System Center 2012 - App Controller Cmdlet Reference

Microsoft Corporation

Published: April 2012

Feedback

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

Copyright

Contents

Add-SCACAzureSubscription

Add-SCACAzureSubscription

Adds a Windows Azure subscription to App Controller.

Syntax

```
Parameter Set: __AllParameterSets
Add-SCACAzureSubscription [-Name] <String> [-Id] <Guid> [-ManagementCertificatePath]
<String> [-ManagementCertificatePassword] <SecureString> [-ACServer <ACServer> ] [-
Description <String> ] [ <CommonParameters>]
```

Detailed Description

The Add-SCACAzureSubscription cmdlet adds a Windows Azure subscription to System Center 2012 - App Controller.

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-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 parameter.

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”
```

Add-SCACCloudSystem

Add-SCACCloudSystem

Adds a new connection to a VMM management server.

Syntax

```
Parameter Set: __AllParameterSets
Add-SCACCloudSystem [-CloudSystemName] <String> [-VMMServerName] <String> [-Port] <Int32>
[[-Description] <String> ] [ <CommonParameters>]
```

Detailed Description

The Add-SCACCloudSystem function adds a new connection to a System Center 2012 – Virtual Machine Manager (VMM) management server.

Parameters

-CloudSystemName<String>

Specifies the name of the cloud system.

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

-Description<String>

Specifies a description for the connection.

Aliases	none
Required?	false

Position?	4
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

-VMMServerName<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

<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 81.

```
PS C:\> Add-SCACCloudsystem -CloudSystemName “CloudSystem01” -VMMServerName
VMMServer01.Contoso.com -Port 81
```

Add-SCACShare

Add-SCACShare

Adds a library share.

Syntax

```
Parameter Set: __AllParameterSets
Add-SCACShare [-SharePath] <String> [ <CommonParameters>]
```

Detailed Description

The Add-SCACShare function adds a library share.

Parameters

-SharePath<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 -SharePath "\\LibraryServer\LibraryPath"
```

Add-SCACUserRole

Add-SCACUserRole

Creates a user role.

Syntax

```
Parameter Set: __AllParameterSets
Add-SCACUserRole [-UserRoleName] <String> [-Description <String> ] [-IsReadOnly] [
<CommonParameters>]
```

Detailed Description

The Add-SCACUserRole function creates a 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 whether the user role is read-only.

Aliases	none
Required?	false
Position?	named

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

-UserRoleName<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 -UserRoleName UserRole01 -Description "Test User Role" -IsReadOnly
```

Add-SCACUserRoleMember

Add-SCACUserRoleMember

Adds a member to a user role.

Syntax

```
Parameter Set: __AllParameterSets
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”
```

Add-SCACUserRoleScope

Add-SCACUserRoleScope

Adds Windows Azure subscription cloud objects to the scope of objects that a user role can manage.

Syntax

```
Parameter Set: __AllParameterSets
Add-SCACUserRoleScope [-UserRole] <ACUserRole> [-Scope] <ACAzureSubscriptionCloud> [
<CommonParameters>]
```

Detailed Description

The Add-SCACUserRoleScope function adds Windows Azure subscription cloud objects to the scope of objects that a user role can manage.

Parameters

-Scope<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

-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 adds 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:\> Add-SCACUserRoleScope -UserRole $UserRole -Scope $Scope
```

Export-SCACAesKey

Export-SCACAesKey

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

Syntax

```
Parameter Set: __AllParameterSets
Export-SCACAesKey [-FilePath] <String> [-Password] <SecureString> [ <CommonParameters>]
```

Detailed Description

The Export-SCACAESKey cmdlet exports the System Center 2012 - App Controller Advanced Encryption Standard (AES) key from the registry to the specified file. Use the FilePath parameter to specify the destination file.

Parameters

-FilePath<String>

Specifies a file location path.

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

-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

<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-SCACAESKey -FilePath "C:\Keys\Key.txt" -Password $Password
```

Get-SCACAdminSetting

Get-SCACAdminSetting

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

Syntax

```
Parameter Set: __AllParameterSets
Get-SCACAdminSetting [-ACServer <ACServer> ] [ <CommonParameters>]

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 System Center 2012 - App Controller server. If a setting name is specified by using the Name parameter, that setting is retrieved. It no setting is specified, all settings on the server are returned.

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-ACServer cmdlet.

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

-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: __AllParameterSets
Get-SCACAzureHostedService [-Cloud] <ACAzureSubscriptionCloud> [-ACServer <ACServer> ] [
<CommonParameters>]
```

Detailed Description

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

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-Cloud<ACAzureSubscriptionCloud>

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

Get-SCACAzureServiceDeployment

Get-SCACAzureServiceDeployment

Gets a specified Windows Azure service deployment.

Syntax

```
Parameter Set: __AllParameterSets
Get-SCACAzureServiceDeployment [-ACServer <ACServer> ] [ <CommonParameters>]

Parameter Set: CloudParamSet
Get-SCACAzureServiceDeployment [-Cloud] <ACCloud> [ <CommonParameters>]

Parameter Set: HostedServiceParamSet
Get-SCACAzureServiceDeployment [-HostedService] <ACAzureHostedService> [[-DeploymentSlot]
<String> ] [ <CommonParameters>]
```

Detailed Description

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

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-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 an 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"
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: __AllParameterSets
Get-SCACCloud [[-CloudSystem] <ACCloudSystem> ] [[-Id] <Guid> ] [-ACServer <ACServer> ] [
<CommonParameters>]
```

Detailed Description

The Get-SCACCloud cmdlet gets one or more System Center 2012 - App Controller cloud objects.

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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?	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"
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: __AllParameterSets
Get-SCACCloudSystem [[-Name] <String> ] [-ACServer <ACServer> ] [ <CommonParameters>]
```

Detailed Description

The Get-SCACCloudSystem cmdlet gets one or more cloud systems on an System Center 2012 - 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

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-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: ParamSetMultipleJobs
Get-SCACJob [[-All]] [ <CommonParameters>]

Parameter Set: __AllParameterSets
Get-SCACJob [-ACServer <ACServer> ] [ <CommonParameters>]

Parameter Set: ParamSetSingleJob
Get-SCACJob [[-Id] <Guid> ] [ <CommonParameters>]
```

Detailed Description

The Get-SCACJob cmlet gets System Center 2012 - 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

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

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

Detailed Description

The Get-SCACServer cmdlet gets and establishes a connection with the specified
System Center 2012 - 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 NewUserRole01 using the credentials stored in $Credentials.

```
PS C:\> $Credentials = Get-Credential
PS C:\> Get-SCACServer -ServerName "https://AppControllerSvr01.Contoso.com" -Credential
$Credentials -UserRole "NewUserRole01"
```

Get-SCACServiceDeployment

Get-SCACServiceDeployment

Gets the service deployment for an App Controller server.

Syntax

```
Parameter Set: CloudParamSet
Get-SCACServiceDeployment [-Cloud] <ACCloud> [ <CommonParameters>]
Parameter Set: __AllParameterSets
Get-SCACServiceDeployment [-ACServer <ACServer> ] [ <CommonParameters>]
```

Detailed Description

The Get-SCACServiceDeployment cmdlet gets the service deployment for a System Center 2012 - App Controller server.

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-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"
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 specified App Controller server.

Syntax

```
Parameter Set: __AllParameterSets
Get-SCACTemporaryStorage [-ACServer <ACServer> ] [ <CommonParameters>]
```

Detailed Description

The Get-SCACTemporaryStorage cmdlet gets the temporary storage used by the specified System Center 2012 - App Controller server. To set temporary storage, use the Set-SCACTemporaryStorage cmdlet.

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

<CommonParameters>

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

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 gets the temporary storage for the App Controller server stored in $ACServer and displays information about the storage to the user.

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

Related topics

Set-SCACTemporaryStorage

Get-SCACUserRole

Get-SCACUserRole

Gets the user roles on an App Controller server.

Syntax

```
Parameter Set: __AllParameterSets
Get-SCACUserRole [-ACServer <ACServer> ] [ <CommonParameters>]

Parameter Set: Managed
Get-SCACUserRole -Managed [ <CommonParameters>]

Parameter Set: MyRoles
Get-SCACUserRole -MyRoles [ <CommonParameters>]
```

Detailed Description

The Get-SCACUserRole cmdlet gets the user roles on a System Center 2012 - App Controller server.

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-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
```

Remove-SCACAzureSubscription

Remove-SCACAzureSubscription

Removes a Windows Azure subscription from App Controller.

Syntax

```
Parameter Set: __AllParameterSets
Remove-SCACAzureSubscription [-AzureSubscription] <ACAzureSubscriptionCloud> [
<CommonParameters>]
```

Detailed Description

The Remove-SCACAzureSubscription function removes a Windows Azure subscription from System Center 2012 - App Controller.

Parameters

-AzureSubscription<ACAzureSubscriptionCloud>

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 –AzureSubscription $Cloud
```

Remove-SCACCloudSystem

Remove-SCACCloudSystem

Removes a connection to a VMM management server.

Syntax

```
Parameter Set: __AllParameterSets
Remove-SCACCloudSystem [-CloudSystem] <ACCloudSystem> [ <CommonParameters>]
```

Detailed Description

The Remove-SCACCloudSystem function removes a connection to a System Center 2012 – Virtual Machine Manager (VMM) management server.

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 –ACCloudSystem $CloudSystem
```

Remove-SCACShare

Remove-SCACShare

Removes a library share.

Syntax

```
Parameter Set: __AllParameterSets
Remove-SCACShare [-SharePath] <String> [ <CommonParameters>]
```

Detailed Description

The Remove-SCACShare function removes a specified library share.

Parameters

-SharePath<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 -SharePath "\\LibraryServer\LibraryShare"
```

Remove-SCACUserRole

Remove-SCACUserRole

Removes the specified user role.

Syntax

```
Parameter Set: __AllParameterSets
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?	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 stored in $UserRole.

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

Related topics

Get-SCACUserRole

Remove-SCACUserRoleMember

Remove-SCACUserRoleMember

Removes a member from a user role.

Syntax

```
Parameter Set: __AllParameterSets
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

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: __AllParameterSets
Remove-SCACUserRoleScope [-UserRole] <ACUserRole> [-Scope] <ACAzureSubscriptionCloud> [
<CommonParameters>]
```

Detailed Description

The Renove-SCACUserRoleScope function removes Windows Azure subscription cloud objects from the scope of objects thata user role can manage.

Parameters

-Scope<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

-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

Get-SCACUserRole

Get-SCACCloud

Resume-SCACServiceDeployment

Resume-SCACServiceDeployment

Resumes the specified suspended service deployment.

Syntax

```
Parameter Set: __AllParameterSets
Resume-SCACServiceDeployment [-ServiceDeployment] <ACServiceDeployment> [[-
RunAsynchronously]] [-ACServer <ACServer> ] [ <CommonParameters>]
```

Detailed Description

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

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-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"
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: __AllParameterSets
Set-SCACAdminSetting [-ACServer <ACServer> ] [ <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

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

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

Related topics

Get-SCACAdminSetting

Set-SCACTemporaryStorage

Set-SCACTemporaryStorage

Sets the temporary storage used by the App Controller server.

Syntax

```
Parameter Set: SpecifyPath
Set-SCACTemporaryStorage [-Path] <String> [ <CommonParameters>]
Parameter Set: __AllParameterSets
Set-SCACTemporaryStorage [-ACServer <ACServer> ] [ <CommonParameters>]
```

Detailed Description

The Set-SCACTemporaryStorage cmdlet sets the temporary storage used by the System Center 2012 - App Controller server.

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

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

Detailed Description

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

Parameters

-ACServer<ACServer>

Specifies an App Controller server object. To retrieve an App Controller server object, use the Get-SCACServer cmdlet.

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

-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"
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