

Microsoft System Center 2012 R2

Cmdlet Reference for System Center 2012 R2 Service Provider Foundation

Microsoft Corporation

Published: November 1, 2013

Applies To

System Center 2012 R2 Service Provider Foundation

Feedback

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

Copyright

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

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

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

© 2013 Microsoft Corporation. All rights reserved.

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

Revision History

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

Contents

Get-SCSPFConnectionString	5
Get-SCSPFEventRegistration	6
Get-SCSPFExtensibleEventHandler	9
Get-SCSPFServer	11
Get-SCSPFSetting	15
Get-SCSPFStamp	18
Get-SCSPFTenant	23
Get-SCSPFTenantUserRole	28
Get-SCSPFTrustedIssuer	31
Get-SCSPFVMRoleGalleryItem	34
Get-SCSPFVMRoleGalleryItemIcon	37
Get-SCSPFVMRoleGalleryItemPackage	40
Import-SCSpfVMRoleGalleryItem	43
New-SCSPFETWEvent	46
New-SCSPFServer	49
New-SCSPFSetting	52
New-SCSPFStamp	56
New-SCSPFTenant	60
New-SCSPFTenantUserRole	65
New-SCSPFTrustedIssuer	68
Remove-SCSPFServer	71
Remove-SCSPFSetting	74
Remove-SCSPFStamp	76
Remove-SCSPFTenant	79
Remove-SCSPFTenantUserRole	82

Remove-SCSPFTrustedIssuer	84
Remove-SCSPFVMRoleGalleryItem	86
Set-SCSPFConnectionString	89
Set-SCSPFEventRegistration	92
Set-SCSPFExtensibleEventHandler	96
Set-SCSPFServer	100
Set-SCSPFStamp	103
Set-SCSPFTenant	107
Set-SCSPFVMRoleGalleryItem	112
Start-SCOrchestratorRunbook	115

Get-SCSPFConnectionString

Get-SCSPFConnectionString

Gets the connection string to the Service Provider Foundation database. Requires running Windows PowerShell as Administrator.

Syntax

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

Detailed Description

The Get-SCSPFConnectionString cmdlet gets the database connection string to the Service Provider Foundation database. The database can be on a different server than the server that is running Service Provider Foundation. This cmdlet requires running Windows PowerShell as Administrator.

Parameters

<CommonParameters>

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

Examples

1: Get the connection string.

This command gets the connection string to the Service Provider Foundation database.

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

Related topics

[Set-SCSPFConnectionString](#)

Get-SCSPFEventRegistration

Get-SCSPFEventRegistration

Gets the event used to automate runbooks.

Syntax

```
Parameter Set: FromEventRegistrationResourceActionEnabledParameterSetName
Get-SCSPFEventRegistration [-Action {Action_Pause | Action_Publish | Action_Repair |
Action_Restart | Action_Resume | Action_Scale | Action_Shutdown | Action_Start | Action_Stop
| Action_Unpublish | Create | Delete | Update | UpdateStatus} ] [-Resource <String> ] [
<CommonParameters>]
```

Detailed Description

The Get-SCSPFEventRegistration cmdlet gets an event that was registered to automate a runbook as performed by Service Management Automation.

Parameters

-Action<EventRegistrationActionNames>

Specifies the name of the event.
The acceptable values for this parameter are:

Action_Pause	
Action_Publish	
Action_Repair	
Action_Restart	
Action_Resume	
Action_Scale	
Action_Shutdown	
Action_Start	

Action_Stop	
Action_Unpublish	
Create	
Delete	
Update	
UpdateStatus	

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

-Resource<String>

Specifies the path to the resource.

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

1. Get an event registration by its name.

This command gets an event with the Action parameter and stores it in the \$event_backup variable.

```
PS C:\> $event_backup = Get-SCSPFEventRegistration -Action "Backup"
```

Get-SCSPFExtensibleEventHandler

Get-SCSPFExtensibleEventHandler

Gets the configuration of a runbook in System Center 2012 - Orchestrator that has been set to be invoked when a specified event is raised.

Syntax

Parameter Set: Default

```
Get-SCSPFExtensibleEventHandler -EventName <String> [ <CommonParameters>]
```

Detailed Description

The Get-SCSPFExtensibleEventHandler cmdlet gets the configuration for the Orchestrator runbook that is invoked whenever the **VirtualMachineCreated** or **ServiceCreated** extensible events are raised.

Parameters

-EventName<String>

Specifies either **VirtualMachineCreated** or **ServiceCreated** for the event that invokes an Orchestrator runbook. These are the only permitted values for this parameter.

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: Get current configuration

This command gets the runbook that is configured to be invoked when the **VirtualMachineCreated** extensible event is raised, and lists the event name, runbook URI, and runbook path.

```
PS C:\> Get-SCSPFExtensibleEventHandler -EventName "VirtualMachineCreated"
```

Related topics

[Set-SCSPFExtensibleEventHandler](#)

Get-SCSPFServer

Get-SCSPFServer

Gets one or more server objects.

Syntax

Parameter Set: Empty

```
Get-SCSPFServer [-ServerType <String> ] [ <CommonParameters>]
```

Parameter Set: FromServerIdParameterSetName

```
Get-SCSPFServer -ID <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromServerNameParameterSetName

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

Parameter Set: FromServerStampParameterSetName

```
Get-SCSPFServer -Stamp <Stamp> [-ServerType <String> ] [ <CommonParameters>]
```

Detailed Description

The Get-SCSPFServer cmdlet gets one or more server objects in Service Provider Foundation. In the current version of Service Provider Foundation, these are VMM server objects. To create a new server, use the New-SCSPFServer cmdlet.

Parameters

-ID<Guid[]>

Specifies one or more identifiers (as a GUID) for a specific object.

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

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

-Name<String[]>

Specifies the name of one or more VMM servers.

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

-ServerType<String>

Specifies the type of server. You can specify only 0 for Virtual Machine Manager (VMM) as the server type, other values are reserved for future use.

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

-Stamp<Stamp>

Specifies a stamp for which this cmdlet will return associated servers. To obtain a stamp, use the Get-SCSPFStamp cmdlet.

Aliases	none
Required?	true

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

<CommonParameters>

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

Examples

1: Get all servers.

This command lists all the defined servers.

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

2: Get a server by its ID.

This command gets a server by its ID.

```
PS C:\> $srvr = Get-SCSPFServer -ID f50e36aa-216f-4ade-a2c0-95cf17b93ee4
```

3: Get a server by its name.

This command gets a server by its name.

```
PS C:\> $srvr = Get-SCSPFServer -Name "VMAIaska"
```

4: Get the server that is associated with a stamp.

This first command gets a stamp.

The second command gets the server associated with the stamp.

```
PS C:\> $stmp = Get-SCSPFStamp -Name "Silver"
```

```
PS C:\> $srvr = Get-SCSPFServer -Stamp $stmp
```

Related topics

[New-SCSPFServer](#)

[Set-SCSPFServer](#)

[Remove-SCSPFServer](#)

Get-SCSPFSetting

Get-SCSPFSetting

Gets a setting for a variety of purposes.

Syntax

Parameter Set: Empty
Get-SCSPFSetting [-SettingType <String>] [<CommonParameters>]

Parameter Set: FromSettingIdParameterSetName
Get-SCSPFSetting -ID <Guid[]> [<CommonParameters>]

Parameter Set: FromSettingNameParameterSetName
Get-SCSPFSetting -Name <String[]> [<CommonParameters>]

Parameter Set: FromSettingServerParameterSetName
Get-SCSPFSetting -Server <Server> -ServerName <String> [-SettingType <String>] [<CommonParameters>]

Detailed Description

The Get-SCSPFSetting cmdlet gets the setting for usage metering, or for a portal endpoint connection.

Parameters

-ID<Guid[]>

Specifies the ID for the setting.

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

-Name<String[]>

Specifies a friendly name for the setting.

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

-Server<Server>

Specifies a server object associated with the setting.

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

-ServerName<String>

Specifies the name of the sever associated with the setting.

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

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

-SettingType<String>

Specifies either DatabaseConnectionString or EndPointConnectionString.

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

1: Get a setting

This command gets an existing setting by its ID.

```
PS C:\> $dbOMDW = Get-SCSPFSetting -ID f3b39608-ac58-40b7-853c-241e343b256a
```

Get-SCSPFStamp

Get-SCSPFStamp

Gets one or more stamp objects.

Syntax

Parameter Set: FromStampDataCenterParameterSetName

Get-SCSPFStamp -DataCenter <DataCenter> [<CommonParameters>]

Parameter Set: FromStampIdAndServerParameterSetName

Get-SCSPFStamp -ID <Guid[]> -Server <Server> [<CommonParameters>]

Parameter Set: FromStampIdAndTenantParameterSetName

Get-SCSPFStamp -ID <Guid[]> -Tenant <Tenant> [<CommonParameters>]

Parameter Set: FromStampIdParameterSetName

Get-SCSPFStamp -ID <Guid[]> [<CommonParameters>]

Parameter Set: FromStampNameAndServerParameterSetName

Get-SCSPFStamp -Name <String[]> -Server <Server> [<CommonParameters>]

Parameter Set: FromStampNameAndTenantParameterSetName

Get-SCSPFStamp -Name <String[]> -Tenant <Tenant> [<CommonParameters>]

Parameter Set: FromStampNameParameterSetName

Get-SCSPFStamp -Name <String[]> [<CommonParameters>]

Parameter Set: FromStampOfferParameterSetName

Get-SCSPFStamp -Offer <Offer> [<CommonParameters>]

Parameter Set: FromStampServerParameterSetName

Get-SCSPFStamp -Server <Server> [<CommonParameters>]

Parameter Set: FromStampTenantParameterSetName

Get-SCSPFStamp -Tenant <Tenant> [<CommonParameters>]

Detailed Description

The Get-SCSPFStamp cmdlet gets one or more stamp objects in Service Provider Foundation. To create a stamp, use the New-SCSPFStamp cmdlet.

Parameters

-DataCenter<DataCenter>

Specifies the name of a DataCenter object. To obtain a DataCenter object, use the New-Object cmdlet with Microsoft.SystemCenter.Foundation.SPFDData.Types.DataCenter specified for its TypeName parameter. Then set its Name property to the name of the your data center, as shown in the following example.

```
PS C:\> $DS = New-Object Microsoft.SystemCenter.Foundation.SPFDData.Types.DataCenter
PS C:\> $DS.Name = "DataCenter_NorthAmerica"
```

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

-ID<Guid[]>

Specifies one or more identifiers (as a GUID) for a specific object.

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

-Name<String[]>

Specifies the name of one or more stamps.

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

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

-Offer<Offer>

Specifies an offer for which this cmdlet will return associated stamps. To obtain an offer, use the Get-SCSPFOffer cmdlet.

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

-Server<Server>

Specifies a server for which this cmdlet will return associated stamps. To obtain a server, use the Get-SCSPFServer cmdlet.

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

-Tenant<Tenant>

Specifies an tenant for which this cmdlet will return associated stamps. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

1: Get all stamps.

This command lists all stamps.

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

2: Get a stamp by its ID.

This command gets a stamp by its ID.

```
PS C:\> $stmp = Get-SCSPFStamp -ID db656655-68ff-4a21-bd1d-0a06bdbbc762f
```

3: Get a stamp by its name.

This command gets a stamp by its name.

```
PS C:\> $stmp = Get-SCSPFStamp -Name "Ultimate"
```

4: Get the stamps that are associated with an offer.

The first command gets an offer. The second command gets the stamps associated with the offer.

```
PS C:\> $offr = Get-SCSPFOffer -Name "Beginner"
PS C:\> $stmp = Get-SCSPFStamp -Offer $offr
```

5: Get the stamp that is associated with a server.

The first command gets a server.

The second command gets the stamp that is associated with the server.

```
PS C:\> $srvr = Get-SCSPFServer -Name "serverVids7"
PS C:\> $stmps = Get-SCSPFStamp -Server $srvr
```

6: Get the stamps that are associated with a tenant.

The first command gets a tenant.

The second command gets the stamps associated with the tenant.

```
PS C:\> $Contoso = Get-SCSPFTenant -Name "Contoso"
PS C:\> $stmps = Get-SCSPFStamp -Tenant $Contoso
```

Related topics

[Set-SCSPFStamp](#)

[New-SCSPFStamp](#)

[Remove-SCSPFStamp](#)

Get-SCSPFTenant

Get-SCSPFTenant

Gets one or more tenant objects.

Syntax

Parameter Set: FromTenantIdParameterSetName
Get-SCSPFTenant -ID <Guid[]> [<CommonParameters>]

Parameter Set: FromTenantIssuerParameterSetName
Get-SCSPFTenant -Issuer <Issuer> [<CommonParameters>]

Parameter Set: FromTenantNameParameterSetName
Get-SCSPFTenant -Name <String[]> [<CommonParameters>]

Parameter Set: FromTenantOfferParameterSetName
Get-SCSPFTenant -Offer <Offer> [<CommonParameters>]

Parameter Set: FromTenantRoleParameterSetName
Get-SCSPFTenant -Role <Role> [<CommonParameters>]

Parameter Set: FromTenantStampParameterSetName
Get-SCSPFTenant -Stamp <Stamp> [<CommonParameters>]

Detailed Description

The Get-SCSPFTenant cmdlet gets one or more tenant objects in Service Provider Foundation. To create a tenant, use the New-SCSPFTenant command.

Parameters

-ID<Guid[]>

Specifies one or more identifiers (as a GUID) for a specific object.

Aliases	none
Required?	true

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

-Issuer<Issuer>

Specifies a trusted issuer object. To obtain a trusted issuer, use the Get-SCSPFTrustedIssuer cmdlet.

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

-Name<String[]>

Specifies the name of one or more tenants.

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

-Offer<Offer>

Specifies an offer for which this cmdlet will return associated tenants. To obtain an offer, use the Get-SCSPFOffer cmdlet.

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

-Role<Role>

Specifies an tenant user role for which this cmdlet will return associated tenants. To obtain a tenant user role, use the Get-SCSPFTenantUserRole cmdlet.

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

-Stamp<Stamp>

Specifies an stamp for which this cmdlet will return associated tenants. To obtain a stamp, use the Get-SCSPFStamp cmdlet.

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

<CommonParameters>

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

Examples

1: Get all tenants.

This command displays the IDs and name of the existing tenants.

```
PS C:\> Get-SCSPFTenant | Format-Table -Property ID, Name
```

2: Get a tenant by its ID.

This command gets a tenant by its ID.

```
PS C:\> $tnnt = Get-SCSPFTenant -ID e5310370-ab2d-4d51-8336-176999c9cc1e
```

3: Get the tenant that is associated with a trusted issuer.

The first command gets a trusted issuer. The second command gets a tenant associated with the issuer.

```
PS C:\> $ti = Get-SCSPFTrustedIssuer -Name "Litware"
```

```
PS C:\> $tnnt = Get-SCSPFTenant -Issuer $ti
```

4: Get a tenant by its name.

This command gets a tenant by its name.

```
PS C:\> $contoso = Get-SCSPFTenant -Name "Contoso"
```

5: Get the tenants that are associated with an offer.

This command gets a tenant associated to an offer. It first gets a specified offer and then uses the pipeline operator to pass the offer to the Get-SCSPFTenant cmdlet to get the associated tenant.

```
PS C:\> Get-SCSPFOffer -Name "Executive" | Get-SCSPFTenant
```

6: Get the tenant that is associated with a tenant user role.

The first command gets a tenant user role.

The second command gets the tenants associated with the tenant user role.

```
PS C:\> $role_supervisor = Get-SCSPFTenantUserRole -ID e6f9f901-2883-4478-a64e-0
0c58a423516
PS C:\> $tnnts = Get-SCSPFTenant -Role $role_supervisor
```

7: Get the tenants that are associated with a stamp.

The first command gets a stamp.

The second command gets the tenants associated with the stamp.

```
PS C:\> $stamp_silver = Get-SCSPFStamp -Name "Silver"
PS C:\> $tnnts = Get-SCSPFTenant -Stamp $stamp_silver
```

Related topics

[New-SCSPFTenant](#)

[Set-SCSPFTenant](#)

[Remove-SCSPFTenant](#)

Get-SCSPFTenantUserRole

Get-SCSPFTenantUserRole

Gets one or more tenant user role objects.

Syntax

Parameter Set: FromRoleIdParameterSetName

```
Get-SCSPFTenantUserRole [-ID] <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromRoleTenantParameterSetName

```
Get-SCSPFTenantUserRole [-Tenant] <Tenant> [[-Name] <String[]> ] [ <CommonParameters>]
```

Detailed Description

The Get-SCSPFTenantUserRole cmdlet gets one or more tenant role objects in Service Provider Foundation. To create a new tenant user role, use the New-SCSPFTenantUserRole cmdlet.

Parameters

-ID<Guid[]>

Specifies one or more identifiers (as a GUID) for a specific object.

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

-Name<String[]>

Specifies the name of one or more tenant user roles.

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

-Tenant<Tenant>

Specifies a tenant for which this cmdlet will return associated tenant user roles. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

<CommonParameters>

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

Examples

1: Get all tenant user roles.

This command lists all tenant user roles.

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

2: Get a tenant user role by its ID.

This command gets a tenant user role by its ID.

```
PS C:\> $tur = Get-SCSPFTenantUserRole -ID e6f9f901-2883-4478-a64e-00c58a423516
```

3: Get the tenant user role that is associated with a tenant.

The first command gets a tenant.

The second command gets the tenant user roles associated with a tenant.

```
PS C:\> $tnnt = Get-SCSPFTenant -Name "Contoso"
```

```
PS C:\> $tur = Get-SCSPFTenantUserRole -Tenant $tnnt
```

Related topics

[New-SCSPFTenantUserRole](#)

[Remove-SCSPFTenantUserRole](#)

Get-SCSPFTrustedIssuer

Get-SCSPFTrustedIssuer

Gets one or more trusted issuer objects.

Syntax

Parameter Set: FromIssuerIdParameterSetName

```
Get-SCSPFTrustedIssuer -ID <Guid[]> [ <CommonParameters>]
```

Parameter Set: FromIssuerNameParameterSetName

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

Parameter Set: FromIssuerTenantParameterSetName

```
Get-SCSPFTrustedIssuer -Tenant <Tenant> [ <CommonParameters>]
```

Detailed Description

The Get-SCSPFTrustedIssuer cmdlet gets one or more trusted issuer objects in Service Provider Foundation. To create a trusted issuer, use the New-SCSPFTrustedIssuer cmdlet.

Parameters

-ID<Guid[]>

Specifies one or more identifiers (as a GUID) for a specific object.

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

-Name<String[]>

Specifies the name of one or more trusted issuers.

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

-Tenant<Tenant>

Specifies a tenant for which this cmdlet will return associated trusted issuers. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

<CommonParameters>

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

Examples

1: Get all trusted issuers.

This command lists all the trusted issuers.

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

2: Get a trusted issuer by its ID.

This command gets a trusted issuer by its ID.

```
PS C:\> $ti = Get-SCSPFTrustedIssuer -ID c9f59be2-dce5-4221-ad60-2204dc457bd8
```

3: Get a trusted issuer by its name.

This command gets a trusted issuer by its name.

```
PS C:\> $ti = Get-SCSPFTrustedIssuer -Name "Woodgrove"
```

4: Get the trusted issuer that is associated with a tenant.

The first command gets a tenant.

The second command gets the trusted issuer associated with the tenant.

```
PS C:\> $Contoso = Get-SCSPFTenant -Name "Contoso"
```

```
PS C:\> $ti = Get-SCSPFTrustedIssuer -Tenant $Contoso
```

Related topics

[Remove-SCSPFTrustedIssuer](#)

[New-SCSPFTrustedIssuer](#)

Get-SCSPFVMRoleGalleryItem

Get-SCSPFVMRoleGalleryItem

Gets a gallery item.

Aliases

The following abbreviations are aliases for this cmdlet:

- The Get-SCVMRoleGalleryItem cmdlets gets a specific gallery item that has been imported.

Syntax

Parameter Set: FromGalleryItemNameParameterSetName

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

Parameter Set: FromGalleryItemNameVersionPublisherParameterSetName

```
Get-SCSPFVMRoleGalleryItem -Name <String> -Publisher <String> -Version <String> [ <CommonParameters>]
```

Parameter Set: FromGalleryItemPublisherParameterSetName

```
Get-SCSPFVMRoleGalleryItem -Publisher <String> [ <CommonParameters>]
```

Detailed Description

The Get-SCSPFVMRoleGalleryItem cmdlets gets a specified gallery item.

Parameters

-Name<String>

Specifies the name that identifies the item in the gallery.

Aliases	none
Required?	true
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-Publisher<String>

Specifies the publisher of the gallery item.

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

-Version<String>

Specifies the version of the gallery item.

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

1: Get a gallery item

This command gets an item from the gallery by specifying its name and stores it in the `$galItem` variable.

```
PS C:\> $galItem = Get-ScSpfVmRoleGalleryItem -Name 570569955cbfb62b374358b34467020750f65c
```

Get-SCSPFVMRoleGalleryItemIcon

Get-SCSPFVMRoleGalleryItemIcon

Gets the icon associated with a virtual machine role item in the gallery.

Syntax

```
Parameter Set: FromGalleryItemParameterSetName
Get-SCSPFVMRoleGalleryItemIcon [-IconFileName] <String> [-VMRoleGalleryItem]
<VMRoleGalleryItem> [ <CommonParameters>]

Parameter Set: FromGalleryItemNameVersionPublisherParameterSetName
Get-SCSPFVMRoleGalleryItemIcon -Name <String> -Publisher <String> -Version <String> [
<CommonParameters>]
```

Detailed Description

The Get-SCSPFVMRoleGalleryItemIcon gets System.IO.MemoryStream object of the icon associated with the gallery item.

Parameters

-Name<String>

Specifies the name of the gallery item.

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

-Publisher<String>

Specifies the publisher of the gallery item.

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

-Version<String>

Specifies the version of the gallery item.

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

-VMRoleGalleryItem<VMRoleGalleryItem>

Specifies the gallery item associated with the icon.

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

-IconFileName<String>

Specifies the file name of the icon.

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

<CommonParameters>

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

Examples

1: Get the icon for a gallery item

The first command gets a gallery item and stores it in the `$galItem` variable. The next command gets the icon object by specifying the required parameters with the variable. The *IconFileName* parameter is explicitly specified in case the variable has a null value for the icon file name.

```
PS C:\> $galItem = Get-SCSPFVMRoleGalleryItem -Name 570569955cbfb62b374358b34467020750f65c
PS C:\> $galItemIcon = Get-SCSPFVMRoleGalleryItemIcon -Name $galItem.Name -Publisher
$galItem.Publisher -Version $galItem.Version -IconFilename "contoso.ico"
```

Get-SCSPFVMRoleGalleryItemPackage

Get-SCSPFVMRoleGalleryItemPackage

Gets the package that created an item in the gallery.

Syntax

```
Parameter Set: FromGalleryItemParameterSetName
Get-SCSPFVMRoleGalleryItemPackage [-VMRoleGalleryItem] <VMRoleGalleryItem> [
<CommonParameters>]

Parameter Set: FromGalleryItemNameVersionPublisherParameterSetName
Get-SCSPFVMRoleGalleryItemPackage -Name <String> -Publisher <String> -Version <String> [
<CommonParameters>]
```

Detailed Description

The Get-SCSPFVMRoleGalleryItem cmdlets gets an object of type System.IO.Memory stream that contains the gallery item resource package.

Parameters

-Name<String>

Specifies the name that package that installs the gallery item.

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

-Publisher<String>

Specifies the publisher of the gallery package.

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

-Version<String>

Specifies the version of the gallery item.

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

-VMRoleGalleryItem<VMRoleGalleryItem>

Specifies a gallery item 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](#).

Examples

1: Get the package by name, publisher, and version

This command gets the package of the gallery item by specifying the name, publisher, and version of the gallery item.

```
PS C:\> $galPkg = Get-SCSPFVMRoleGalleryItemPackage -Name 570569955cbfb62b374358b34467020750f65c -Publisher Microsoft -Version 1.0.0.0
```

2: Get the package from a gallery item

The first command gets a gallery item from its name and stores it in the `$galItem` object variable. The next command gets the package with *VMRoleGalleryItem* parameter.

```
PS C:\> $galItem = Get-SCSPFVMRoleGalleryItem -Name 570569955cbfb62b374358b34467020750f65c
PS C:\> $galPkg = Get-SCSPFVMRoleGalleryItemPackage -VMRoleGalleryItem $galItem
```

Import-SCSpfVMRoleGalleryItem

Import-SCSpfVMRoleGalleryItem

Imports a gallery item.

Syntax

```
Parameter Set: FromGalleryItemPackageParameterSetName
Import-SCSpfVMRoleGalleryItem [-Package <Stream> ] [-Confirm] [-WhatIf] [
<CommonParameters>]

Parameter Set: FromGalleryItemPackageFilePathParameterSetName
Import-SCSpfVMRoleGalleryItem [-PackageFilePath <String> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The Import-SCSpfVMRoleGalleryItem imports a gallery item for use by portal applications. The virtual machine role item is imported into Virtual Machine Manager (VMM) and is tracked in the Service Provider Foundation database.

Parameters

-Package<Stream>

Specifies an System.IO.FileStream object that contains the resource package.

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

-PackageFilePath<String>

Specifies the full path to the resource package file, with the .resdefpkg extension.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1. Import an item.

The first command gets the path to the resource package and stores it in the \$Path variable. The second command gets a System.IO.FileStream object of the package. The third command imports the package.

```
PS C:\> $Path = "c:\packages\create.resdefpkg"
PS C:\> $FStream = New-Object IO.FileStream $Path, Open
PS C:\> Import-SCSPFVMRoleGalleryItem -Package $FStream
```

New-SCSPFETWEvent

New-SCSPFETWEvent

Adds a new event for ETW logging. Internal use only.

Syntax

```
Parameter Set: Default
New-SCSPFETWEvent -EventName <String> -ProviderName <String> [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The New-SCSPFETWEvent cmdlet adds an event for Event Tracing for Windows (ETW) to the event log. This cmdlet is for internal use only and is not meant to be used directly by Administrators.

Parameters

-EventName<String>

Specifies the name of the event to add to the ETW event log. Internal use only.

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

-ProviderName<String>

Specifies an event provider. Internal use only.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

PS C:\> # The New-SCSPFETWEvent cmdlet is for internal use only.

New-SCSPFServer

New-SCSPFServer

Registers a server in the Service Provider Foundation database.

Syntax

```
Parameter Set: Empty
New-SCSPFServer -Name <String> -ServerType <String> [-Stamps <Stamp[]> ] [-Confirm] [-
WhatIf] [ <CommonParameters>]
```

Detailed Description

The New-SCSPFServer cmdlet adds a server to the Service Provider Foundation database.

Parameters

-Name<String>

Specifies the name of the VMM server. Specify the fully qualified domain name (FQDN).

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

-ServerType<String>

Specifies the type of server.

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

-Stamps<Stamp[]>

Specifies the name of one or more stamp objects to associate with the new server. To obtain a stamp, use the Get-SCSPFStamp cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Add a server.

The first command gets a stamp.

The second command creates a server with a specified name, a VMM server type, and associates the server with the stamp.

```
PS C:\> $stamp = Get-SCSPFStamp -Name "UltimatePro"
```

```
PS C:\> New-SCSPFServer -Name "server23G" -ServerType 0 -Stamps $stamp
```

Related topics

[Get-SCSPFServer](#)

[Set-SCSPFServer](#)

[Remove-SCSPFServer](#)

New-SCSPFSetting

New-SCSPFSetting

Creates a setting for a variety of purposes.

Syntax

```
Parameter Set: Empty
New-SCSPFSetting -ServerName <String> -SettingType <String> -Value <String> [-Name <String>
] [-Server <Server> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The New-SCSPFSetting cmdlet creates a connection string for either a database connection or for a portal endpoint connection.

Parameters

-Name<String>

Specifies a friendly name for the setting.

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

-Server<Server>

Specifies the sever associated with the setting.

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

-Value<String>

Specifies the value for the setting.

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

-SettingType<String>

Specifies either DatabaseConnectionString or EndPointConnectionString.

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

-ServerName<String>

Specifies the name of the server the setting is associated with. The actual association is made by the *Server* parameter with a server object.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Create a setting

This command creates a new setting for a database connection. The setting is associated with the server stored in the \$curSvr variable as obtained from the [Get-SCSPFServer](#) cmdlet.

```
PS C:\> $setting = New-SCSPFSetting -Name mysetting -SettingType DatabaseConnectionString -
SettingString
"Server=myomdwserver\\myomdwinstance;Database=OperationsManagerDW;TrustedConnection=True;Con
nection Timeout=300" -Server $curSvr
```

New-SCSPFStamp

New-SCSPFStamp

Creates a logical scale unit for servers and instances of System Center components.

Syntax

Parameter Set: Empty

```
New-SCSPFStamp -Name <String> [-DataCenter <DataCenter> ] [-Offers <Offer[]> ] [-Servers  
<Server[]> ] [-Tenants <Tenant[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The New-SCSPFStamp cmdlet creates a new stamp, a logical scale unit that specifies a System Center component instance running on a server. As tenants need additional capacity, additional stamps can be provided.

Parameters

-DataCenter<DataCenter>

Specifies the name of a DataCenter object. To obtain a DataCenter object, use the New-Object cmdlet with Microsoft.SystemCenter.Foundation.SPFDData.Types.DataCenter specified for its TypeName parameter. Then set its Name property to the name of the your data center, as shown in the following example.

```
PS C:\> $DS = New-Object Microsoft.SystemCenter.Foundation.SPFDData.Types.DataCenter
```

```
PS C:\> $DS.Name = "DataCenter_NorthAmerica"
```

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

-Name<String>

Specifies a name for the stamp.

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

-Offers<Offer[]>

Specifies the name of one or more offer objects to associate with the new stamp. To obtain an offer, use the Get-SCSPFOffer cmdlet.

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

-Servers<Server[]>

Specifies the name of one or more server objects to associate with the new stamp. To obtain a server, use the Get-SCSPFServer cmdlet.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-Tenants<Tenant[]>

Specifies the name of one or more tenant objects to associate with the new stamp. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

<CommonParameters>

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

Examples

1: Create a stamp.

This command creates a new stamp.

```
PS C:\> New-SCSPFStamp -Name "Introductory"
```

Related topics

[Set-SCSPFStamp](#)

[Get-SCSPFStamp](#)

[Remove-SCSPFStamp](#)

New-SCSPFTenant

New-SCSPFTenant

Creates a new tenant for a hoster.

Syntax

```
Parameter Set: Empty
New-SCSPFTenant -Name <String> [-AccountStatus <String> ] [-Offer <Offer> ] [-Stamps
<Stamp[]> ] [-SubscriptionId <Guid> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromTenantCertParameterSetName
New-SCSPFTenant -Certificate <String> -IssuerName <String> -Name <String> [-AccountStatus
<String> ] [-Offer <Offer> ] [-Stamps <Stamp[]> ] [-SubscriptionId <Guid> ] [-Confirm] [-
WhatIf] [ <CommonParameters>]

Parameter Set: FromTenantIssuereParameterSetName
New-SCSPFTenant -IssuerName <String> -Key <String> -Name <String> [-AccountStatus <String> ]
[-Offer <Offer> ] [-Stamps <Stamp[]> ] [-SubscriptionId <Guid> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The New-SCSPFTenant cmdlet creates a new tenant in Service Provider Foundation. If you are a service provider, these are your paying customers. If you are using Service Provider Foundation in private cloud, these are the business units of your organization.

Parameters

-AccountStatus<String>

Sets the status of a tenant. Specify 0 for Active, 1 for Suspended.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-Certificate<String>

Specifies the path to the certificate file for the tenant.

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

-IssuerName<String>

Specifies the name of the party that issued the certificate for the tenant. This parameter is not to be confused with a trusted issuer object, as it is metadata to verify a token submitted by the tenant.

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

-Key<String>

Specifies the public key that validates a signed token submitted by a tenant in claims-based authentication.

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

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

-Name<String>

Specifies the name to give to the tenant.

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

-Offer<Offer>

Specifies the name of one or more offer objects to associate with the new tenant. To obtain an offer, use the Get-SCSPFOffer cmdlet.

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

-Stamps<Stamp[]>

Specifies the name of one or more stamp objects to associate with the new tenant. To obtain a stamp, use the Get-SCSPFStamp cmdlet.

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

-SubscriptionId<Guid>

Specifies the subscription identifier (as a GUID) for a tenant. This value cannot be changed after the tenant is created.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Onboard a tenant.

The first two commands create the \$cert variable that contains the tenant's issued certificate.

The third command creates the \$key variable that contains the certificate's public key.

The fourth command creates the \$subID variable that contains a GUID for the subscription ID.

The fifth command creates the tenant with the name of the trusted issuer of the certificate, the public key, a specified name, and the subscription ID.

```
PS C:\> $path = "C:\Temp\ADatum29D.cer"
```

```
PS C:\> $cert = New-Object  
System.Security.Cryptography.X509Certificates.X509Certificate2($path)
```

```
PS C:\> $key = [Convert]::ToBase64String($cert.RawData)
```

```
PS C:\> $subID = [System.Guid]::NewGuid().ToString()
```

```
PS C:\> New-SCSPFTenant -IssuerName "Woodgrove" -Key $key -Name "ADatum" -SubscriptionID  
$subID
```

Related topics

[Remove-SCSPFTenant](#)

[Set-SCSPFTenant](#)

[Get-SCSPFTenant](#)

New-SCSPFTenantUserRole

New-SCSPFTenantUserRole

Creates an association between a tenant and one or more self-service user roles.

Syntax

```
Parameter Set: Default
New-SCSPFTenantUserRole -Name <String[]> -Tenant <Tenant> [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The New-SCSPFTenantUserRole cmdlet creates an association between a tenant and one or more self-service user roles in Service Provider Foundation. The self-service roles must already have been created in Virtual Machine Manager (VMM), or by using the New-SCUserRole cmdlet in the "virtualmachinemanager" Windows PowerShell module.

You cannot use this cmdlet to create an association with a tenant administrator user role. The tenant administrator user role is created automatically when the tenant is created by the [New-SCSPFTenant](#) cmdlet.

Parameters

-Name<String[]>

Specifies the name of one or more tenant user roles.

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

-Tenant<Tenant>

Specifies a tenant object to associate with the new tenant user role. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Create a tenant user role.

The first command defines an array of self-service user roles.

The second command gets a tenant.

The third command creates the tenant user role for the tenant with its individual self-service user roles.

```
PS C:\> $roles = "Assistant", "Admin", "Developer"
```

```
PS C:\> $tnnt = Get-SCSPFTenant -Name "Contoso"
```

```
PS C:\> New-SCSPFTenantUserRole -Name $roles -Tenant $tnnt
```

Related topics

[Get-SCSPFTenantUserRole](#)

[Remove-SCSPFTenantUserRole](#)

New-SCSPFTrustedIssuer

New-SCSPFTrustedIssuer

Creates a trusted issuer for claims-based authentication.

Syntax

```
Parameter Set: Empty
New-SCSPFTrustedIssuer -Key <String> -Name <String> [-Tenant <Tenant> ] [-Confirm] [-WhatIf]
[ <CommonParameters>]
```

Detailed Description

The New-SCSPFTrustedIssuer cmdlet creates certification credentials for claims-based authentication in Service Provider Foundation. The hoster obtains the public key of the certificate's key value pair from the tenant (or on behalf of a tenant). The trusted issuer object is then used to validate signed tokens from the tenant so that access to resources can be granted.

Parameters

-Key<String>

Specifies the public key that validates a signed token submitted by a tenant in claims-based authentication.

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

-Name<String>

Specifies a name to give to the trusted issuer.

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

-Tenant<Tenant>

Specifies a tenant object to associate with the new trusted issuer. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Create a trusted issuer.

This command creates a trusted issuer with a previously defined key and a specified name.

```
PS C:\> New-SCSPFTrustedIssuer -Key $key -Name "Trey"
```

Related topics

[Remove-SCSPFTrustedIssuer](#)

[Get-SCSPFTrustedIssuer](#)

Remove-SCSPFServer

Remove-SCSPFServer

Removes one or more server objects.

Syntax

Parameter Set: FromServerParameterSetName
Remove-SCSPFServer -Server <Server[]> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Remove-SCSPFServer cmdlet removes one or more VMM servers from the Service Provider Foundation database.

Parameters

-Server<Server[]>

Specifies the name of one or more VMM server objects. To obtain a server, use the Get-SCSPFServer cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Remove a server.

The first command gets the server object named server23G and stores the object in the \$srvr variable.

The second command removes the server.

```
PS C:\> $srvr = Get-SCSPFServer -Name "server23G"
```

```
PS C:\> Remove-SCSPFServer -Server $srvr
```

Related topics

[New-SCSPFServer](#)

[Set-SCSPFServer](#)

[Get-SCSPFServer](#)

Remove-SCSPFSetting

Remove-SCSPFSetting

Removes a setting.

Syntax

Parameter Set: FromSettingParameterSetName
Remove-SCSPFSetting -SpfSetting <SpfSetting[]> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Remove-SCSPFSetting cmdlet removes a connection string for either a database connection or for a portal endpoint connection.

Parameters

-SpfSetting<SpfSetting[]>

Specifies one or more settings to remove.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Remove a setting

This command obtains a setting by its name and then sends it to the next command in the pipeline to be removed.

```
PS C:\> $OldSetting = Get-SCSPFSetting -Name 'mysetting' | Remove-SCSPFSetting
```

Remove-SCSPFStamp

Remove-SCSPFStamp

Removes one or more stamp objects.

Syntax

Parameter Set: FromStampParameterSetName
Remove-SCSPFStamp -Stamp <Stamp[]> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Remove-SCSPFStamp cmdlet removes one or more stamps from the Service Provider Foundation database.

Parameters

-Stamp<Stamp[]>

Specifies the name of one or more stamp objects. To obtain a stamp, use the Get-SCSPFStamp cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Remove a stamp.

The first command gets the stamp object named DeLuxe and stores the object in the \$stmp variable.

The second command removes the stamp.

```
PS C:\> $stmp = Get-SCSPFStamp -Name "DeLuxe"
```

```
PS C:\> Remove-SCSPFStamp -Stamp $stmp
```

Related topics

[Set-SCSPFStamp](#)

[New-SCSPFStamp](#)

[Get-SCSPFStamp](#)

Remove-SCSPFTenant

Remove-SCSPFTenant

Removes one or more tenant objects.

Syntax

Parameter Set: Default

Remove-SCSPFTenant -Tenant <Tenant[]> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Remove-SCSPFTenant cmdlet removes one or more tenants from the Service Provider Foundation database.

Parameters

-Tenant<Tenant[]>

Specifies the name of one or more tenant objects. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Remove a tenant.

The first command gets the tenant object named Fabrikam and stores the object in the \$tnnt variable.

The second command removes the tenant.

```
PS C:\> $tnnt = Get-SCSPFTenant -Name "Fabrikam"
```

```
PS C:\> Remove-SCSPFTenant -Tenant $tnnt
```

Related topics

[New-SCSPFTenant](#)

[Set-SCSPFTenant](#)

[Get-SCSPFTenant](#)

Remove-SCSPFTenantUserRole

Remove-SCSPFTenantUserRole

Removes one or more tenant user role objects.

Syntax

Parameter Set: Default
Remove-SCSPFTenantUserRole -UserRole <Role[]> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Remove-SCSPFTenantUserRole cmdlet removes one or more tenant user roles from the Service Provider Foundation database.

Parameters

-UserRole<Role[]>

Specifies the name of one or more tenant user role objects. To obtain a tenant user role, use the Get-SCSPFTenantUserRole cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Remove a tenant user role.

The first command gets the tenant user role object by its ID and stores the object in the \$tur variable.

The second command removes the tenant user role.

```
PS C:\> $tur = Get-SCSPFTenantUserRole -ID 9adb708d-f47e-4dda-9e56-91e1eb2808df
```

```
PS C:\> Remove-SCSPFTenantUserRole -UserRole $tur
```

Related topics

[New-SCSPFTenantUserRole](#)

[Get-SCSPFTenantUserRole](#)

Remove-SCSPFTrustedIssuer

Remove-SCSPFTrustedIssuer

Removes one or more trusted issuer objects.

Syntax

Parameter Set: Default

Remove-SCSPFTrustedIssuer -Issuer <Issuer[]> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Remove-SCSPFTrustedIssuer cmdlet removes one or more trusted issuers from the Service Provider Foundation database.

Parameters

-Issuer<Issuer[]>

Specifies one or more trusted issuer objects. To obtain a trusted issuer object, use the Get-SCSPFTrustedIssuer cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Remove a trusted issuer.

The first command gets the trusted issuer object by its ID and stores the object in the \$ti variable.

The second command removes the trusted issuer.

```
PS C:\> $ti = Get-SCSPFTrustedIssuer -ID 938d2412-ba64-497e-b97d-12991087d66c
```

```
PS C:\> Remove-SCSPFTrustedIssuer -Issuer $ti
```

Related topics

[New-SCSPFTrustedIssuer](#)

[Get-SCSPFTrustedIssuer](#)

Remove-SCSPFVMRoleGalleryItem

Remove-SCSPFVMRoleGalleryItem

Removes an item from the gallery.

Syntax

Parameter Set: FromGalleryItemParameterSetName
Remove-SCSPFVMRoleGalleryItem [-Name] <String> [-Version] <VMRoleGalleryItem[]> [-VMRoleGalleryItem] <VMRoleGalleryItem[]> [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Remove-SCSPFVMRoleGalleryItem cmdlet removes a gallery item from VMM and the Service Provider Foundation database.

Parameters

-VMRoleGalleryItem<VMRoleGalleryItem[]>

Specifies one or more gallery items to be removed.

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

-Name<String>

Specifies the publisher of the gallery item to be removed.

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

-Version<VMRoleGalleryItem[]>

Specifies the version of the item to be removed.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Remove a gallery item

The first command gets an item by its name and stores it in the \$oldItem variable. The next command removes the item.

```
PS C:\> $oldItem = Get-SCSPFVMRoleGalleryItem -Name 570569955cbfb62b374358b34467
020750f65c
PS C:\> Remove-SCSPFVmRoleGalleryItem -VMRoleGalleryItem $oldItem
```

Set-SCSPFConnectionString

Set-SCSPFConnectionString

Sets the connection string to the Service Provider Foundation database. Requires running Windows PowerShell as Administrator.

Syntax

Parameter Set: FromConnectionStringConnectionStringParameterSetName
Set-SCSPFConnectionString -ConnectionString <String> [-Confirm] [-WhatIf] [
<CommonParameters>]

Detailed Description

The Set-SCSPFConnectionString cmdlet sets the database connection string to the Service Provider Foundation database that is used by the server running Service Provider Foundation. The database can be on a separate server. This cmdlet requires running Windows PowerShell as Administrator.

Parameters

-ConnectionString<String>

Specifies the database connection string to a Service Provider Foundation database.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Set the connection string.

This command sets the connection string to the Service Provider Foundation database on a server named serverSFPDB_7.

```
PS C:\> Set-SCSPFConnectionString -ConnectionString "Data
Source=serverSFPDB_7,1433;Database=SCSPFDB;Integrated
Security=True;MultipleActiveResultSets=True;"
```

Related topics

[Get-SCSPFConnectionString](#)

Set-SCSPFEventRegistration

Set-SCSPFEventRegistration

Sets a Service Management Automation runbook event.

Syntax

```
Parameter Set: FromEventRegistrationResourceActionEnabledParameterSetName
Set-SCSPFEventRegistration -Action {Action_Pause | Action_Publish | Action_Repair |
Action_Restart | Action_Resume | Action_Scale | Action_Shutdown | Action_Start | Action_Stop
| Action_Unpublish | Create | Delete | Update | UpdateStatus} -Resource <String> [-Enabled]
[-RunbookName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The Set-SCSPFEventRegistration cmdlet registers an event for Service Management Automation to automate a runbook.

Parameters

-Action<EventRegistrationActionNames>

Specifies the action (Create, Update, Delete, Action) on a resource for which Orchestrator event is registered.

The acceptable values for this parameter are:

Action_Pause	
Action_Publish	
Action_Repair	
Action_Restart	
Action_Resume	
Action_Scale	
Action_Shutdown	
Action_Start	

Action_Stop	
Action_Unpublish	
Create	
Delete	
Update	
UpdateStatus	

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

-Enabled

Enables or disables the event from being used.

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

-Resource<String>

Specifies the resource for which the Orchestrator event is registered.

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

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

-RunbookName<String>

Specifies the runbook that should be invoked when the Orchestrator event is raised.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

Required?	false
-----------	-------

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

<CommonParameters>

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

Examples

1: Set an event

This command sets a runbook to be invoked when the Create event for a new virtual machine is raised.

```
PS C:\> Set-SCSPFEventRegistration -ResourceName "VMM.VirtualMachine" - ActionName "Create"
-RunbookName "Invoke-SampleCmdlet"
```

Set-SCSPFExtensibleEventHandler

Set-SCSPFExtensibleEventHandler

Sets an extensible event handler in Service Provider Foundation.

Syntax

Parameter Set: Default
Set-SCSPFExtensibleEventHandler -EventName <String> [-Enable] [-OrchestratorUri <String>]
[-RunbookPath <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Set-SCSPFExtensibleEvent cmdlet invokes a runbook in System Center 2012 - Orchestrator to be run whenever a new virtual machine or new service is created. The virtual machine or service must be created by calls to Service Provider Foundation with the VMM Service. Service Provider Foundation raises internal events to invoke the runbook and the runbook will continue to be invoked continuously as long as the extensible event handler is enabled.

Service Provider Foundation will not invoke the runbook if the virtual machine or service was created by other means such as by using Windows PowerShell cmdlets or by using the console in System Center 2012 – Virtual Machine Manager (VMM).

To support the infrastructure for invoking a runbook, Service Provider Foundation calls the **Start-SCOrchestratorRunbook** cmdlet internally and it does to need to be explicitly called by the user.

Parameters

-Enable

Enables the extensible event handler. Specify this parameter by itself to enable the handler. There is no associated value. Omit this parameter to disable the handler.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

-EventName<String>

Specifies either **VirtualMachineCreated** or **ServiceCreated** for the event that invokes an Orchestrator runbook. These are the only permitted values for this parameter.

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

-OrchestratorUri<String>

Specifies the URI to the Orchestrator web service as specified in the web.config file at C:\Program Files (x86)\Microsoft System Center 2012\Orchestrator\Orchestration Console.

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

-RunbookPath<String>

Specifies the path to a runbook in System Center 2012 - Orchestrator.

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

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Set an event handler.

This command enables the event handler to invoke a Runbook.

```
PS C:\> Set-SCSPFExtensibleEventHandler -EventName "VirtualMachineCreated" -OrchestratorUri  
"https://scxspf-ws8-24:82/Orchestrator2012/Orchestrator.svc" -RunbookPath "\SPF  
Runbooks\Extensibility\VM Created" -Enable
```

2: Disable an event handler.

This command disables the event handler from invoking a runbook because the *Enable* parameter is omitted.

```
PS C:\> Set-SCSPFExtensibleEventHandler -EventName "VirtualMachineCreated" -OrchestratorUri  
"" -RunbookPath ""
```

Set-SCSPFServer

Set-SCSPFServer

Associates a server with one or more stamps.

Syntax

```
Parameter Set: Default
Set-SCSPFServer -Server <Server> [-Stamps <Stamp[]> ] [-Confirm] [-WhatIf] [
<CommonParameters>]
```

Detailed Description

The Set-SCSPFServer cmdlet associates a server with a stamp in Service Provider Foundation. Currently, you can associate only a VMM server with a stamp.

If you want to add an association of a stamp to a server without changing the stamps currently associated with the server, use the [Set-SCSPFStamp](#) cmdlet.

Parameters

-Server<Server>

Specifies the server object to be set to new values by the other parameters in this cmdlet. To obtain a server, use the Get-SCSPFServer cmdlet.

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

-Stamps<Stamp[]>

Specifies one or more stamps to associate with the server. To obtain a stamp, use the Get-SCSPFStamp cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Set a server to a stamp.

The first two commands get a stamp and a server.

The third command associates the server with the stamp.

```
PS C:\> $stamp_silver = Get-SCSPFStamp -Name "Silver"
```

```
PS C:\> $server = Get-SCSPFServer -Name "serverD7"
```

```
PS C:\> Set-SCSPFServer -Server $server -Stamps $stamp_silver
```

Related topics

[New-SCSPFServer](#)

[Get-SCSPFServer](#)

[Remove-SCSPFServer](#)

Set-SCSPFStamp

Set-SCSPFStamp

Associates a stamp with offers, servers, and tenants.

Syntax

```
Parameter Set: Default
Set-SCSPFStamp -Stamp <Stamp> [-Offers <Offer[]> ] [-Servers <Server[]> ] [-Tenants
<Tenant[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The Set-SCSPFStamp cmdlet associates a stamp with one or more servers, one or more tenants, and one or more offers in Service Provider Foundation.

If you want to add an offer, server, or tenant to a stamp without changing the currently associated objects, use the [Set-SCSPFOffer](#), [Set-SCSPFServer](#), or [Set-SCSPFTenant](#) cmdlets.

Parameters

-Offers<Offer[]>

Specifies one or more offers to associate with the stamp.To obtain an offer, use the Get-SCSPFOffer cmdlet.

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

-Servers<Server[]>

Specifies one or more servers to associate with the stamp. To obtain a server, use the Get-SCSPFServer cmdlet.

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

-Stamp<Stamp>

Specifies the stamp object to be set to new values by the other parameters in this cmdlet. To obtain a stamp, use the Get-SCSPFStamp cmdlet.

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

-Tenants<Tenant[]>

Specifies the name of one or more tenant objects. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Set a stamp to a tenant.

The first two commands get a stamp and a tenant.

The third command associates the tenant with the stamp.

```
PS C:\> $stamp = Get-SCSPFStamp -ID db656655-68ff-4a21-bd1d-0a06bdbbc762f
```

```
PS C:\> $tnnt = Get-SCSPFTenant -Name "AdventureWorks"  
PS C:\> Set-SCSPFStamp -Stamp $stamp -Tenants $tnnt
```

Related topics

[Get-SCSPFStamp](#)

[New-SCSPFStamp](#)

[Remove-SCSPFStamp](#)

Set-SCSPFTenant

Set-SCSPFTenant

Sets resources and data for a tenant.

Syntax

```
Parameter Set: Empty
Set-SCSPFTenant -Tenant <Tenant> [-AccountStatus <String> ] [-Offer <Offer> ] [-Stamps
<Stamp[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]

Parameter Set: FromTenantIssuereParameterSetName
Set-SCSPFTenant -IssuerName <String> -Key <String> -Tenant <Tenant> [-AccountStatus <String>
] [-Offer <Offer> ] [-Stamps <Stamp[]> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The Set-SCSPFTenant cmdlet associates a tenant in Service Provider Foundation with various data, including an offer, stamps, and values for claims-based authentication.

If you want to add an association of a stamp to a tenant without changing the stamps currently associated with the tenant, use the [Set-SCSPFStamp](#) cmdlet.

Parameters

-AccountStatus<String>

Sets the status of a tenant. Specify 0 for Active, 1 for Suspended.

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

-IssuerName<String>

Specifies the name of the party that issued the certificate for the tenant. This parameter is not to be confused with a trusted issuer object, as it is metadata to verify a token submitted by the tenant.

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

-Key<String>

Specifies the public key that validates a signed token submitted by a tenant in claims-based authentication.

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

-Offer<Offer>

Specifies an offer to associate with the tenant. To obtain an offer, use the Get-SCSPFOffer cmdlet.

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

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

-Stamps<Stamp[]>

Specifies one or more stamps to associate with the tenant. To obtain a stamp, use the New-SCSPFStamp cmdlet.

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

-Tenant<Tenant>

Specifies the tenant object to be set to new values by the other parameters in this cmdlet. To obtain a tenant, use the Get-SCSPFTenant cmdlet.

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

-Confirm

Prompts you for confirmation before executing the command.

Required?	false
Position?	named

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Set a tenant's account status.

This command sets the account status for a tenant to 1 (Suspended).

```
PS C:\> Set-SCSPFTenant -Tenant (Get-SCSPFTenant -Name "ADatum") -AccountStatus 1
```

2: Set a tenant to collection of stamps.

The first two commands get a tenant and a collection of stamps.

The third command associates the tenant with the stamps.

```
PS C:\> $tnnt = Get-SCSPFTenant -Name "Northwind"
PS C:\> $stmps = Get-SCSPFServer | Where-Object {
>> $_.Name -contains "silver">> }
PS C:\> Set-SCSPFTenant -Tenant $tnnt -Stamps $stmps
```

Related topics

[New-SCSPFTenant](#)

[Remove-SCSPFTenant](#)

[Get-SCSPFTenant](#)

Set-SCSPFVMRoleGalleryItem

Set-SCSPFVMRoleGalleryItem

Sets the publishing status of an item in the gallery.

Syntax

Parameter Set: FromGalleryItemParameterSetName
Set-SCSPFVMRoleGalleryItem -Package <Stream> -VmRoleGalleryItem <VMRoleGalleryItem> [-PublishingStatus <String>] [-Confirm] [-WhatIf] [<CommonParameters>]

Detailed Description

The Set-SCSPFVMRoleGalleryItem cmdlets sets the status of an item in the gallery to either publish or un-publish.

Parameters

-PublishingStatus<String>

Specifies either published or un-unpublished.

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

-VmRoleGalleryItem<VMRoleGalleryItem>

Specifies the gallery item to set.

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

-Package<Stream>

Specifies an object of type System.IO.MemoryStream that contains the resource package of the gallery item.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

1: Set a gallery item to publish

This command sets an item to be published in the gallery.

```
PS C:\> Set-SCSPFVmRoleGalleryItem -VmRoleGalleryItem $gall7 -PublishingStatus published
```

Start-SCOrchestratorRunbook

Start-SCOrchestratorRunbook

Invokes a runbook in System Center 2012 - Orchestrator. Internal use only.

Syntax

Parameter Set: Default

```
Start-SCOrchestratorRunbook -RunbookPath <String> -Uri <String> [-RunbookParameters  
<Hashtable> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Detailed Description

The Start-SCOrchestratorRunbook cmdlet invokes the runbook as specified by the RunbookPath parameter for the Set-SCSPFExtensibleEventHandler command. This action is performed automatically by Service Provider Foundation and is not meant to be used by Administrators.

Parameters

-RunbookParameters<Hashtable>

Specifies the parameters for a runbook in System Center 2012 - Orchestrator.

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

-RunbookPath<String>

Specifies the path to a runbook in Orchestrator.

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

-Uri<String>

Specifies the Uri to the runbook in Orchestrator.

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

-Confirm

Prompts you for confirmation before executing the command.

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

-WhatIf

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

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

<CommonParameters>

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

Examples

PS C:\> # The Start-SCOrchestratorRunbook cmdlet is for internal use only.
