



Cmdlet Reference for App Controller in System Center 2012 SP1

Microsoft Corporation

Published: January 15, 2013

Applies To

System Center 2012 Service Pack 1 (SP1) - App Controller

Feedback

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

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

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

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

This document is confidential and proprietary to Microsoft. It is disclosed and can be used only pursuant to a non-disclosure agreement.

© 2011 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
January 15, 2013	Original Release of this guide.

Contents

Add-SCACAzureDisk	5
Add-SCACAzureImage	10
Add-SCACAzureSubscription	15
Add-SCACCloudSystem	19
Add-SCACShare	24
Add-SCACUserRole	26
Add-SCACUserRoleMember	28
Add-SCACUserRoleScope	30
Export-SCACAesKey	33
Get-SCACAdminSetting	35
Get-SCACAzureHostedService	38
Get-SCACAzureRoleInstance	40
Get-SCACAzureServiceDeployment	41
Get-SCACCloud	44
Get-SCACCloudSystem	47
Get-SCACJob	49
Get-SCACServer	51
Get-SCACServiceDeployment	54
Get-SCACTemporaryStorage	56
Get-SCACUserRole	58
Remove-SCACAzureSubscription	61
Remove-SCACCloudSystem	63
Remove-SCACShare	65

Remove-SCACUserRole	67
Remove-SCACUserRoleMember	69
Remove-SCACUserRoleScope	71
Resume-SCACServiceDeployment	73
Set-SCACAdminSetting	76
Set-SCACCloudSystem	80
Set-SCACTemporaryStorage	83
Suspend-SCACServiceDeployment	85

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 <StandaloneVirtualHardDisk> -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 cloud object, use the **Get-SCACCloud** 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<StandaloneVirtualHardDisk>

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 -----

Description

The first command gets all cloud objects and stores the objects in the \$Clouds variable.

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

```
PS C:\> $Clouds = Get-SCACCloud
```

```
PS C:\> Add-SCACAzureDisk -Name "VHD01" -DisplayName "VHD01.vhd" -Cloud $Clouds[0] -
StorageBlob "https://container01.blob.core.windows.net/vhds/VHD01.vhd" -SourcePath
"\\FileShare\VMs\Vhd01.vhd" -OperatingSystem "Windows" -Force
```

Related topics

[Add-SCACAzureImage](#)

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 {None | Windows | Linux} -SourcePath <String> -StorageBlob <Uri> [-Force] [  
<CommonParameters>]
```

Parameter Set: SourceVHD

```
Add-SCACAzureImage -Cloud <ACAzureSubscriptionCloud> -DisplayName <String> -Name <String> -  
OperatingSystem {None | Windows | Linux} -SourceVHD <StandaloneVirtualHardDisk> -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 cloud object, use the **Get-SCACCloud** 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: 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 image.

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

-SourceVHD<StandaloneVirtualHardDisk>

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 -----

Description

The first command gets all cloud objects and adds the objects to the \$Clouds parameter.

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 and names it Image01.

```
PS C:\> $Clouds = Get-SCACCloud
PS C:\> $VHD = Get-SCVirtualHardDisk -Name "VHD01.vhd"
PS C:\> Add-SCAAzureImage -Name "Image01" -DisplayName "Image01" -Cloud $Clouds[0] -
StorageBlob "http://container01.blob.core.windows.net/vhds/VHD01.vhd" -SourceVHD $VHD -
OperatingSystem "Windows"
```

Related topics

[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> ] [ <CommonParameters>]
```

Detailed Description

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

Parameters

-Description<String>

Specifies a description for the Windows Azure subscription.

Aliases	none
Required?	false
Position?	named
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

<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 -----

Description

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-SCAAzureSubscription](#)

Add-SCACCloudSystem

Add-SCACCloudSystem

Adds a new 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> ] [ <CommonParameters>]
```

Parameter Set: VMM

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

Detailed Description

The **Add-SCACCloudSystem** function 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>

Specifies 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

-Port<Int32>

Specifies the port to be used 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 -----

Description

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 -----

Description

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] <String> [ <CommonParameters>]
```

Detailed Description

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

Parameters

-Path<String>

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 -----

Description

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

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

Related topics

[Remove-SCACShare](#)

Add-SCACUserRole

Add-SCACUserRole

Creates an App Controller user role.

Syntax

Parameter Set: Default

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

Detailed Description

The **Add-SCACUserRole** function creates an App Controller user role.

Parameters

-Description<String>

Specifies a description for the user role.

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

-IsReadOnly

Indicates that the user role is read-only.

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

Required?	false
Position?	named
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 -----

Description

This example creates a read-only user role named UserRole01.

```
PS C:\>Add-SCACUserRole -Name "UserRole01" -Description "Test User Role" -IsReadOnly
```

Related topics

[Get-SCACUserRole](#)

[Remove-SCACUserRole](#)

Add-SCACUserRoleMember

Add-SCACUserRoleMember

Adds a member to a user role.

Syntax

Parameter Set: Default

```
Add-SCACUserRoleMember [-UserRole] <ACUserRole> [-Member] <String> [ <CommonParameters>]
```

Detailed Description

The **Add-SCACUserRoleMember** function adds a member to a specified user role.

Parameters

-Member<String>

Specifies the name of a member to add to a user role.

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

-UserRole<ACUserRole>

Specifies the user role object to which to add members. To retrieve 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 -----

Description

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 member named Contoso\ReneeLo from the user role stored in \$UserRole.

```
PS C:\> $UserRole = Get-SCACUserRole -Managed | where { $_.Name -eq "UserRole01" }
```

```
PS C:\> Add-SCACUserRoleMember -UserRole $UserRole -Member "Contoso\ReneeLo"
```

Related topics

[Remove-SCACUserRoleMember](#)

Add-SCACUserRoleScope

Add-SCACUserRoleScope

Adds Service Provider Foundation connection cloud objects to the scope of objects that a user role can manage.

Syntax

Parameter Set: AzureScope

```
Add-SCACUserRoleScope [-UserRole] <ACUserRole> [-AzureSubscriptionScope]  
<ACAzureSubscriptionCloud> [ <CommonParameters>]
```

Parameter Set: SPFScope

```
Add-SCACUserRoleScope [-UserRole] <ACUserRole> [-ServiceProviderConnectionScope]  
<ACCloudSystem> [-VmmUserRoleName] <String> [ <CommonParameters>]
```

Detailed Description

The **Add-SCACUserRoleScope** function adds Service Provider Foundation connection cloud objects to the scope of objects that a user role can manage. This cmdlet only works with Virtual Machine Manager (VMM) clouds.

Parameters

-AzureSubscriptionScope<ACAzureSubscriptionCloud>

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

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

-ServiceProviderConnectionScope<ACCloudSystem>

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

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

-UserRole<ACUserRole>

Specifies a user role object. 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

-VmmUserRoleName<String>

Specifies the name of a Virtual Machine Manager (VMM) user role.

Aliases	none
Required?	true
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](#).

Examples

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

Description

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

The second command gets the cloud system object named SvcProvider01.Contoso.com and stores the object in the \$CloudSystem variable.

The last command adds the cloud system stored in \$CloudSystem to the scope of UserRole01.

```
PS C:\> $UserRole = Get-SCACUserRole -Managed | where {$_.Name -eq "UserRole01"}
PS C:\> $CloudSystem = Get-SCACCloudSystem -Name "SvcProvider01.Contoso.com"
PS C:\> Add-SCACUserRoleScope -UserRole $UserRole -ServiceProviderConnectionScope
$CloudSystem -VMMUserRoleName "TenantSSUserRole01"
```

Related topics

[Remove-SCACUserRoleScope](#)

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.

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 -----

Description

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

The second 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:\>$Password = ConvertTo-SecureString "PassWord!" -AsPlainText -Force
```

```
PS C:\> Export-SCAESAESKey -Path "C:\Keys\Key.txt" -Password $Password
```

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?	1
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?	1
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?	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**

Outputs

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

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

Examples

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

Description

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

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

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

Description

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

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

Related topics

[Set-SCACAdminSetting](#)

Get-SCACAzureHostedService

Get-SCACAzureHostedService

Gets the hosted services on a specified Windows Azure cloud.

Syntax

Parameter Set: Default

```
Get-SCACAzureHostedService [-Cloud] <CAzureSubscriptionCloud> [ <CommonParameters>]
```

Detailed Description

The **Get-SCACAzureHostedService** cmdlet gets the hosted services on a specified Windows Azure cloud.

Parameters

-Cloud<CAzureSubscriptionCloud>

Specifies a Windows Azure subscription cloud object.

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 -----

Description

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

The second command gets the cloud object for the cloud system stored in \$CloudSystem and stores the cloud object in the \$Cloud variable.

The last command gets all Windows Azure hosted services for the cloud stored in \$Cloud and displays information about the Windows Azure hosted services to the user.

```
PS C:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01.Contoso.com"
```

```
PS C:\> $Cloud = Get-SCACCloud -CloudSystem $CloudSystem
```

```
PS C:\> Get-SCAAzureHostedService -Cloud $Cloud
```

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 [-Cloud] <ACCloud> [ <CommonParameters>]
```

Parameter Set: HostedService

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

Detailed Description

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

Parameters

-Cloud<ACCloud>

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

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

-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

<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 -----

Description

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

The second command gets the cloud object for the cloud system stored in \$CloudSystem and stores the object in the \$Cloud variable.

The third command gets all Windows Azure hosted services for the cloud stored in \$Cloud and stores the hosted service objects in the \$HostedService array.

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:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01.Contoso.com"
```

```
PS C:\> $Cloud = Get-SCACCloud -CloudSystem $CloudSystem
```

```
PS C:\> $HostedService = @(Get-SCACAzureHostedService -Cloud $Cloud)
```

```
PS C:\> Get-SCACAzureServiceDeployment -HostedService $HostedService[0]
```

Related topics

[Get-SCACCloudSystem](#)

[Get-SCACCloud](#)

[Get-SCACAzureHostedService](#)

Get-SCACCloud

Get-SCACCloud

Gets one or more App Controller cloud objects.

Syntax

Parameter Set: Default

```
Get-SCACCloud [[-CloudSystem] <ACCloudSystem> ] [[-Id] <Guid> ] [ <CommonParameters>]
```

Detailed Description

The **Get-SCACCloud** cmdlet gets one or more App Controller cloud objects.

Parameters

-CloudSystem<ACCloudSystem>

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

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

-Id<Guid>

Specifies a GUID that represents the subscription ID for a cloud.

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

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

<CommonParameters>

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

Inputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACCloudSystem, System.Guid**

Outputs

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

- **Microsoft.SystemCenter.CloudManager.PowerShell.ACCloud**

Examples

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

Description

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

The second command gets the clouds for the cloud system stored in \$CloudSystem (in this case, CloudSystem01).

```
PS C:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01.Contoso.com"
```

```
PS C:\> Get-SCACCloud -CloudSystem $CloudSystem
```

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

Description

This command gets the cloud object with the subscription id of bb4915e4-e4bf-499d-8126-b6e80c63daff.

PS C:\> Get-SCACCloud -Id "bb4915e4-e4bf-499d-8126-b6e80c63daff"

Related topics

[Get-SCACCloudSystem](#)

Get-SCACCloudSystem

Get-SCACCloudSystem

Gets one or more cloud systems on an App Controller server.

Syntax

Parameter Set: Default

```
Get-SCACCloudSystem [[-Name] <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

<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 -----

Description

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

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

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

Description

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

```
PS C:\> Get-SCACCloudSystem -Name "Management.Core.Windows.Net"
```

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 -----

Description

This command gets all App Controller jobs.

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

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

Description

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 -----

Description

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 -----

Description

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 -----

Description

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-SCACServiceDeployment

Get-SCACServiceDeployment

Gets the service deployment for a cloud.

Syntax

Parameter Set: Cloud

```
Get-SCACServiceDeployment [-Cloud] <ACCloud> [ <CommonParameters>]
```

Detailed Description

The **Get-SCACServiceDeployment** cmdlet gets the service deployment for a cloud.

Parameters

-Cloud<ACCloud>

Specifies an App Controller cloud object. To retrieve an App Controller cloud object, use the **Get-SCACCloud** 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.ACCloud**

Outputs

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

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

Examples

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

Description

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

The second command gets the cloud object for the cloud system stored in \$CloudSystem and stores the cloud object in the \$Cloud variable.

The last command gets all service deployment objects for the cloud stored in \$Cloud and displays information about each service deployment to the user.

```
PS C:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01.Contoso.com"
```

```
PS C:\> $Cloud = Get-SCACCloud -CloudSystem $CloudSystem
```

```
PS C:\> Get-SCACServiceDeployment -Cloud $Cloud
```

Related topics

[Resume-SCACServiceDeployment](#)

[Suspend-SCACServiceDeployment](#)

Get-SCACTemporaryStorage

Get-SCACTemporaryStorage

Gets the 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 -----

Description

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

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

Related topics

[Set-SCATemporaryStorage](#)

Get-SCACUserRole

Get-SCACUserRole

Gets user roles on the App Controller server.

Syntax

Parameter Set: Managed

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

Parameter Set: MyRoles

```
Get-SCACUserRole -MyRoles [ <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?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-MyRoles

Indicates that the user roles of which the logged-in user is a member are returned.

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.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 -----

Description

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 -----

Description

This command gets all App Controller user roles that the current user is a member of, and displays information about each user role to the user.

PS C:\> Get-SCACUserRole -MyRoles

Related topics

[Add-SCACUserRole](#)

[Remove-SCACUserRole](#)

Remove-SCACAzureSubscription

Remove-SCACAzureSubscription

Removes a Windows Azure subscription from App Controller.

Syntax

Parameter Set: Default

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

Detailed Description

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

Parameters

-Subscription<CAzureSubscriptionCloud>

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

Description

The first command gets the cloud object named AzureSubscription01 and stores the object in the \$Cloud variable.

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

```
PS C:\> $Cloud = Get-SCACCloud | where { $_.Name -eq "AzureSubscription01" }
```

```
PS C:\> Remove-SCACAzureSubscription -Subscription $Cloud
```

Related topics

[Add-SCACAzureSubscription](#)

Remove-SCACCloudSystem

Remove-SCACCloudSystem

Removes a connection to a cloud system.

Syntax

Parameter Set: Default

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

<CommonParameters>

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

Examples

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

Description

The first command gets the cloud system object named Management.Core.Windows.Net 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 "Management.Core.Windows.Net"
```

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

Detailed Description

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

Parameters

-Path<String>

Specifies the path to the share to be removed.

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 -----

Description

This command removes the path named \\LibraryServer\LibraryShare.

```
PS C:\> Remove-SCACShare -Path "\\LibraryServer\LibraryShare"
```

Related topics

[Add-SCACShare](#)

Remove-SCACUserRole

Remove-SCACUserRole

Removes the specified user role.

Syntax

Parameter Set: Default

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

Detailed Description

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

Parameters

-UserRole<ACUserRole>

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

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 -----

Description

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 -Managed | where { $_.Name -eq "UserRole01" }
```

```
PS C:\> Remove-SCACUserRole -UserRole $UserRole
```

Related topics

[Add-SCACUserRole](#)

[Get-SCACUserRole](#)

Remove-SCACUserRoleMember

Remove-SCACUserRoleMember

Removes a member from a user role.

Syntax

Parameter Set: Default

```
Remove-SCACUserRoleMember [-UserRole] <ACUserRole> [-Member] <String> [ <CommonParameters>]
```

Detailed Description

The **Remove-SCACUserRoleMember** function removes the specified member from a user role.

Parameters

-Member<String>

Specifies the name of the member to remove from a user role.

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

-UserRole<ACUserRole>

Specifies a user role object. To retrieve a user role object, use the **Get-SCUserRole** 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 -----

Description

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

The second command removes the member named Contoso\ReneeLo from the user role stored in \$UserRole.

```
PS C:\> $UserRole = Get-SCACUserRole -Managed | where { $_.Name -eq "UserRole01" }
PS C:\> Remove-SCACUserRoleMember -UserRole $UserRole -Member "Contoso\ReneeLo"
```

Related topics

[Add-SCACUserRoleMember](#)

[Get-SCACUserRole](#)

Remove-SCACUserRoleScope

Remove-SCACUserRoleScope

Removes a Windows Azure subscription cloud object from the scope of objects that a user role can manage.

Syntax

Parameter Set: Default

```
Remove-SCACUserRoleScope [-UserRole] <ACUserRole> [-Scope] <CAzureSubscriptionCloud> [  
<CommonParameters>]
```

Detailed Description

The **Remove-SCACUserRoleScope** function removes Windows Azure subscription cloud objects from the scope of objects that a user role can manage.

Parameters

-Scope<CAzureSubscriptionCloud>

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

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

-UserRole<ACUserRole>

Specifies a user role object. 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 -----

Description

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

The second command gets the cloud object named AzureSubscription01 and stores the object in the \$Scope variable.

The last command removes the cloud stored in \$Scope to the user role stored in \$UserRole.

```
PS C:\> $UserRole = Get-SCACUserRole -Managed | where { $_.Name -eq "UserRole01" }
```

```
PS C:\> $Scope = Get-SCACCloud | where { $_.Name -eq "AzureSubscription01" }
```

```
PS C:\> Remove-SCACUserRoleScope -UserRole $UserRole -Scope $Scope
```

Related topics

[Add-SCACUserRoleScope](#)

[Get-SCACUserRole](#)

[Get-SCACCloud](#)

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-SCACServiceDeployment 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 -----

Description

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

The second command gets the cloud object for the cloud system stored in \$CloudSystem and stores the cloud object in the \$Cloud variable.

The third command gets all service deployment objects for the cloud stored in \$Cloud that have a status of Suspended and stores the objects in the \$SvcDeployment array.

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:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01.Contoso.com"
```

```
PS C:\> $Cloud = Get-SCACCloud -CloudSystem $CloudSystem
```

```
PS C:\> $SvcDeployment = @(Get-SCACServiceDeployment -Cloud $Cloud | where { $_.Status -eq  
"Suspended" })  
PS C:\> Resume-SCACServiceDeployment -ServiceDeployment $SvcDeployment[0] -RunAsynchronously
```

Related topics

[Get-SCACServiceDeployment](#)

[Suspend-SCACServiceDeployment](#)

Set-SCACAdminSetting

Set-SCACAdminSetting

Sets an administrator setting to a specified value.

Syntax

Parameter Set: AdminSettingBySettingObject

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

Parameter Set: CEIP

```
Set-SCACAdminSetting [-CEIPEnabled] <Boolean> [ <CommonParameters>]
```

Parameter Set: JobHistoryPeriod

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

Parameter Set: RefreshInterval

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

Detailed Description

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

Parameters

-AdminSetting<ACAdminSetting>

Specifies an admin setting object. To get an admin 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<Boolean>

Specifies, when set to \$True, that participation in the Microsoft Customer Experience Improvement Program (CEIP) is enabled.

Aliases	none
Required?	true
Position?	1
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?	1
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?	1
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 -----

Description

This command sets the JobHistoryPeriodInDays admin setting to 60.

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

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

Description

This command enables the CEIP setting by setting the CEIPEnabled admin setting to \$True.

```
PS C:\> Set-SCACAdminSetting -CEIPEnabled $True
```

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

Description

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

The second command sets the value for the administrator setting stored in \$Setting (RefreshIntervalInSeconds) 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> [-CertificateFilePath <String> ] [-Description <String> ] [-Name <String> ] [ <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?	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>

Specifies a description for the cloud system.

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

<CommonParameters>

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

Examples

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

Description

The first command gets the cloud system object named CloudSystem01.Contoso.com 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.Contoso.com"
```

```
PS C:\> Set-SCACCloudSystem -CloudSystem $CloudSystem -Name "Test Cloud System" -Description  
"This is a test cloud system."
```

Set-SCACTemporaryStorage

Set-SCACTemporaryStorage

Sets the temporary storage used by the App Controller server.

Syntax

Parameter Set: SpecifyPath

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

Detailed Description

The **Set-SCACTemporaryStorage** cmdlet sets the temporary storage used 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 -----

Description

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

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

Related topics

[Get-SCACTemporaryStorage](#)

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-SCACServiceDeployment** 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 -----

Description

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

The second command gets the cloud object for the cloud system stored in \$CloudSystem and stores the cloud object in the \$Cloud variable.

The third command gets all service deployment objects for the cloud stored in \$Cloud and stores the objects in the \$SvcDeployment array.

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:\> $CloudSystem = Get-SCACCloudSystem -Name "CloudSystem01.Contoso.com"
```

```
PS C:\> $Cloud = Get-SCACCloud -CloudSystem $CloudSystem
```

```
PS C:\> $SvcDeployment = @(Get-SCACServiceDeployment -Cloud $Cloud)
PS C:\> Suspend-SCACServiceDeployment -ServiceDeployment $SvcDeployment[0] -
RunAsynchronously
```

Related topics

[Resume-SCACServiceDeployment](#)