manage. Specify a variable that stores classes, or use a cmdlet such as **Get-SCOMClass** that gets classes.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-Group<MonitoringObject[]>**

Specifies an array of monitoring objects that represent the groups that the Run As account can manage. Specify a variable that stores groups, or use a cmdlet such as **Get-SCOMGroup** that gets groups.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Instance<MonitoringObject[]>**

Specifies an array of monitoring objects that represent the class instances that the Run As account can manage. Specify a variable that stores class instances, or use a cmdlet such as **Get-SCOMClassInstance** that gets class instances. This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## **<Profile<ManagementPackSecureReference>**

Specifies the Run As profile to update. Specify a variable that stores a Run As profile, or use a cmdlet such as **Get-SCOMRunAsProfile** that gets a profile.

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: Add a Run As account to a Run As profile by specifying an account name**

This example adds a Run As account to a Run As profile by specifying an account name.

The first command gets the Run As profile that has 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. Because the command does not specify a class, group or object, it configures the Run As account to manage all targeted objects.

```
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: Add a Run As account to a Run As profile by specifying a path

This example adds a Run As account to a Run As profile by specifying a path name.

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: Add a Run As account to a Run As profile by specifying an object

This example adds a Run As account to a Run As profile by specifying an account object.

The first command gets the Run As profile object that has 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 that has 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, which is stored in the \$Profile 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
```

## Related topics

[Add-SCOMRunAsProfile](#)

[Get-SCOMClass](#)

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

---

[Get-SCOMRunAsAccount](#)

[Get-SCOMRunAsProfile](#)

[Get-SCOMProfile](#)

[Remove-SCOMRunAsProfile](#)

---

# 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 a System Center 2012 – Operations Manager user role. A user role contains two features:

- Profile. Defines the collection of operations to which the user role has access.
- Scope. Defines the boundaries for profile operations; for example, tasks and groups.

You can use this cmdlet to update the user list or the scope of the user role.

---

## Parameters

### **-AllClass**

Indicates that the permissions that an Author user role grants extend to all classes.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AllGroup**

Indicates that the permissions that a user role grants extend to all groups.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AllTask**

Indicates that the permissions that an Author, Operator, or Advanced Operator user role grants extend to all tasks.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-ClassScope<ManagementPackClass[]>**

Specifies an array of **ManagementPackClass** objects that represent the classes to which an Author role has access. To obtain a **ManagementPackClass** object, use the **Get-SCOMClass** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-GroupScope<MonitoringObjectGroup[]>**

Specifies an array of **MonitoringObjectGroup** objects that represent the groups to which a user role has access. To obtain a monitoring object group object, use the **Get-MonitoringObjectGroup** cmdlet.

If you do not specify a value for this parameter, the role has access to all groups. To deny the role access to all 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**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

### **-TaskScope<ManagementPackTask[]>**

Specifies an array of **ManagementPackTask** objects that represent the tasks to which an Author, Operator, or Advanced Operator role has access.

If you do not specify a value for this parameter, the role has access to all tasks. To deny the Author, Operator, or Advanced Operator role access to all 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 an array of user names that are part of this 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. To obtain a **UserRole** object, us the **Get-SCOMUserRole** 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: Grant a user role access to all tasks

This example shows how to update a user role to grant it access to all tasks.

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 the \$Role variable to the **Set-SCOMUserRole** cmdlet. This cmdlet 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** by using the pipeline operator, 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: Grant a user role access to specific tasks

This example shows how to update a user role to grant it access to specific tasks.

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 names and stores the objects in the \$NewTaskList variable.

The last command uses the pipeline operator to pass the user role stored in the \$Role variable to **Set-SCOMUserRole**, which resets the collection of approved tasks to the list stored in the \$NewTaskList variable.

```
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: Add a user a user role

This example adds a user to a user role.

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 object stored in the \$Role variable to the **Set-SCOMUserRole** cmdlet, 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")
```

---

## Related topics

[Add-SCOMUserRole](#)

[Get-SCOMUserRole](#)

[Get-MonitoringObjectGroup](#)

[Get-SCOMClass](#)

---

# Set-SCOMWebAddressSetting

---

## Set-SCOMWebAddressSetting

Sets the URLs for the web console and for online product knowledge.

### 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 URLs for the web console and for online product knowledge of a management group.

The web console is a browser-based application that enables users to monitor management group data from the Internet.

Product knowledge is the collection of notes that are attached to objects in a management pack. These notes document issues and suggested resolutions for those issues.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The computer must run the System Center Data Access service.

If you do not specify this parameter, the default is the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-OnlineProductKnowledgeUrl<Uri>**

Specifies 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**

Indicates that the cmdlet creates or modifies an object that a command can use in the pipeline. 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

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To get **Connection** objects, specify the **Get-SCOMManagementGroupConnection** cmdlet.

If you do not specify a value for this parameter, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-WebConsoleUrl<Uri>**

Specifies 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: Set the web console URL for a management group

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: Set the product knowledge URL for a management group

This command sets the online product knowledge URL to `http://OM01.contoso.com/Knowledge`.

```
PS C:\> Set-SCOMWebAddressSetting -OnlineProductKnowledgeUrl  
"http://OM01.contoso.com/Knowledge"
```

---

## Related topics

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMWebAddressSetting](#)

---

# Start-SCOMMaintenanceMode

---

## Start-SCOMMaintenanceMode

Puts an object into maintenance mode and creates an active maintenance mode entry.

### Syntax

Parameter Set: FromInstance

```
Start-SCOMMaintenanceMode [-Instance] <MonitoringObject[]> [-EndTime] <DateTime> [[-Comment] <String> ] [[-Reason] <MaintenanceModeReason> ] [-PassThru] [ <CommonParameters>]
```

### Detailed Description

The **Start-SCOMMaintenanceMode** cmdlet puts a monitored object, such as a computer or distributed application, into maintenance mode and creates an active maintenance mode entry. When a resource is in maintenance mode, System Center 2012 – Operations Manager suppresses alerts, notifications, rules, monitors, automatic responses, state changes, and new alerts.

Specify a class instance to put into maintenance mode and an end time for the maintenance window. You can also include a comment and a reason for the maintenance mode. You can use the **Set-SCOMMaintenanceMode** cmdlet to update an active maintenance mode entry and use the **Get-SCOMMaintenanceMode** cmdlet to get both active and inactive entries.

### Parameters

#### **-Comment<String>**

Specifies a comment for the maintenance mode entry.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-EndTime<DateTime>**

Specifies when maintenance mode ends, as a **DateTime** object. A resource cannot be in maintenance mode for fewer than five minutes. To obtain a **DateTime** object, use the **Get-Date** cmdlet. For more information, type `Get-Help Get-Date`.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Instance<MonitoringObject[]>**

Specifies an array of monitoring objects that represent instances. To obtain monitoring objects, use the **Get-SCOMClassInstance** cmdlet.

This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## **-Reason<MaintenanceModeReason>**

Specifies a reason for maintenance mode. The acceptable values for this parameter 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: Put a resource into maintenance mode

This example puts a resource into maintenance mode for ten minutes. The first command gets the class instance named Server01.Contoso.com by using the **Get-SCOMClassInstance** cmdlet.

The second command creates a **DateTime** object for ten minutes in the future and then stores it in the \$Time variable.

The third command puts the resource defined by the object stored in the \$Instance variable into maintenance mode. Maintenance mode ends at the time stored in the \$Time variable. The command includes a reason for maintenance mode and a comment.

```
PS C:\> $Instance = Get-SCOMClassInstance -Name "Server01.Contoso.com"
PS C:\> $Time = ((Get-Date).AddMinutes(10))
PS C:\> Start-SCOMMaintenanceMode -Instance $Instance -EndTime $Time -Comment "Applying
software update." -Reason "SecurityIssue"
```

### Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMMaintenanceMode](#)

[Set-SCOMMaintenanceMode](#)

[Get-SCOMGroup](#)

---

# Start-SCOMTask

---

## Start-SCOMTask

Starts a task for a specified object.

### Syntax

Parameter Set: Empty

```
Start-SCOMTask [-Instance] <EnterpriseManagementObject[]> [-Task] <ManagementPackTask> [[-TaskCredentials] <PSCredential> ] [[-Override] <Hashtable> ] [ <CommonParameters>]
```

### Detailed Description

The **Start-SCOMTask** cmdlet starts a task for a specified object. This cmdlet takes a task object and a class instance object as input. The cmdlet accepts only one task, but it can accept multiple class instances and overrides.

### Parameters

#### **-Instance<EnterpriseManagementObject[]>**

Specifies an array of [EnterpriseManagementObject](#) objects that represent class instance objects for which to start a task. Specify a variable that stores the class instances or use a cmdlet such as the **Get-SCOMClassInstance** cmdlet that gets the class instances. This parameter also accepts group objects. To obtain a group object, use the **Get-SCOMGroup** cmdlet.

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 new values for one or more task parameters, if those parameters allow overrides. For more information about hash tables, type "Get-Help about\_Hash\_Tables".

To determine which parameters for a task allow overrides, use the **GetOverrideableParameters** method for the task object.

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. Specify a variable that represents a task, or use a cmdlet such as the **Get-SCOMTask** cmdlet that gets a task. This parameter accepts only one task object.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-TaskCredentials<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type "Get-Help Get-Credential".

This account must have access to the server that is specified in the *ComputerName* parameter if that parameter appears.

If you do not specify this parameter, the default is the account for the current user.

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: Start a task by using a display name

This example starts a task by using a display name.

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 that has the display name Get Monitor State and starts the task for each class instance that is 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: Start a task by using a display name and a timeout specification

This example shows how to start a task by using a display name and a hash table that specifies a timeout value.

The first command creates a hash table that has a value for the Timeout key and stores it in the \$Overrides variable.

The second command gets all class instances that have the name Contoso Service and stores them in the \$Instances variable.

The third command gets the task that has the display name Start NT Service and starts the task for each class instance that is stored in the \$Instances variable.

The fourth command starts the task for each class instance that is stored in the \$Instances variable. The command specifies a Timeout key value 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: Start a task by using credentials

This example starts a task by using credentials.

The first command prompts the user for a user name and password, creates a credential object from the input, and stores the credential object in the `$Credential` 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 that has the display name `Reset State` and starts the task for the class instance that is stored in `$Instance`. The command runs the task under the credentials that are stored in `$Credential`.

```
PS C:\> $Credential = Get-Credential
PS C:\> $Instance = Get-SCOMClassInstance -Name "Server01.Contoso.com"
PS C:\> Get-SCOMTask -DisplayName "Reset State" | Start-SCOMTask -Instance $Instance -
TaskCredentials $Credential
```

## Related topics

[Get-SCOMClassInstance](#)

[Get-SCOMGroup](#)

[Get-SCOMTask](#)

[Get-SCOMTaskResult](#)

---

# Test-SCOMCEIP

---

## Test-SCOMCEIP

Determines if Operations Manager 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 System Center 2012 – Operations Manager Customer Experience Improvement Program (CEIP) is enabled on the local computer. The cmdlet returns \$True if CEIP is enabled or \$False if CEIP is disabled.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **<Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential1`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: `-Verbose`, `-Debug`, `-ErrorAction`, `-ErrorVariable`, `-OutBuffer`, and `-OutVariable`. For more information, see [about\\_CommonParameters](#).

---

## Examples

### Example 1: Determine if CEIP is enabled on a computer

This command returns the result of whether CEIP is enabled on the local computer.

```
PS C:\> Test-SCOMCEIP
```

## Related topics

[Join-SCOMCEIP](#)

[Exit-SCOMCEIP](#)

---

# Test-SCOMOperationalDataReporting

---

## Test-SCOMOperationalDataReporting

Determines the status of operational data reporting for the management group.

### Syntax

Parameter Set: Default

```
Test-SCOMOperationalDataReporting [-ComputerName <String[]> ] [-Credential <PSCredential> ]  
[-SCSession <Connection[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Test-SCOMOperationalDataReporting** cmdlet determines the status of operational data in System Center 2012 – Operations Manager.

The cmdlet returns \$True if operational data reporting is enabled; otherwise, the cmdlet returns \$False.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). 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 **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Determine the status of operational data reporting**

This command determines the status of operational data reporting.

```
PS C:\> Test-SCOMOperationalDataReporting
```

---

## Related topics

[Disable-SCOMOperationalDataReporting](#)

[Enable-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. Before you uninstall an agent, you must remove the Active Directory Domain Services (AD DS) agent assignments from the management group. You can use the **Remove-SCOMADAgentAssignment** cmdlet to remove AD DS agent assignments from the management group.

### Parameters

#### **-ActionAccount<PSCredential>**

Specifies a **PSCredential** object. This parameter specifies the credentials that Operations Manager uses to run the deployment task. If you do not specify this parameter or you specify a null value, Operations Manager uses the default action account of the management server for the agent.

To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Agent<AgentManagedComputer[]>**

Specifies an array of **AgentManagedComputer** objects. This parameter specifies the Operations Manager agents to uninstall. To obtain an **AgentManagedComputer** object, use the **Get-SCOMADAgent** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

## **<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: Uninstall an agent

The first command gets the Operations Manager agent object named server01.contoso.com and passes it to the **Uninstall-SCOMAgent** cmdlet by using the pipeline operator. The second command uninstalls the agent named server01.contoso.com.

```
PS C:\> Get-SCOMAgent -DNSHostName "server01.contoso.com" | Uninstall-SCOMAgent
```

### Example 2: Uninstall an agent by using an object variable

This example uninstalls the Operations Manager agent on an agent-managed computer.

The first command gets the agent-managed computer object named server01.contoso.com and stores the object in the \$Agent variable.

The second command uninstalls the agent stored in \$Agent.

```
PS C:\> $Agent = Get-SCOMAgent -DNSHostName "server01.contoso.com"
PS C:\> Uninstall-SCOMAgent -Agent $Agent
```

### Example 3: Uninstall an agent by using an action account

This example uninstalls the Operations Manager agent on an agent-managed computer after the user enters the credentials that Operations Manager requires to uninstall the agent.

The first command gets the agent object named server01.contoso.com and stores the object in the \$Agent variable.

The second command prompts the user to enter the credentials that Operations Manager uses to uninstall an agent. The command then uninstalls the agent stored in \$Agent.

```
PS C:\> $Agent = Get-SCOMAgent -DNSHostName "server01.contoso.com"
PS C:\> Uninstall-SCOMAgent -Agent $Agent -ActionAccount (Get-Credential)
```

## Related topics

[Get-SCOMAgent](#)

[Install-SCOMAgent](#)

[Get-SCOMRunAsAccount](#)

[Get-SCOMManagementServer](#)

[Repair-SCOMAgent](#)

---

# Update-SCOMADAgentAssignment

---

## Update-SCOMADAgentAssignment

Changes settings of an AD DS 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 changes settings of an Active Directory Domain Services (AD DS) agent assignment. You can use this cmdlet to change the LDAP query and exclusion list settings of the agent assignment. To change other settings of the agent assignment, remove the agent assignment and re-create it.

### Parameters

#### **-AgentAssignment<AgentAssignment>**

Specifies an **AgentAssignment** object. To obtain an **AgentAssignment** object, use the **Get-SCOMADAgentAssignment** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

---

## **-ComputerName<String[]>**

Specifies an array of names of computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, **localhost**, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-Exclude<String[]>**

Specifies an array of names of computers. Operations Manager excludes the computers that you specify from the agent assignment. Operations Manager excludes these computers even if the LDAP query returns the computers.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-LdapQuery<String>**

Specifies the LDAP query in the domain which selects the target agent computers. If you do not specify this parameter, the cmdlet uses the current LDAP query.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

---

## **-PrimaryServer<ManagementServer>**

Specifies a **ManagementServer** object. This parameter specifies the primary management server for the target agent-managed computers. To obtain a **ManagementServer** object, use the **Get-SCOMManagementServer** cmdlet.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

## **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To obtain a **Connection** object, use the **Get-SCOMManagementGroupConnection** cmdlet.

A connection object represents a connection to a management server. The default is the current management group connection.

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: Change the LDAP query for an agent assignment

This command gets the AD DS agent assignment for a domain. The command uses the **Get-SCOMADAgentAssignment** cmdlet to get the AD DS agent assignment named contoso.com, and passes the result to the **Update-SCOMADAgentAssignment** cmdlet by using the pipeline operator. The command changes the LDAP query of the agent assignment for the domain named contoso.com to return only computers with names that match SQLSERVER.

```
PS C:\> Get-SCOMADAgentAssignment -Domain "contoso.com" | Update-SCOMADAgentAssignment -LdapQuery "&(sAMAccountType=805306369)(name=SQLSERVER*)"
```

## Related topics

[Get-SCOMADAgentAssignment](#)

[Add-SCOMADAgentAssignment](#)

[Remove-SCOMADAgentAssignment](#)

---

# Update-SCOMLocation

---

## Update-SCOMLocation

Updates the display name, latitude, or longitude of a location.

### 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 display name, latitude, and longitude of a location. You can get a location to update by using the **Get-SCOMLocation** cmdlet. You can create a location by using the **New-SCOMLocation** cmdlet.

By default, this cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

### Parameters

#### **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be running on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Credential<PSCredential>**

Specifies a **PSCredential** object for the management group connection. To obtain a **PSCredential** object, use the **Get-Credential** cmdlet. For more information, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-DisplayName<String>**

Specifies a new display name for the location.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-Latitude<String>**

Specifies a latitude value for the 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 a location as an **EnterpriseManagementObject** object. To obtain a location, 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 a longitude value for the 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 cmdlet creates or modifies an object that a command can use in 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 an array of **Connection** objects. To obtain a **Connection** object, use 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: Update a location**

This command updates the latitude and longitude of the location Los Angeles, CA. The command uses the **Get-SCOMLocation** cmdlet to get the location with the specified display name and passes the result to the **Update-SCOMLocation** cmdlet by using the pipeline operator. The **Update-SCOMLocation** command updates the latitude and longitude of the location.

```
PS C:\> Get-SCOMLocation -DisplayName "Los Angeles, CA" | Update-SCOMLocation -Latitude 33.942809 -Longitude -118.4047064
```

---

## Related topics

[Get-SCOMAgent](#)

[Get-SCOMManagementServer](#)

[Get-SCOMResourcePool](#)

[Get-SCOMManagementGroupConnection](#)

[Get-SCOMLocation](#)

[New-SCOMLocation](#)

[Remove-SCOMLocation](#)

[Set-SCOMLocation](#)

---

# 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 System Center 2012 – Operations Manager Run As account. A Run As account gives users the ability to specify the necessary permissions for use with rules, tasks, monitors, and discoveries targeted to specific computers on an as-needed basis.

This cmdlet does not support SCX (cross-platform) accounts or Simple Network Management Protocol version 3 (SNMPv3) accounts.

## Parameters

### **-AccountType<String>**

Specifies the type of the Run As account. The pipeline typically populates this value automatically.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	true (ByPropertyName)
Accept Wildcard Characters?	false

### **-ActionAccount<ActionAccountSecureData>**

Specifies an action account as a 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 Run As account that uses Basic Authentication.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-BinaryAccount<GenericSecureData>**

Specifies a Run As account that uses Binary Authentication.

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 an SNMP version 2 Run As account that uses community string authentication.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-CommunityStringAccount<CommunityStringSecureData>**

Specifies an SNMP version 2 Run As account that uses community string authentication.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-ComputerName<String[]>**

Specifies an array of names of computers. The cmdlet establishes temporary connections with management groups for these computers. You can use NetBIOS names, IP addresses, or fully qualified domain names (FQDNs). To specify the local computer, type the computer name, localhost, or a dot (.).

The System Center Data Access service must be started on the computer. If you do not specify a computer, the cmdlet uses the computer for the current management group connection.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Credential<PSCredential>**

Specifies the user account under which the management group connection runs. Specify a **PSCredential** object, such as one that the **Get-Credential** cmdlet returns, for this parameter. For more information about credential objects, type `Get-Help Get-Credential`.

If you specify a computer in the *ComputerName* parameter, use an account that has access to that computer. The default is the current user.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DigestAccount<SimpleCredentialSecureData>**

Specifies a Run As account that uses standard digest web authentication.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

### **-PassThru**

Indicates that the cmdlet creates or modifies an object that a command can use in 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

### **-Path<String>**

Specifies the path to a file that contains new credential data for Run As accounts that use Binary Authentication.

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 username and password.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SCSession<Connection[]>**

Specifies an array of **Connection** objects. To get **Connection** objects, use the **Get-SCOMManagementGroupConnection** cmdlet.

If this parameter is not specified, the cmdlet uses the active persistent connection to a management group. Use the *SCSession* parameter to specify a different persistent connection. You can create a temporary connection to a management group by using the *ComputerName* and *Credential* parameters. For more information, type `Get-Help about_OpsMgr_Connections`.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-SimpleAccount<SimpleCredentialSecureData>**

Specifies a Run As account that uses simple web authentication.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	true (ByValue)
Accept Wildcard Characters?	false

## **-WindowsAccount<WindowsCredentialSecureData>**

Specifies a Run As account that uses Windows authentication.

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: Update a Windows Run As account

This example updates a Windows Run As account.

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 the \$WindowsAccount variable to the **Update-SCOMRunAsAccount** cmdlet by using the pipeline operator. That cmdlet updates the credentials for the account to the user name and password that the user provides by responding to the prompt from the **Get-Credential** cmdlet.

```
PS C:\> $WindowsAccount = Get-SCOMRunAsAccount -Name "Domain Admin"  
PS C:\> $WindowsAccount | Update-SCOMRunAsAccount -RunAsCredential (Get-Credential)
```

### Example 2: Update an action Run As account

This example updates credentials for an action Run As account.

The first command creates a string that contains an action account name and stores the string in the \$UserName variable.

The second command prompts the user to enter a string that represents the password. It then stores the user input as a secure string in the \$Password variable.

The third command creates a **PSCredential** object by using the name stored in the \$UserName variable and the password stored in \$Password. It then stores the **PSCredential** object in the \$NewCred variable.

---

The last command gets the action account named SCOM Action Account. It then uses the pipeline operator to pass the account to the **Update-SCOMRunAsAccount** cmdlet, which updates the account with the credentials stored in the \$NewCred variable.

```
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: Update a Community String account

This command updates credentials for a Run As account that uses community string authentication for SNMP version 2. It gets the community string account named MyCommunityStringAccount and uses the pipeline operator to pass the account to the **Update-SCOMRunAsAccount** cmdlet. This cmdlet updates the community string with the value that the user enters in response to prompts from the **Read-Host** cmdlet. For more information, type `Get-Help Read-Host`.

```
PS C:\> Get-SCOMRunAsAccount -Name "MyCommunityStringAccount" | Update-SCOMRunAsAccount -
CommunityString (Read-Host -AsSecureString)
```

### Example 4: Update a Binary Authentication account

This example gets the binary account object named MyBinaryAccount and uses the pipeline operator to pass the object to the **Update-SCOMRunAsAccount** cmdlet. This cmdlet uses the *Path* parameter to indicate that it gets the new account data from the file named data.txt.

```
PS C:\> Get-SCOMRunAsAccount -Name "MyBinaryAccount" | Update-SCOMRunAsAccount -Path
"..\data.txt"
```

### Related topics

[Add-SCOMRunAsAccount](#)

[Get-SCOMRunAsAccount](#)

[Get-SCOMManagementGroupConnection](#)

[New-SCOMRunAsAccount](#)

---

# Write-SCOMCommand

---

## Write-SCOMCommand

Writes a new 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 Windows PowerShell command to interact with System Center 2012 – Operations Manager. The new command uses the same common SDK connection code as the standard Operations Manager cmdlets.

### Parameters

#### **-Name<String>**

Specifies the name of the command.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-NoName**

Indicates that the cmdlet creates 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: Write a new script command

This command outputs a new script command template named Test-DivBManagementPack.

```
PS C:\> Write-SCOMCommand -Name "Test-DivBManagementPack"
```

## Related topics

[Get-SCOMCommand](#)

