Microsoft System Center 2012 R2

Cmdlet Reference for UNIX and Linux 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.

Copyright

Revision History

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

Contents

Get-SCXAgent 4

Get-SCXSSHCredential 9

Install-SCXAgent 13

Invoke-SCXDiscovery 17

Remove-SCXAgent 23

Set-SCXResourcePool 27

Uninstall-SCXAgent 31

Update-SCXAgent 36

Get-SCXAgent

Get-SCXAgent

Gets a list of managed UNIX and Linux computers in a management group.

Syntax

```
Parameter Set: Empty
Get-SCXAgent [-SCSession <Connection[]> ] [ <CommonParameters>]

Parameter Set: FromAgentGuidsParameterSetName
Get-SCXAgent [-ID] <Guid[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [
<CommonParameters>]

Parameter Set: FromAgentNamesParameterSetName
Get-SCXAgent [-Name] <String[]> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [
<CommonParameters>]

Parameter Set: FromHostNameAndResourcePoolParameterSetName
Get-SCXAgent -ResourcePool <ManagementServicePool> [ <CommonParameters>]

Parameter Set: FromIPRangeAndResourcePoolParameterSetName
Get-SCXAgent -ResourcePool <ManagementServicePool> [ <CommonParameters>]

Parameter Set: FromManagementServerParameterSetName
Get-SCXAgent [-ComputerName <String[]> ] [-Credential <PSCredential> ] [ <CommonParameters>]
```

Detailed Description

The **Get-SCXAgent** cmdlet retrieves the managed UNIX and Linux computers that match the provided selection criteria. If no selection criteria are specified, all managed UNIX and Linux computers in the management group are returned.

Results are returned as an array of managed UNIX and Linux computer objects.

Parameters

-ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

-Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

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

-ID<Guid[]>

Specifies the ID of a managed UNIX or Linux computer object. Returns only the only managed computers with ID properties that match this value. This parameter is accepted from the pipeline.

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

-Name<String[]>

Specifies a list of fully qualified domain names for one or more managed UNIX and Linux computers. Returns only the managed computers that match the **Name** property.

You can include wildcards, in which case all computers matching the wildcard are returned. For information about using wildcards, type `Get-Help About_Wildcards`.

This parameter is accepted from the pipeline.

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

-ResourcePool<ManagementServicePool>

Specifies a resource pool of servers, one of which is assigned to be the current management server and the other serving as backup management servers. This parameter requires a resource pool object and returns only the managed computers in that resource pool.

For information about how to get a SCOMResourcePool object, type `Get-Help Get-SCOMResourcePool`.

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

-SCSession<Connection[]>

Specifies a management group connection to use instead of the default connection.

For more information, type `Get-Help Get-ManagementGroupConnection`.

Aliases	none
Required?	false

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

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -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.Unix.Computer[]**

  This cmdlet returns one or more Computer objects that represent managed UNIX and Linux computers when the UNIX and Linux computers match input criteria.

Examples

-------------------------- EXAMPLE 1 --------------------------

Description

This command returns all managed UNIX and Linux computers in the resource pool.

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

-------------------------- EXAMPLE 2 --------------------------

Description

This command returns all managed UNIX and Linux computers for a specified resource pool.

```
C:\PS>$myPool = Get-SCOMResourcePool -DisplayName "pool01"
Get-SCXAgent -ResourcePool $myPool
```

-------------------------- EXAMPLE 3 --------------------------

Description

This command returns managed UNIX and Linux computers with names that are matching a provided list of strings.

```
C:\PS>Get-SCXAgent -Name nx1.contoso.com,nx2.*,*.development.contoso.com
```

-------------------------- EXAMPLE 4 --------------------------

Description

This command provides the *Name* parameter through the pipeline.

```
C:\PS>"server1", "server2", "server3" | Get-SCXAgent
```

-------------------------- EXAMPLE 5 --------------------------

Description

This command connects to a resource pool with a temporary connection, by using the current user's credentials, and returns only the managed computers that match the **Name** property.

```
C:\PS>Get-SCXAgent -Name *development.contoso.com -ComputerName server1.contoso.com
```

Get-SCXSSHCredential

Get-SCXSSHCredential

Creates a privileged credential, by using the Secure Shell (SSH) protocol, for management operations on UNIX and Linux computers.

Syntax

```
Parameter Set: Default
Get-SCXSSHCredential [-Username] <String> [-ElevationType <String> ] [-SSHKey <String> ] [
<CommonParameters>]
```

Detailed Description

The **Get-SCXSSHCredential** cmdlet creates a credential for use in privileged UNIX or Linux SSH agent management operations.

Parameters

-ElevationType<String>

Specifies the elevation method, either "su" or "sudo", that elevates the supplied credential to a privileged account on the UNIX or Linux computer. If left unspecified, the cmdlet treats the credential that is used to create the SSH connection as privileged. If *ElevationType* is "su", a root password must be provided for the *SuPassword* parameter.

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

-SSHKey<String>

Specifies the file name, including its path, of an SSH key. The SSH key file must be in PuTTY key format. This parameter is required unless a password is specified.

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

-Username<String>

Specifies the user name for an SSH connection.

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

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see about_CommonParameters.

Outputs

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

- **Microsoft.Unix.SSHCredential** This cmdlet returns the SSHCredential object as output.

Examples

-------------------------- EXAMPLE 1 --------------------------

Description

This command creates a privileged SSH credential with a user name and prompts for a password.

```
C:\PS>Get-SCXSSHCredential -UserName root
Password:  *************
```

-------------------------- EXAMPLE 2 --------------------------

Description

This command creates a low-privileged credential with a user name and “su” elevation type. It then prompts for a password for the low-privileged account and the “su” password for the elevation.

```
C:\PS>Get-SCXSSHCredential -UserName joe -ElevationType su
Password:  *************
Su Password: *************
```

-------------------------- EXAMPLE 3 --------------------------

Description

This command creates a credential with a low-privileged account with a user name and a “sudo” elevation type. It then prompts for a password for the low-privileged account.

```
C:\PS>Get-SCXSSHCredential -UserName joe -ElevationType sudo
Password: *************
```

-------------------------- EXAMPLE 4 --------------------------

Description

This command creates a privileged credential with a user name and an SSH key. It then prompts for a passphrase. If a passphrase was not configured with the SSH key, just press Enter.

```
C:\PS>Get-SCXSSHCredential -UserName root -SSHKey c:\keys\admin.ppk
Key Passphrase: *************
```

-------------------------- EXAMPLE 5 --------------------------

Description

This command creates a low-privileged credential with a user name, an SSH key, and a “su” elevation type. It then prompts for the “su” password and a key passphrase.

```
C:\PS>Get-SCXSSHCredential -UserName joe -SSHKey c:\keys\userJoe.ppk -ElevationType su
Su Password: *************
Key Passphrase:
```

-------------------------- EXAMPLE 6 --------------------------

Description

This command creates a low-privileged credential with a user name, an SSH key, and a “sudo” elevation type. It then prompts for a key passphrase.

```
C:\PS>Get-SCXSSHCredential -UserName joe -SSHKey c:\keys\userJoe.ppk -ElevationType sudo
Key Passphrase:
```

Install-SCXAgent

Install-SCXAgent

Installs the Operations Manager agent for discovered UNIX and Linux computers, signs the certificate for WS-Management communication, and registers the agent with Operations Manager for management.

Syntax

```
Parameter Set: Default
Install-SCXAgent [-DiscoveryResult] <DiscoveryResult[]> [-ComputerName <String[]> ] [-
Credential <PSCredential> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: Empty
Install-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromAgentNames
Install-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromManagementServer
Install-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Install-SCXAgent** cmdlet installs the Operations Manager agent for discovered UNIX and Linux computers, signs the certificate for WS-Management communication, and registers the agent with Operations Manager for management. UNIX and Linux computers should be discovered with the **Invoke-SCXDiscovery** cmdlet. The output object of the **Invoke-SCXDiscovery** cmdlet is provided as the **DiscoveryResult** input for the **Install-SCXAgent** cmdlet.

For information about discovering UNIX or Linux computers, type `Get-Help Invoke-SCXDiscovery`.

This cmdlet returns an array of managed UNIX or Linux computer objects that represent the targeted systems that were successfully installed.

Parameters

-ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully-qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

-Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

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

-DiscoveryResult<DiscoveryResult[]>

Specifies the DiscoveryResult object to process for management agent installation as returned by the **Invoke-SCXDiscovery** cmdlet.

For more information about discovering targeted systems, type `Get-Help Invoke-SCXDiscovery`.

Aliases	none
Required?	true
Position?	1
Default Value	none

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

-SCSession<Connection[]>

Specifies a 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.

Outputs

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

- **Microsoft.Unix.Computer[]**
  This cmdlet returns an array of managed UNIX and Linux computers that were successfully installed.

Examples

-------------------------- EXAMPLE 1 --------------------------

Description

This command invokes a discovery of UNIX and Linux computers within a range of IP Addresses and install the management agent.

```
C:\PS>$WSCredential = Get-Credential userBob

$SSHCredential=Get-SCXSSHCredential -UserName userBob -Key c:\keys\foo.ppk -ElevationType
sudo

$MyPool = Get-SCOMResourcePool -DisplayName:"pool01"

Invoke-SCXDiscovery -IPRange 192.168.1.50,192.168.1.75  -ResourcePool:$MyPool -
WSManCredential $WSCredential -SSHCredential $SSHCredential | Install-SCXAgent
```

Invoke-SCXDiscovery

Invoke-SCXDiscovery

Invokes the discovery operation for the specified configuration of UNIX and Linux computers.

Syntax

```
Parameter Set: Empty
Invoke-SCXDiscovery [-SCSession <Connection[]> ] [ <CommonParameters>]

Parameter Set: FromAgentNames
Invoke-SCXDiscovery [-SCSession <Connection[]> ] [ <CommonParameters>]

Parameter Set: FromHostNameAndManagementServerParameterSetName
Invoke-SCXDiscovery [-Name] <String[]> -WsManCredential <PSCredential> [-ComputerName
<String[]> ] [-Credential <PSCredential> ] [-SshCredential <PSCredential> ] [-SshPort
<Int32> ] [ <CommonParameters>]

Parameter Set: FromHostNameAndResourcePoolParameterSetName
Invoke-SCXDiscovery [-Name] <String[]> -ResourcePool <ManagementServicePool> -
WsManCredential <PSCredential> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-
SshCredential <PSCredential> ] [-SshPort <Int32> ] [ <CommonParameters>]

Parameter Set: FromIPRangeAndManagementServerParameterSetName
Invoke-SCXDiscovery [-IPRange] <IPAddress[]> -WsManCredential <PSCredential> [-ComputerName
<String[]> ] [-Credential <PSCredential> ] [-SshCredential <PSCredential> ] [-SshPort
<Int32> ] [ <CommonParameters>]

Parameter Set: FromIPRangeAndResourcePoolParameterSetName
Invoke-SCXDiscovery [-IPRange] <IPAddress[]> -ResourcePool <ManagementServicePool> -
WsManCredential <PSCredential> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-
SshCredential <PSCredential> ] [-SshPort <Int32> ] [ <CommonParameters>]

Parameter Set: FromManagementServer
Invoke-SCXDiscovery [-SCSession <Connection[]> ] [ <CommonParameters>]
```

Detailed Description

The **Invoke-SCXDiscovery** cmdlet invokes the discovery operation for the specified configuration of UNIX and Linux computers.

The operation uses an SSH credential object for privileged SSH installation actions and an optional Windows PowerShell credential object for low-privileged WS-Management agent communication. If the agent has been manually installed on the targeted hosts, only the low-privileged WS-Management credential is required.

The output of this cmdlet is an array of DiscoveryResult objects representing the discovery result for each targeted computer for which discovery was attempted.

Parameters

-ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

-Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

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

-IPRange<IPAddress[]>

Specifies a range of IP Addresses that will be used for discovery. The *IPRange* must contain two IP Addresses (for example, 192.168.0.1,192.168.0.254). Either the *IPRange* or *Name* parameters must be specified.

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

-Name<String[]>

Specifies a list of valid host names, FQDNs, or IP Addresses (separated by a comma) that will be used for discovery. Either the *IPRange* or *Name* parameters must be specified. This parameter is accepted from the pipeline.

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

-ResourcePool<ManagementServicePool>

Specifies a resource pool of servers, one of which is assigned to be the current management server and the others serving as backup management servers. This parameter requires a resource pool object and returns only the managed computers in that resource pool.

For information about how to get a resource pool object, type `Get-Help Get-SCOMResourcePool`.

Aliases	none
Required?	true
Position?	named

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

-SCSession<Connection[]>

Specifies a connection.

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

-SshCredential<PSCredential>

Specifies a privileged SSH credential used to perform the agent installation actions. If this parameter is not specified, the discovery fails unless an agent has been manually installed on the targeted computer.

For information about how to get an SSH credential object, type `Get-Help Get-SCXSSHCredential`.

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

-SshPort<Int32>

Specifies the port to use for SSH communication with the host. If not specified, the default value of 22 is used.

Aliases	none

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

-WsManCredential<PSCredential>

Specifies a credential used for low-privileged agent communication through WS-Management.

Type a user name, such as "User01" or enter a **PSCredential** object, such as one that is returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

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

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see about_CommonParameters.

Outputs

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

- **DiscoveryResult**This cmdlet returns an array of **DiscoveryResult** objects representing the discovery result for each targeted computer for which discovery was planned.

Examples

-------------------------- EXAMPLE 1 --------------------------

Description

This command discovers UNIX and Linux computers within a range of IP addresses.

```
C:\PS>$WSCredential = Get-Credential userBob
```

```
$SSHCredential=Get-SCXSSHCredential -UserName userBob -Key c:\keys\mykey.ppk -ElevationType sudo
```

```
$MyPool = Get-SCOMResourcePool "pool01"
```

```
Invoke-SCXDiscovery -IPRange 192.168.1.50,192.168.1.75  -ResourcePool $MyPool -WSManCredential $WSCredential -SSHCredential $SSHCredential
```

-------------------------- EXAMPLE 2 --------------------------

Description

This command discovers UNIX and Linux computers by name using a resource pool and specifying a non-standard SSH port.

```
C:\PS>$WSCredential = Get-Credential userBob
```

```
$SSHCredential=Get-SCXSSHCredential -UserName userBob -Key c:\keys\userBob.ppk -ElevationType sudo
```

```
$MyPool = Get-SCOMResourcePool -DisplayName:"pool01"
```

```
Invoke-SCXDiscovery -Name "nx1.contoso.com,nx2.contoso.com,nx3.contoso.com" -SSHPort 8022  -ResourcePool $MyPool -WSManCredential $WSCredential -SSHCredential $SSHCredential
```

-------------------------- EXAMPLE 3 --------------------------

Description

This command discovers UNIX and Linux computers by name where the management agent has already been installed.

```
C:\PS>$MyPool = Get-SCOMResourcePool "RP1"
```

```
Invoke-SCXDiscovery -Name "nx1.contoso.com,nx2.contoso.com,nx3.contoso.com" -ResourcePool $MyPool -WSManCredential userBob
```

Remove-SCXAgent

Remove-SCXAgent

Removes the targeted managed UNIX and Linux computers from the management group.

Syntax

```
Parameter Set: Default
Remove-SCXAgent [-Agent] <IPersistedUnixComputer[]> [-ComputerName <String[]> ] [-Credential
<PSCredential> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: Empty
Remove-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromAgentNames
Remove-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromManagementServer
Remove-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Remove-SCXAgent** cmdlet removes the targeted managed UNIX and Linux computers from the management group. Managed UNIX and Linux computers are passed to the **Remove-SCXAgent** cmdlet as an array of UNIX and Linux computer objects.

For information about retrieving managed UNIX and Linux computers, type `Get-Help Get-SCXAgent`.

This cmdlet does not return any output.

Parameters

-Agent<IPersistedUnixComputer[]>

Specifies one or more managed UNIX and Linux computer objects. For information about how to get a managed UNIX or Linux computer object, type `Get-Help Get-SCXAgent`.

Aliases	none
Required?	true
Position?	1

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

-ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

-Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as the one that is returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

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

-SCSession<Connection[]>

Specifies a connection.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see about_CommonParameters.

Outputs

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

- **None.**

  This cmdlet does not generate any output. There is no failure case for the **Remove-SCXAgent** cmdlet.

Examples

-------------------------- EXAMPLE 1 --------------------------

Description

This command gets managed UNIX and Linux computers with names matching the provided input, and removes them from management.

```
C:\PS>$myAgents = Get-SCXAgent -Name
scxserver1.contoso.com,scxserver2*,*.development.contoso.com
Remove-SCXAgent -Agent $myAgents
```

-------------------------- EXAMPLE 2 --------------------------

Description

This command gets managed UNIX and Linux computers with names matching provided input, and removes them from management by using a pipeline operation.

```
C:\PS>Get-SCXAgent -Name scxserver1.contoso.com,scxserver2*,*.development.contoso.com |
Remove-SCXAgent
```

Set-SCXResourcePool

Set-SCXResourcePool

Changes the managing resource pool for the targeted managed UNIX and Linux computers.

Syntax

```
Parameter Set: FromResourcePoolParameterSetName
Set-SCXResourcePool [-Agent] <IPersistedUnixComputer[]> -ResourcePool
<ManagementServicePool> [-ComputerName <String[]> ] [-Credential <PSCredential> ] [-
SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Set-SCXResourcePool** cmdlet changes the managing resource pool for the targeted managed UNIX and Linux computers.

Parameters

-Agent<IPersistedUnixComputer[]>

Specifies one or more managed UNIX or Linux computer objects. For information about how to get a managed UNIX or Linux computer object, type Get-Help Get-SCXAgent.

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

-ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

-Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one that is returned by the Get-Credential cmdlet. When you type a user name, you are prompted for a password. For more information, type Get-Help Get-Credential.

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

-ResourcePool<ManagementServicePool>

Specifies a resource pool of servers, one of which is assigned to be the current management server and the other serving as backup management servers. This parameter requires a resource pool object and returns only the managed computers in that resource pool.

For information about how to get a resource pool object, type `Get-Help Get-SCOMResourcePool`.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-SCSession<Connection[]>

Specifies a 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 --------------------------

Description

This command moves a single agent to the resource pool named ‘Pool 1’.

```
PS C:\> $MyPool = Get-SCOMResourcePool -DisplayName "Pool 1"
PS C:\> Get-SCXAgent –Name scxserver1.contoso.com | Set-SCXResourcePool -ResourcePool
$MyPool
```

-------------------------- EXAMPLE 2 --------------------------

Description

This command moves all UNIX and Linux agents from ‘Pool 1’ to ‘Pool 2’.

```
PS C:\> $MyPool1 = Get-SCOMResourcePool -DisplayName "Pool 1"
PS C:\> $MyPool2 = Get-SCOMResourcePool -DisplayName "Pool 2"
PS C:\> $MyAgents = Get-SCXAgent -ResourcePool $MyPool1
PS C:\> Set-SCXResourcePool -Agent $MyAgents -ResourcePool $MyPool2
```

Uninstall-SCXAgent

Uninstall-SCXAgent

Uninstalls the management agent on the specified managed UNIX and Linux computers.

Syntax

```
Parameter Set: Default
Uninstall-SCXAgent [-Agent] <IPersistedUnixComputer[]> [-ComputerName <String[]> ] [-
Credential <PSCredential> ] [-SshCredential <PSCredential> ] [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: Empty
Uninstall-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromAgentNames
Uninstall-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromManagementServer
Uninstall-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Uninstall-SCXAgent** cmdlet uninstalls the management agent on the specified UNIX and Linux computers.

The required *Agent* parameter specifies the targeted managed UNIX or Linux computers to uninstall and requires a UNIX or Linux computer object. For information about retrieving managed UNIX or Linux computers, type `Get-Help Get-SCXAgent`. An optional *SSHCredential* parameter can be specified to identify the privileged credential used for uninstallation of the management agent. If the *SSHCredential* is not specified, the cmdlet attempts to use the defined privileged Run As account for the agent.

This cmdlet does not return any output.

Parameters

-Agent<IPersistedUnixComputer[]>

Specifies one or more managed UNIX or Linux computer objects. For information about how to get a managed UNIX or Linux computer object, type `Get-Help Get-SCXAgent`.

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

-ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name (FQDN). To specify the local computer, type the computer name, "localhost", or a dot (.).

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

-Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one that is returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

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

-SCSession<Connection[]>

Specifies a connection.

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

-SshCredential<PSCredential>

Specifies a privileged SSH credential used to perform the agent maintenance actions. If this parameter is not specified, the cmdlet attempts to use the defined agent maintenance Run As account for the agent.

For information about how to get an SSH credential object, type `Get-Help Get-SCXSSHCredential`.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see about_CommonParameters.

Outputs

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

- **None.**
  The output is a list of results, each representing the uninstall status for a computer:
  hostname, uninstall status, description
  *Description includes the details of why uninstall failed.

Examples

-------------------------- EXAMPLE 1 --------------------------

Description

This command uninstalls the agent by using a specified SSH credential.

```
C:\PS>$MyAgent = Get-SCXAgent -Name nx1.contoso.com
$SSHCredential=Get-SCXSSHCredential -UserName userBob -SSHKey c:\keys\key.ppk -ElevationType sudo
Uninstall-SCXAgent -SSHCredential $SSHCredential -Agent $MyAgent
```

-------------------------- EXAMPLE 2 --------------------------

Description

This command uninstalls the agent by using the defined Run As account associated with the agent.

```
C:\PS>Get-SCXAgent -Name nx1.contoso.com | Uninstall-SCXAgent
```

Update-SCXAgent

Update-SCXAgent

Upgrades the management agent on the specified UNIX and Linux computers to the latest version available.

Syntax

```
Parameter Set: Default
Update-SCXAgent [-Agent] <IPersistedUnixComputer[]> -WsManCredential <PSCredential> [-
ComputerName <String[]> ] [-Credential <PSCredential> ] [-SshCredential <PSCredential> ] [-
Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: Empty
Update-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromAgentNames
Update-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromManagementServer
Update-SCXAgent [-SCSession <Connection[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The **Update-SCXAgent** cmdlet updates the agent on the specified computers to the latest version available. If the latest available version of the management agent is already installed a targeted computer, no action is taken.

The required *Agent* parameter specifies the targeted managed UNIX and Linux computers to uninstall and requires a UNIX and Linux computer object. For information about retrieving managed UNIX and Linux computers, type `Get-Help Get-SCXAgent`. This cmdlet accepts *SshCredential* and *WsManCredential* input parameters. The supplied *SshCredential* parameter is used for privileged agent update actions while the supplied *WsManCredential* is used for low-privileged agent communication. If the *SshCredential* parameter is not provided, the cmdlet attempts to use the defined agent maintenance Run As account associated with the targeted computer.

The output of this cmdlet is an array of managed UNIX or Linux computer objects representing the targeted systems that were successfully updated.

Parameters

-Agent<IPersistedUnixComputer[]>

Specifies one or more managed UNIX or Linux computer objects. For information about how to get a managed UNIX or Linux computer object, type `Get-Help Get-SCXAgent`.

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

-ComputerName<String[]>

Specifies the name of a computer to establish a connection with. The computer must be running the System Center Data Access Service. The default value is the computer for the current management group connection. Valid formats include a NetBIOS name, an IP address, or a fully qualified domain name. To specify the local computer, type the computer name, "localhost", or a dot (.).

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

-Credential<PSCredential>

Specifies a user account under which the management group connection will run. The default is the current user. Type a user name, such as "User01", "Domain01\User01", or "User@Domain.com", or enter a PSCredential object, such as one returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

Aliases	none

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

-SCSession<Connection[]>

Specifies a connection.

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

-SshCredential<PSCredential>

Specifies a privileged *SshCredential* parameter used to perform the agent update actions. If this parameter is not specified, the cmdlet attempts to use the defined agent maintenance Run As account for the agent.

For information about how to get an SSH credential object, type `Get-Help Get-SCXSSHCredential`.

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

-WsManCredential<PSCredential>

Specifies a credential used for low-privileged agent communication through WSMan. This cmdlet uses this credential to verify agent availability.

Type a user name, such as "User01" or enter a PSCredential object, such as one that is returned by the **Get-Credential** cmdlet. When you type a user name, you are prompted for a password. For more information, type `Get-Help Get-Credential`.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see about_CommonParameters.

Outputs

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

- **Microsoft.Unix.Computer[]**

  This cmdlet returns a collection of upgrade results, each representing the upgrade status for a computer. Each result must contain the following information:

  Hostname, operating systemS, old Agent version, new Agent version, upgrade status, description

  *Description includes the details of why upgrade failed.

Notes

- This cmdlet takes in a collection of host names in enumerated fashion. It collects all host names and processes the entire collection in parallel at the end of the pipeline.

Examples

-------------------------- EXAMPLE 1 --------------------------

Description

This command updates the management agent by using a specified **SSHCredential** and **WSManCredential**.

```
C:\PS>$MyAgent = Get-SCXAgent -Name nx1.contoso.com
$WSCredential = Get-Credential userBob
$SSHCredential=Get-SCXSSHCredential -UserName userBob -Key c:\keys\key.ppk -ElevationType sudo
Update-SCXAgent -SSHCredential $SSHCredential -WSManCredential $WSCredential -Agent $MyAgent
```

-------------------------- EXAMPLE 2 --------------------------

Description

This command updates the management agent by using the defined Run As account.

```
C:\PS>Get-SCXAgent -Name nx1.contoso.com | Update-SCXAgent
```