

Microsoft System Center 2012 R2

Cmdlet Reference for System Center 2012 R2 App Controller

Microsoft Corporation

Published: November 20, 2013

Applies To

System Center 2012 R2 App Controller

Feedback

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

Copyright

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

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

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

© 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 20, 2013	Initial release of this document.

Contents

Add-SCACAzureDisk	5
Add-SCACAzureImage	10
Add-SCACAzureSubscription	15
Add-SCACCloudSystem	19
Add-SCACShare	24
Export-SCACAesKey	26
Get-SCACAdminSetting	28
Get-SCACAzureHostedService	31
Get-SCACAzureRoleInstance	33
Get-SCACAzureServiceDeployment	34
Get-SCACAzureSubscription	37
Get-SCACCloudSystem	39
Get-SCACJob	42
Get-SCACServer	44
Get-SCACShare	47
Get-SCACTemporaryStorage	49
Get-SCACUserRole	51
New-SCACUserRole	53
New-SCACUserRoleScope	56
Remove-SCACAzureSubscription	59
Remove-SCACCloudSystem	61
Remove-SCACShare	63
Remove-SCACUserRole	65
Resume-SCACServiceDeployment	67
Set-SCACAdminSetting	70

Set-SCACCloudSystem	74
Set-SCACTemporaryStorage	78
Set-SCACUserRole	80
Suspend-SCACServiceDeployment	84

Add-SCACAzureDisk

Add-SCACAzureDisk

Adds a virtual hard disk to Windows Azure.

Syntax

Parameter Set: SourcePath

```
Add-SCACAzureDisk -Cloud <CAzureSubscriptionCloud> -DisplayName <String> -Name <String> -  
OperatingSystem {None | Windows | Linux} -SourcePath <String> -StorageBlob <Uri> [-Force] [  
<CommonParameters>]
```

Parameter Set: SourceVHD

```
Add-SCACAzureDisk -Cloud <CAzureSubscriptionCloud> -DisplayName <String> -Name <String> -  
OperatingSystem {None | Windows | Linux} -SourceVHD <Object> -StorageBlob <Uri> [-Force] [  
<CommonParameters>]
```

Detailed Description

The **Add-SCACAzureDisk** cmdlet adds a virtual hard disk to Windows Azure.

You must import the Virtual Machine Manager (VMM) module into your session to run this cmdlet.

Parameters

-Cloud<CAzureSubscriptionCloud>

Specifies a Windows Azure subscription cloud object. To retrieve a subscription cloud object, use the **Get-SCACAzureSubscription** cmdlet.

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

-DisplayName<String>

Specifies a display name for the image or virtual hard disk in Windows Azure.

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

-Force

Indicates that the image or virtual hard disk is added to Windows Azure without prompting for confirmation.

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

-Name<String>

Specifies the name of the virtual hard disk.

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

-OperatingSystem<OperatingSystemType>

Specifies the operating system type of the virtual hard disk. Valid values are: None, Windows, Linux.

The acceptable values for this parameter are:

None	
Windows	
Linux	

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

-SourcePath<String>

Specifies a path to the source virtual hard disk.

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

-SourceVHD<Object>

Specifies the source virtual hard disk.

Aliases	none
Required?	true

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

-StorageBlob<Uri>

Specifies the URI where the image or virtual hard disk is stored in Windows Azure.

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: Add a VHD to Windows Azure

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command adds the virtual hard disk named VHD01.vhd to the specified container in the subscription cloud stored in \$Subscription.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> Add-SCACAzureDisk -Name "VHD01" -DisplayName "VHD01.vhd" -Cloud $Subscription -
StorageBlob "https://container01.blob.core.windows.net/vhds/VHD01.vhd" -SourcePath
"\\FileShare\VHDS\Vhd01.vhd" -OperatingSystem "Windows" -Force
```

Related topics

[Add-SCACAzureImage](#)

[Get-SCACAzureSubscription](#)

Add-SCACAzureImage

Add-SCACAzureImage

Adds a virtual hard drive to the Windows Azure image store.

Syntax

Parameter Set: SourcePath

```
Add-SCACAzureImage -Cloud <ACAzureSubscriptionCloud> -DisplayName <String> -Name <String> -  
OperatingSystem {Windows | Linux} -SourcePath <String> -StorageBlob <Uri> [-Force] [  
<CommonParameters>]
```

Parameter Set: SourceVHD

```
Add-SCACAzureImage -Cloud <ACAzureSubscriptionCloud> -DisplayName <String> -Name <String> -  
OperatingSystem {Windows | Linux} -SourceVHD <Object> -StorageBlob <Uri> [-Force] [  
<CommonParameters>]
```

Detailed Description

The **Add-SCACAzureImage** cmdlet adds a virtual hard drive to the Windows Azure image store.

You must import the Virtual Machine Manager (VMM) module into your session to run this cmdlet.

Parameters

-Cloud<ACAzureSubscriptionCloud>

Specifies a Windows Azure subscription cloud object. To retrieve a subscription cloud object, use the **Get-SCACAzureSubscription** cmdlet.

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

-DisplayName<String>

Specifies a display name for the image in Windows Azure.

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

-Force

Indicates that the image is added to the Windows Azure image store without prompting for confirmation.

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

-Name<String>

Specifies the name of the image.

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

-OperatingSystem<OperatingSystemType>

Specifies the operating system type for the virtual hard disk. Valid values are: Windows, Linux.

The acceptable values for this parameter are:

Windows	
Linux	

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

-SourcePath<String>

Specifies a path to the source image.

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

-SourceVHD<Object>

Specifies the source virtual hard disk object.

Aliases	none
Required?	true
Position?	named

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

-StorageBlob<Uri>

Specifies the URI where the image is stored in Windows Azure.

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: Add a VHD image to the Windows Azure image store

The first command gets the Windows Azure subscription object named "AzureSubscription01" and stores the object in the \$Subscription variable.

The second command gets the virtual hard disk object named VHD01.vhd and stores the object in the \$VHD variable.

The last command adds the VHD image to the image store and names it Image01.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> $VHD = Get-SCVirtualHardDisk -Name "VHD01.vhd"
PS C:\> Add-SCACAzureImage -Name "Image01" -DisplayName "Image01" -Cloud $Subscription -
StorageBlob "http://container01.blob.core.windows.net/vhds/VHD01.vhd" -SourceVHD $VHD -
OperatingSystem "Windows"
```

Related topics

[Get-SCACAzureSubscription](#)

[Add-SCACAzureDisk](#)

Add-SCACAzureSubscription

Add-SCACAzureSubscription

Adds a Windows Azure subscription to App Controller.

Syntax

Parameter Set: Default

```
Add-SCACAzureSubscription [-Name] <String> [-Id] <Guid> [-ManagementCertificatePath] <String> [-ManagementCertificatePassword] <SecureString> [[-Description] <String> ] [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Add-SCACAzureSubscription** cmdlet adds a Windows Azure subscription to App Controller.

Parameters

-Description<String>

Provides a description for the Windows Azure subscription.

Aliases	none
Required?	false
Position?	5
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-Id<Guid>

Specifies a GUID that represents the ID for a Windows Azure subscription.

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

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

-ManagementCertificatePassword<SecureString>

Specifies a secure string that contains a password for the management certificate.

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

-ManagementCertificatePath<String>

Specifies a file path to the management certificate.

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

-Name<String>

Specifies a name for the Windows Azure subscription.

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

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

-PassThru

Returns a Windows Azure subscription object. By default, this cmdlet does not generate 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, System.Guid, System.String, System.Security.SecureString, Microsoft.SystemCenter.CloudManager.PowerShell.ACServer**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACAzureSubscriptionCloud**

Examples

Example 1: Add a Windows Azure subscription

The first command creates a secure string for the provided password and stores the secure string in the \$MCPassword variable.

The second command adds the Windows Azure subscription with the ID of 1626950e-3af6-4fe8-a7d2-e489c18931a2 using the password stored in \$MCPassword and gives it the name AzureSubscription01.

```
PS C:\> $MCPassword = ConvertTo-SecureString "PassWord!" -AsPlainText -Force
PS C:\> Add-SCACAzureSubscription -Id "1626950e-3af6-4fe8-a7d2-e489c18931a2" -
ManagementCertificatePassword $MCPassword -ManagementCertificatePath "C:\CertificatePath" -
Name "AzureSubscription01"
```

Related topics

[Remove-SCACAzureSubscription](#)

Add-SCACCloudSystem

Add-SCACCloudSystem

Adds a connection to a VMM management server or service provider.

Syntax

Parameter Set: SPF

```
Add-SCACCloudSystem [-Name] <String> [-ServiceLocation] <Uri> [-CertificateFilePath] <String> [-CertificatePassword] <SecureString> -SPF [-Description <String> ] [-PassThru] [ <CommonParameters>]
```

Parameter Set: VMM

```
Add-SCACCloudSystem [-Name] <String> [-ServerName] <String> [-Port] <Int32> -VMM [-Description <String> ] [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Add-SCACCloudSystem** cmdlet adds a new connection to a VMM management server or service provider.

Parameters

-CertificateFilePath<String>

Specifies a file path to the certificate.

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

-CertificatePassword<SecureString>

Specifies a secure string that contains a password for the certificate.

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

-Description<String>

Provides a description for the connection.

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

-Name<String>

Specifies a name for the cloud system.

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

-PassThru

Returns a cloud system object. By default, this cmdlet does not generate output.

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

-Port<Int32>

Specifies a port to use for the connection.

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

-ServerName<String>

Specifies the name of a VMM management server.

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

-ServiceLocation<Uri>

Specifies the Service Provider Foundation service location URI.

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

-SPF

Indicates that the connection is made with Service Provider Foundation.

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

-VMM

Indicates that the connection is made with a VMM management server.

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

<CommonParameters>

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

Examples

Example 1: Add a connection to a VMM management server

This command adds a connection named CloudSystem01 to the VMM management server named VMMServer01 using port 8100.

```
PS C:\> Add-SCACCloudsystem -Name "VMMServer01" -VMM -ServerName  
"VMMServer01.Contoso.com" -Port 8100
```

Example 2: Add a connection to a service provider

The first command creates a secure string for the provided password and stores the secure string in the \$Password variable.

The second command adds a service provider connection.

```
PS C:\> $Password = ConvertTo-SecureString "PassWord!" -AsPlainText -Force  
PS C:\> Add-SCACCloudsystem -Name "ServiceProvider01" -ServiceLocation  
"https://Server01.contoso.com:8090/SC2012/VMM/Microsoft.Management.OData.svc/4deca2d5-6169-  
49ca-aa7a-efd6e4b372a3" -CertificateFilePath "\\FileServer01\Certs\TenantCert.pfx" -  
CertificatePassword $Password -SPF
```

Related topics

[Get-SCACCloudSystem](#)

[Set-SCACCloudSystem](#)

[Remove-SCACCloudSystem](#)

Add-SCACShare

Add-SCACShare

Adds a library share.

Syntax

Parameter Set: Default

```
Add-SCACShare [-Path] <Uri> [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Add-SCACShare** cmdlet adds a library share.

Parameters

-PassThru

Returns a share path object. By default, this cmdlet does not generate output.

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

-Path<Uri>

Specifies the path of the share to add.

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](#).

Examples

Example 1: Add a library share

This command adds the share named \\LibraryServer\LibraryPath.

```
PS C:\> Add-SCACShare -Path "\\LibraryServer\LibraryPath"
```

Related topics

[Get-SCACShare](#)

[Remove-SCACShare](#)

Export-SCACAesKey

Export-SCACAesKey

Exports the App Controller AES key from the registry to the specified file.

Syntax

Parameter Set: Default

```
Export-SCACAesKey [-Path] <String> [-Password] <SecureString> [ <CommonParameters>]
```

Detailed Description

The **Export-SCACAesKey** cmdlet exports the App Controller Advanced Encryption Standard (AES) key from the registry to the specified file. Use the *Path* parameter to specify the destination file.

You must run the App Controller command shell as Administrator when using this cmdlet. This cmdlet must be run on the computer on which App Controller is installed.

NOTE: Prior to running this cmdlet, you must create a connection to the App Controller server by using the [Get-SCACServer](#) cmdlet.

Parameters

-Password<SecureString>

Specifies a secure string that contains a password.

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

-Path<String>

Specifies a file location path.

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, System.Security.SecureString**

Examples

Example 1: Export the AES key to a file

The first command prompts you to supply credentials with permission to connect with the App Controller server, and stores the credentials in the \$Credentials parameter.

The second command establishes a connection to the App Controller server named AppControllerSvr01 using the credentials stored in \$Credentials.

The third command creates a secure string for the provided password and stores the secure string in the \$Password variable.

The last command exports the App Controller AES key to the Key.txt file at the specified file path using the password stored in \$Password.

```
PS C:\> $Credentials = Get-Credential
PS C:\> Get-SCACServer -ServerName "https://AppControllerSvr01.Contoso.com" -Credential
$Credentials
PS C:\> $Password = ConvertTo-SecureString "PassWord!" -AsPlainText -Force
PS C:\> Export-SCACAESKey -Path "C:\Keys\Key.txt" -Password $Password
```

Related topics

[Get-SCACServer](#)

Get-SCACAdminSetting

Get-SCACAdminSetting

Gets one or more administrator settings from the App Controller server.

Syntax

Parameter Set: CEIP

```
Get-SCACAdminSetting [-CEIPEnabled] [ <CommonParameters>]
```

Parameter Set: JobHistoryPeriod

```
Get-SCACAdminSetting [-JobHistoryPeriodInDays] [ <CommonParameters>]
```

Parameter Set: RefreshInterval

```
Get-SCACAdminSetting [-RefreshIntervalInSeconds] [ <CommonParameters>]
```

Detailed Description

The **Get-SCACAdminSetting** cmdlet gets one or more administrator settings from the App Controller server. If a setting name is specified by using the *Name* parameter, that setting is retrieved. If no setting is specified, all settings on the server are returned.

Parameters

-CEIPEnabled

Indicates that the value for the CEIPEnabled setting is displayed.

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

-JobHistoryPeriodInDays

Indicates that the value for the job history period setting is displayed.

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

-RefreshIntervalInSeconds

Indicates that the value for the refresh interval setting is displayed.

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**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACAdminSetting**

Examples

Example 1: Get all administrator settings

This command gets all administrator settings and displays information about the settings for the user.

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

Example 2: Get the CEIPEabled administrator setting

This command gets the CEIPEabled administrator setting and displays information about the setting for the user.

```
PS C:\>Get-SCACAdminSetting -CEIPEabled
```

Related topics

[Set-SCACAdminSetting](#)

Get-SCACAzureHostedService

Get-SCACAzureHostedService

Gets the hosted services for a specified Windows Azure subscription.

Syntax

Parameter Set: Default

```
Get-SCACAzureHostedService [-Subscription] <ACAzureSubscriptionCloud> [ <CommonParameters>]
```

Detailed Description

The **Get-SCACAzureHostedService** cmdlet gets the hosted services for a specified Windows Azure subscription.

Parameters

-Subscription<ACAzureSubscriptionCloud>

Specifies a Windows Azure subscription. To get a Windows Azure subscription, use the **Get-SCACAzureSubscription** 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.SystemCenter.CloudManager.PowerShell.ACAzureSubscriptionCloud**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACAzureHostedService**

Examples

Example 1: Get all Windows Azure hosted services for a subscription

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command gets all Windows Azure hosted services for the subscription stored in \$Subscription and displays information about the Windows Azure hosted services to the user.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"  
PS C:\> Get-SCACAzureHostedService -Subscription $Subscription
```

Related topics

[Get-SCACAzureSubscription](#)

Get-SCACAzureRoleInstance

Get-SCACAzureRoleInstance

Gets all role instances running in Windows Azure.

Syntax

```
Get-SCACAzureRoleInstance [ <CommonParameters> ]
```

Detailed Description

The **Get-SCACAzureRoleInstance** cmdlet gets all role instances running in Windows Azure. This cmdlet is for internal use only and is not meant to be used directly by Administrators.

Parameters

<CommonParameters>

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

Examples

PS C:\> # The Get-SCACAzureRoleInstance cmdlet is for internal use only.

Get-SCACAzureServiceDeployment

Get-SCACAzureServiceDeployment

Gets a specified Windows Azure service deployment.

Syntax

Parameter Set: Cloud

```
Get-SCACAzureServiceDeployment [-Subscription] <CAzureSubscriptionCloud> [  
<CommonParameters>]
```

Parameter Set: HostedService

```
Get-SCACAzureServiceDeployment [-HostedService] <CAzureHostedService> [[-DeploymentSlot]  
<String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-SCACAzureServiceDeployment** cmdlet gets a Windows Azure service deployment.

Parameters

-DeploymentSlot<String>

Specifies the name of a deployment slot.

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

-HostedService<ACAzureHostedService>

Specifies a Windows Azure hosted service object. To retrieve a Windows Azure hosted service object, use the **Get-SCACAzureHostedService** cmdlet.

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

-Subscription<ACAzureSubscriptionCloud>

Specifies a Windows Azure subscription object. To obtain a Windows Azure subscription object, use the **Get-SCACAzureSubscription** 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.SystemCenter.CloudManager.PowerShell.ACAzureHostedService, System.String**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACAzureServiceDeployment**

Examples

Example 1: Get the Windows Azure service deployment for a Windows Azure hosted service

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command gets all hosted service objects for the subscription stored in \$Subscription, and then stores the hosted service objects in the \$HostedService variable.

The last command gets the Windows Azure service deployment for the first hosted service stored in the \$HostedService array and displays information about the service deployment to the user.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> $HostedService = Get-SCACAzureHostedService -Subscription $Subscription
PS C:\> Get-SCACAzureServiceDeployment -HostedService $HostedService[0]
```

Example 2: Get all Windows Azure Service Deployments for a Windows Azure subscription

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command gets all of the Windows Azure service deployments for the subscription stored in \$Subscription.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> Get-SCACAzureServiceDeployment -Subscription $Subscription
```

Related topics

[Resume-SCACServiceDeployment](#)

[Suspend-SCACServiceDeployment](#)

[Get-SCACAzureSubscription](#)

[Get-SCACAzureHostedService](#)

Get-SCACAzureSubscription

Get-SCACAzureSubscription

Gets a Windows Azure subscription.

Syntax

Parameter Set: SearchById

```
Get-SCACAzureSubscription [-Id] <Guid> [ <CommonParameters>]
```

Parameter Set: SearchByName

```
Get-SCACAzureSubscription [-Name] <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-SCACAzureSubscription** cmdlet gets one or more Windows Azure subscriptions. You can get an individual subscription by its name or ID.

Parameters

-Id<Guid>

Specifies the ID, as a GUID, for a Windows Azure subscription.

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

-Name<String>

Specifies the name of a Windows Azure subscription.

Aliases	none
Required?	false
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](#).

Examples

Example 1: Get a Windows Azure subscription by its name

This command gets the Windows Azure subscription named AzureSubscription01 by its name.

```
PS C:\> Get-SCACAzureSubscription -Name "AzureSubscription01"
```

Related topics

[Add-SCACAzureSubscription](#)

[Remove-SCACAzureSubscription](#)

Get-SCACCloudSystem

Get-SCACCloudSystem

Gets a cloud system on an App Controller server.

Syntax

Parameter Set: SearchByName

```
Get-SCACCloudSystem [-Name] <String> ] [ <CommonParameters>]
```

Parameter Set: SearchByServer

```
Get-SCACCloudSystem -ServerName <String> [ <CommonParameters>]
```

Detailed Description

The **Get-SCACCloudSystem** cmdlet gets one or more cloud systems on an App Controller server. If a cloud system is specified by using the Name parameter, **Get-SCACCloudSystem** retrieves the specified cloud system. If no cloud system is specified, **Get-SCACCloudSystem** retrieves all cloud systems on the App Controller server.

Parameters

-Name<String>

Specifies the name of a cloud system.

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

-ServerName<String>

Specifies the name of the server for a cloud system.

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](#).

Inputs

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

- **System.String**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACCloudSystem**

Examples

Example 1: Get all cloud systems on the App Controller server

This command gets all cloud systems on the App Controller server.

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

Example 2: Get a cloud system by its server name

This command gets the cloud system object on the server named Management.Core.Windows.Net.

```
PS C:\> Get-SCACCLOUDSystem -ServerName "Management.Core.Windows.Net"
```

Related topics

[Add-SCACCloudSystem](#)

[Remove-SCACCloudSystem](#)

Get-SCACJob

Get-SCACJob

Gets App Controller jobs.

Syntax

Parameter Set: AllJobs

```
Get-SCACJob [[-All]] [ <CommonParameters>]
```

Parameter Set: SingleJob

```
Get-SCACJob [-Id] <Guid> ] [ <CommonParameters>]
```

Detailed Description

The **Get-SCACJob** cmdlet gets App Controller jobs. You can get all jobs, or a single job by its ID. If neither the *All* nor the *ID* parameter is specified, then all jobs started in the previous 48 hours are returned.

Parameters

-All

Indicates that all job objects are returned.

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

-Id<Guid>

Specifies a GUID that identifies a job.

Aliases	none
Required?	false
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.Management.Automation.SwitchParameter, System.Guid**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACJob**

Examples

Example 1: Get all App Controller jobs

This command gets all App Controller jobs.

```
PS C:\> Get-SCACJob -All
```

Example 2: Get a job by its ID

This command gets the App Controller job with the ID of bb4915e4-e4bf-499d-8126-b6e80c63daff.

```
PS C:\> Get-SCACJob -ID "bb4915e4-e4bf-499d-8126-b6e80c63daff"
```

Get-SCACServer

Get-SCACServer

Gets and establishes a connection with an App Controller server.

Syntax

Parameter Set: Default

```
Get-SCACServer [-ServerName] <String> [[-Credential] <PSCredential> ] [[-UserRole] <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-SCACServer** cmdlet gets and establishes a connection with the specified App Controller server. You must run this cmdlet prior to running any other App Controller cmdlets.

Parameters

-Credential<PSCredential>

Specifies a user account that has permission to perform this action. To generate a PSCredential object, use the **Get-Credential** cmdlet.

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

-ServerName<String>

Specifies the name of the App Controller Server.

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

-UserRole<String>

Specifies the name of a user role.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	false
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, System.Management.Automation.PSCredential, System.String**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACServer**

Examples

Example 1: Connect to an App Controller server

The first command prompts you to supply credentials with permission to connect with the App Controller server, and stores the credentials in the \$Credentials parameter.

The second command establishes a connection to the App Controller server named AppControllerSvr01 using the credentials stored in \$Credentials.

```
PS C:\> $Credentials = Get-Credential
PS C:\> Get-SCACServer -ServerName "https://AppControllerSvr01.Contoso.com" -Credential
$Credentials
```

Example 2: Connect to an App Controller server and store the connection in a variable

The first command prompts you to supply credentials with permission to connect with the App Controller server, and stores the credentials in the \$Credentials parameter.

The second command establishes a connection to the App Controller server named AppControllerSvr01 using the credentials stored in \$Credentials, and then stores the connection in the \$ACServer variable. You can then use \$ACServer to provide the server information to other cmdlets.

The last command displays information about the App Controller server stored in \$ACServer to the user.

```
PS C:\> $Credentials = Get-Credential
PS C:\> $ACServer = Get-SCACServer -ServerName "https://AppControllerSvr01.Contoso.com" -
Credential $Credentials
PS C:\> $ACServer
```

Example 3: Connect to an App Controller server with a user role

The first command prompts you to supply credentials with permission to connect with the App Controller server, and stores the credentials in the \$Credentials parameter.

The second command establishes a connection to the App Controller server named AppControllerSvr01 with the user role UserRole01 using the credentials stored in \$Credentials.

```
PS C:\> $Credentials = Get-Credential
PS C:\> Get-SCACServer -ServerName "https://AppControllerSvr01.Contoso.com" -Credential
$Credentials -UserRole "UserRole01"
```

Get-SCACShare

Get-SCACShare

Gets all shares in App Controller.

Syntax

```
Get-SCACShare [ <CommonParameters> ]
```

Detailed Description

The **Get-SCACShare** cmdlet gets all shares that have been added to App Controller. Both library shares and shares added as temporary storage are returned.

To add a library share, use the **Add-SCACShare** cmdlet.

Parameters

<CommonParameters>

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

Examples

Example 1: Get all shares in App Controller

This command returns all shares in App Controller.

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

Example 2: Get a specific share by its name

The first command gets the library share object named \\LibraryServer\LibraryShare and stores the object in the \$Share variable.

The second command displays the information stored about the library share in \$Share to the user.

```
PS C:\> $Share = Get-SCACShare | where {$_.SharePath -eq "\\LibraryServer\LibraryShare"}
PS C:\> $Share
```

Related topics

[Add-SCACShare](#)

[Remove-SCACShare](#)

Get-SCACTemporaryStorage

Get-SCACTemporaryStorage

Gets temporary storage used by the App Controller server.

Syntax

```
Get-SCACTemporaryStorage [ <CommonParameters> ]
```

Detailed Description

The **Get-SCACTemporaryStorage** cmdlet gets the temporary storage used by the App Controller server.

To set temporary storage, use the **Set-SCACTemporaryStorage** cmdlet.

Parameters

<CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -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.SystemCenter.CloudManager.PowerShell.ACLibraryFileShare**

Examples

Example 1: Get temporary storage

This command returns the temporary storage for the App Controller server.

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

Related topics

[Set-SCACTemporaryStorage](#)

Get-SCACUserRole

Get-SCACUserRole

Gets user roles on the App Controller server.

Syntax

Parameter Set: Default

```
Get-SCACUserRole [-Managed] [-Name <String> ] [ <CommonParameters>]
```

Detailed Description

The **Get-SCACUserRole** cmdlet gets user roles on the App Controller server.

Parameters

-Managed

Indicates that only managed user roles are returned.

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

-Name<String>

Specifies the name of a user role.

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.Management.Automation.SwitchParameter, System.Management.Automation.SwitchParameter**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACUserRole**

Examples

Example 1: Get all managed user roles

This command gets all managed App Controller user roles on the App Controller server, and displays information about each user role to the user.

```
PS C:\> Get-SCACUserRole -Managed
```

Example 2: Get a user role by name

This command gets the user role named "UserRole01".

```
PS C:\> Get-SCACUserRole -Name "UserRole01"
```

Related topics

[New-SCACUserRole](#)

[Set-SCACUserRole](#)

New-SCACUserRole

New-SCACUserRole

Creates a user role in App Controller.

Syntax

Parameter Set: Default

```
New-SCACUserRole [-Name] <String> [[-Description] <String> ] [[-IsReadOnly] <Boolean> ] [ <CommonParameters> ]
```

Detailed Description

The **New-SCACUserRole** cmdlet creates a user role in App Controller. This cmdlet creates the user role and allows you to set it as read-only. To add or remove members or scope from the user role, use the **Set-SCACUserRole** cmdlet. The user role is not added to App Controller until you have set the user role.

Parameters

-Description<String>

Provides a description of the user role.

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

-IsReadOnly<Boolean>

Determines whether the user role is read only.

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

-Name<String>

Specifies a name for the user role.

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](#).

Examples

Example 1: Create a user role

This command creates a user role named UserRole01 and stores the user role object in the \$UserRole01 variable.

```
PS C:\> $UserRole01 = New-SCACUserRole -Name "UserRole01" -Description "First user role"
```

Example 2: Create a read-only user role

This command creates a read-only user role named UserRole02 and stores the user role object in the \$UserRole02 variable.

```
PS C:\> $UserRole02 = New-SCACUserRole -Name "UserRole02" -Description "Read-only user role"
-IsReadOnly $True
```

Related topics

[Get-SCACUserRole](#)

[Set-SCACUserRole](#)

New-SCACUserRoleScope

New-SCACUserRoleScope

Creates a user role scope object.

Syntax

Parameter Set: Azure

```
New-SCACUserRoleScope [-AzureCloud] <ACAzureSubscriptionCloud[]> [ <CommonParameters>]
```

Parameter Set: SPF

```
New-SCACUserRoleScope [-SPFCloudSystem] <ACCloudSystem> [-VMMUserRoleName] <String> [ <CommonParameters>]
```

Detailed Description

The **New-SCACUserRoleScope** cmdlet creates a user role scope object that you can use to add scope to a user role.

Parameters

-AzureCloud<ACAzureSubscriptionCloud[]>

Specifies a Windows Azure subscription object. To get a Windows Azure subscription object, use the **Get-SCACAzureSubscription** cmdlet.

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

-SPFCloudSystem<ACCloudSystem>

Specifies a Service Provider Foundation cloud system object. To get a cloud system object, use the **Get-SCACCloudSystem** cmdlet.

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

-VMMUserRoleName<String>

Specifies the name of a Virtual Machine Manager user role.

Aliases	none
Required?	true
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](#).

Examples

Example 1: Create a user role scope object by using a Windows Azure subscription

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command creates a user role scope that includes the Windows Azure subscription stored in \$Subscription and then stores the user role scope in the \$UserRoleScope01 variable.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"  
PS C:\> $UserRoleScope01 = New-SCACUserRoleScope -AzureCloud $Subscription
```

Related topics

[Set-SCACUserRole](#)

Remove-SCACAzureSubscription

Remove-SCACAzureSubscription

Removes a Windows Azure subscription from App Controller.

Syntax

Parameter Set: Default

```
Remove-SCACAzureSubscription [-Subscription] <CAzureSubscriptionCloud> [-PassThru] [  
<CommonParameters>]
```

Detailed Description

The **Remove-SCACAzureSubscription** cmdlet removes a Windows Azure subscription from App Controller.

Parameters

-PassThru

Returns a Windows Azure subscription object. By default, this cmdlet does not generate output.

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

-Subscription<CAzureSubscriptionCloud>

Specifies a Windows Azure subscription object. To retrieve a subscription object, use the **Get-SCACAzureSubscription** cmdlet.

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

<CommonParameters>

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

Examples

Example 1: Remove a Windows Azure subscription

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command removes the Windows Azure subscription stored in \$Subscription.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> Remove-SCACAzureSubscription -Subscription $Subscription
```

Related topics

[Add-SCACAzureSubscription](#)

[Get-SCACAzureSubscription](#)

Remove-SCACCloudSystem

Remove-SCACCloudSystem

Removes a connection to a cloud system.

Syntax

Parameter Set: Default

```
Remove-SCACCloudSystem [-CloudSystem] <ACCloudSystem> [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Remove-SCACCloudSystem** function removes a connection to a cloud system.

Parameters

-CloudSystem<ACCloudSystem>

Specifies a cloud system object. To retrieve a cloud system object, use the **Get-SCACCloudSystem** cmdlet.

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

-PassThru

Returns a cloud system object. By default, this cmdlet does not generate 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: Remove a cloud system

The first command gets the cloud system object named CloudSystem01 and stores the object in the \$CloudSystem variable.

The second command removes the cloud system object stored in \$CloudSystem.

```
PS C:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01"  
PS C:\> Remove-SCACCloudSystem -CloudSystem $CloudSystem
```

Related topics

[Add-SCACCloudSystem](#)

[Get-SCACCloudSystem](#)

[Set-SCACCloudSystem](#)

Remove-SCACShare

Remove-SCACShare

Removes a library share.

Syntax

Parameter Set: Default

```
Remove-SCACShare [-Share] <ACLlibraryFileShare> [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Remove-SCACShare** cmdlet removes a specified library share.

Parameters

-PassThru

Returns a share path object. By default, this cmdlet does not generate output.

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

-Share<ACLlibraryFileShare>

Specifies a library file share.

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](#).

Examples

Example 1: Remove a library share

The first command gets the library share object named \\LibraryServer\LibraryShare and stores it in the \$Share variable.

The second command removes the library share stored in \$Share.

```
PS C:\> Get-SCACShare | where {$_.SharePath -eq \\LibraryServer\LibraryShare}
PS C:\> Remove-SCACShare -Share $Share
```

Related topics

[Add-SCACShare](#)

[Get-SCACShare](#)

Remove-SCACUserRole

Remove-SCACUserRole

Removes the specified user role.

Syntax

Parameter Set: Default

```
Remove-SCACUserRole [-UserRole] <ACUserRole> [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Remove-SCACUserRole** cmdlet removes the specified user role.

Parameters

-PassThru

Returns a user role object. By default, this cmdlet does not generate output.

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

-UserRole<ACUserRole>

Specifies the user role object to remove. To retrieve a user role, use the **Get-SCACUserRole** cmdlet.

Aliases	none
Required?	true

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

<CommonParameters>

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

Examples

Example 1: Remove a user role

The first command gets the user role object named UserRole01 and stores the object in the \$UserRole variable.

The second command removes the user role stored in \$UserRole.

```
PS C:\> $UserRole = Get-SCACUserRole -Name "UserRole01" -Managed
PS C:\> Remove-SCACUserRole -UserRole $UserRole
```

Related topics

[New-SCACUserRole](#)

[Get-SCACUserRole](#)

[Set-SCACUserRole](#)

Resume-SCACServiceDeployment

Resume-SCACServiceDeployment

Resumes the specified suspended service deployment.

Syntax

Parameter Set: Default

```
Resume-SCACServiceDeployment [-ServiceDeployment] <ACServiceDeployment> [[-RunAsynchronously]] [ <CommonParameters>]
```

Detailed Description

The **Resume-SCACServiceDeployment** cmdlet resumes the specified suspended service deployment. To suspend a service deployment, use the **Suspend-SCACServiceDeployment** cmdlet.

Parameters

-RunAsynchronously

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

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

-ServiceDeployment<ACServiceDeployment>

Specifies a service deployment object. To get a service deployment object, use the **Get-SCACAzureServiceDeployment** 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.SystemCenter.CloudManager.PowerShell.ACServiceDeployment, System.Management.Automation.SwitchParameter**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACServiceDeployment**

Examples

Example 1: Resume a suspended service deployment

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command gets all service deployment objects for the Windows Azure subscription stored in \$Subscription and stores the objects in the \$SvcDeployment variable.

The last command resumes the first suspended service deployment stored in \$SvcDeployment, and runs the command asynchronously so that control returns to the command shell immediately, before the command stops running.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> $SvcDeployment = Get-SCACAzureServiceDeployment -Subscription $Subscription
PS C:\> Resume-SCACServiceDeployment -ServiceDeployment $SvcDeployment[0] -
RunAsynchronously
```

Related topics

[Get-SCACAzureServiceDeployment](#)

[Suspend-SCACServiceDeployment](#)

[Get-SCACAzureSubscription](#)

Set-SCACAdminSetting

Set-SCACAdminSetting

Sets an administrator setting to a specified value.

Syntax

Parameter Set: AdminSettingBySettingObject

```
Set-SCACAdminSetting [-AdminSetting] <ACAdminSetting> [-Value] <Int32> [-PassThru] [  
<CommonParameters>]
```

Parameter Set: CEIP

```
Set-SCACAdminSetting -CEIPEnabled <Int32> [-PassThru] [ <CommonParameters>]
```

Parameter Set: JobHistoryPeriod

```
Set-SCACAdminSetting -JobHistoryPeriodInDays <Int32> [-PassThru] [ <CommonParameters>]
```

Parameter Set: RefreshInterval

```
Set-SCACAdminSetting -RefreshIntervalInSeconds <Int32> [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Set-SCACAdminSetting** cmdlet sets an administrator setting to a specified value.

Parameters

-AdminSetting<ACAdminSetting>

Specifies an administrator setting object. To get an administrator setting object, use the **Get-SCAdminSetting** cmdlet.

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

-CEIPEnabled<Int32>

Specifies whether participation in the Microsoft Customer Experience Improvement Program (CEIP) is enabled. Valid values are:

-- 0 = Not enabled

-- 1 = Enabled

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

-JobHistoryPeriodInDays<Int32>

Specifies the number of days for the job history period.

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

-PassThru

Returns an administrator setting object. By default, this cmdlet does not generate output.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-RefreshIntervalInSeconds<Int32>

Specifies the number of seconds for the refresh interval.

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

-Value<Int32>

Specifies the value for an administrator setting. Valid values are integers.

Aliases	none
Required?	true
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.

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACAdminSetting, System.String, System.Int32**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACAdminSetting**

Examples

Example 1: Set the JobHistoryPeriodInDays administrator setting

This command sets the JobHistoryPeriodInDays administrator setting to 60.

```
PS C:\> Set-SCACAdminSetting -JobHistoryPeriodInDays 60
```

Example 2: Enable the CEIPEnabled administrator setting

This command enables the CEIP setting by setting the CEIPEnabled administrator setting to 1.

```
PS C:\> Set-SCACAdminSetting -CEIPEnabled "1"
```

Example 3: Update the value for the RefreshIntervallInSeconds administrator setting

The first command gets the administrator setting object named RefreshIntervallInSeconds and stores the object in the \$Setting variable.

The second command sets the value for the administrator setting stored in \$Setting (RefreshIntervallInSeconds) to 30.

```
PS C:\> $Setting = Get-SCACAdminSetting -RefreshIntervalInSeconds  
PS C:\> Set-SCACAdminSetting -AdminSetting $Setting -Value 30
```

Related topics

[Get-SCACAdminSetting](#)

Set-SCACCloudSystem

Set-SCACCloudSystem

Updates the properties of a cloud system.

Syntax

Parameter Set: Default

```
Set-SCACCloudSystem [-CloudSystem] <ACCloudSystem> [[-Name] <String> ] [[-Description] <String> ] [[-CertificateFilePath] <String> ] [-CertificatePassword <SecureString> ] [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Set-SCACCloudSystem** cmdlet updates the properties of a cloud system.

Parameters

-CertificateFilePath<String>

Specifies a file path to the certificate.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-CertificatePassword<SecureString>

Specifies a certificate password as a secure string. This parameter is for use with Service Provider Foundation.

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

-CloudSystem<ACCloudSystem>

Specifies a cloud system object. To retrieve a cloud system object, use the **Get-SCACCloudSystem** cmdlet.

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

-Description<String>

Provides a description for the cloud system.

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

-Name<String>

Specifies a name for the cloud system.

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

-PassThru

Returns a cloud system object. By default, this cmdlet does not generate 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: Update the properties of a cloud system

The first command gets the cloud system object named `CloudSystem01` and stores the object in the `$CloudSystem` variable.

The second command updates the name of the cloud system stored in `$CloudSystem` and adds a description.

```
PS C:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01"
PS C:\> Set-SCACCloudSystem -CloudSystem $CloudSystem -Name "CloudSystemTest01" -
Description "This is a test cloud system."
```

Related topics

[Add-SCACCloudSystem](#)

[Get-SCACCloudSystem](#)

[Remove-SCACCloudSystem](#)

Set-SCACTemporaryStorage

Set-SCACTemporaryStorage

Sets temporary storage used by the App Controller server.

Syntax

Parameter Set: SpecifyPath

```
Set-SCACTemporaryStorage [-Path] <String> [ <CommonParameters>]
```

Detailed Description

The **Set-SCACTemporaryStorage** cmdlet sets a share path as temporary storage for use by the App Controller server.

Parameters

-Path<String>

Specifies the path to the share to use as temporary storage.

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, Microsoft.SystemCenter.CloudManager.PowerShell.ACLibraryFileShare**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACLibraryFileShare**

Examples

Example 1: Add temporary storage

This command sets the share named \\ACServer02\ACStorage as temporary storage.

```
PS C:\> Set-SCACTemporaryStorage -Path "\\ACServer02\ACStorage"
```

Related topics

[Get-SCACTemporaryStorage](#)

Set-SCACUserRole

Set-SCACUserRole

Sets a user role.

Syntax

Parameter Set: Default

```
Set-SCACUserRole [-UserRole] <PSObject> [[-AddMembers] <String[]> ] [[-AddScope] <ACUserRoleScope[]> ] [[-RemoveMembers] <String[]> ] [[-RemoveScope] <PSObject[]> ] [-PassThru] [ <CommonParameters>]
```

Detailed Description

The **Set-SCACUserRole** cmdlet sets a user role. You can add or remove members and scope from the user role. A new user role is not created until you set it with this cmdlet. To create a user role, use the **New-SCACUserRole** cmdlet.

After you have run **Set-SCACUserRole** on a user role or user role scope, you cannot run this cmdlet on that user role or user role scope again. To make additional updates, get the property values from the user role or user role scope by using the **Get-SCACUserRole** cmdlet, and then create a new user role scope with those property values and any additional changes.

Parameters

-AddMembers<String[]>

Specifies an array of members to add to the user role. You can add a user or group. Use the format domain\name.

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

-AddScope<ACUserRoleScope[]>

Specifies a user role scope object. To create a user role scope, use the **New-SCACUserRoleScope** cmdlet.

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

-PassThru

Returns a user role object. By default, this cmdlet does not generate output.

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

-RemoveMembers<String[]>

Specifies an array of members to remove from the user role. You can remove a user or group. Use the format domain\name.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-RemoveScope<PSObject[]>

Specifies the scope object to remove from the user role.

Aliases	none
Required?	false
Position?	5
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-UserRole<PSObject>

Specifies a user role object. To get a user role object, use the **Get-SCACUserRole** cmdlet.

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

<CommonParameters>

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

Examples

Example 1: Add members and a user role scope to a new user role

The first command creates a user role named UserRole01 and stores the user role object in the \$UserRole variable.

The second command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$AzureSubscription variable

The third command creates a user role scope with the Windows Azure subscription stored in \$AzureSubscription and stores the user role scope object in the \$UserRoleScope variable.

The last command adds the specified users and the user role scope stored in \$UserRoleScope to the user role stored in \$UserRole.

```
PS C:\> $UserRole = New-SCACUserRole -Name "UserRole01" -Description "First user role"
PS C:\> $AzureSubscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> $UserRoleScope = New-SCACUserRoleScope -AzureCloud $AzureSubscription
PS C:\> Set-SCACUserRole -UserRole $UserRole -AddMembers
"Contoso\ReneeLo", "Contoso\NevenSokec" -AddScope $UserRoleScope
```

Example 2: Remove a member from an existing user role

The first command gets the user role object named UserRole01 and stores the object in the \$UserRole variable.

The second command removes the specified member from the user role stored in \$UserRole.

```
PS C:\> $UserRole = Get-SCACUserRole -Name "UserRole01" -Managed
PS C:\> Set-SCACUserRole -UserRole $UserRole -RemoveMembers "Contoso\ReneeLo"
```

Related topics

[Get-SCACUserRole](#)

[New-SCACUserRole](#)

[New-SCACUserRoleScope](#)

Suspend-SCACServiceDeployment

Suspend-SCACServiceDeployment

Suspends the specified service deployment.

Syntax

Parameter Set: Default

```
Suspend-SCACServiceDeployment [-ServiceDeployment] <ACServiceDeployment> [[-RunAsynchronously]] [ <CommonParameters>]
```

Detailed Description

The **Suspend-SCACServiceDeployment** cmdlet suspends the specified service deployment. To resume a service deployment, use the **Resume-SCACServiceDeployment** cmdlet.

Parameters

-RunAsynchronously

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

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

-ServiceDeployment<ACServiceDeployment>

Specifies a service deployment object. To get a service deployment object, use the **Get-SCACAzureServiceDeployment** 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.SystemCenter.CloudManager.PowerShell.ACServiceDeployment, System.Management.Automation.SwitchParameter**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACServiceDeployment**

Examples

Example 1: Suspend a service deployment

The first command gets the Windows Azure subscription object named AzureSubscription01 and stores the object in the \$Subscription variable.

The second command gets all service deployment objects for the subscription stored in \$Subscription and stores the service deployment objects in the \$SvcDeployment variable.

The last command suspends the first service deployment stored in \$SvcDeployment, and runs the command asynchronously so that control returns to the command shell immediately, before the command stops running.

```
PS C:\> $Subscription = Get-SCACAzureSubscription -Name "AzureSubscription01"
PS C:\> $SvcDeployment = Get-SCACAzureServiceDeployment -Subscription $Subscription
PS C:\> Suspend-SCACServiceDeployment -ServiceDeployment $SvcDeployment[0] -
RunAsynchronously
```

Related topics

[Get-SCACAzureServiceDeployment](#)

[Resume-SCACServiceDeployment](#)

[Get-SCACAzureSubscription](#)

