Cisco UCS Integration Pack User Guide, Release 1.0

For Microsoft System Center 2012 and 2012 SP1 Orchestrator
April 2013

Cisco Systems, Inc.
www.cisco.com

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco website at www.cisco.com/go/offices.

Text Part Number OL-29521-01

Cisco UCS Integration Pack User Guide, Release 1.0

Preface

This preface includes the following sections:

Audience

This guide is intended primarily for data center administrators with responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security

Organization

This guide includes the following sections:

Chapter	Title	Description
1	Overview	Describes a general overview of the product and includes system requirements.
2	Integration Pack Installation	Describes how to install, register, deploy, and uninstall the Integration Pack.
3	Configuring the Integration Pack	Describes how to configure a PSModule Path and how to configure activity properties.
4	Cisco UCS Activities	Describes Cisco UCS Integration Pack activities.

Conventions

This document uses the following conventions:

Convention	Indication
bold font	Commands and keywords and user-entered text appear in **bold** font.
italic font	Document titles, new or emphasized terms, and arguments for which you supply values are in *italic* font.
[ ]	Elements in square brackets are optional.
{x | y | z }	Required alternative keywords are grouped in braces and separated by vertical bars.
[x | y | z]	Optional alternative keywords are grouped in brackets and separated by vertical bars.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.
`courier` font	Terminal sessions and information the system displays appear in `courier` font.
< >	Nonprinting characters such as passwords are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Tip Means *the following information will help you solve a problem*. The tips information might not be troubleshooting or even an action, but could be useful information, similar to a Timesaver.

Caution Means *reader be careful*. In this situation, you might perform an action that could result in equipment damage or loss of data.

Timesaver Means *the described action saves time*. You can save time by performing the action described in the paragraph.

Warning **IMPORTANT SAFETY INSTRUCTIONS**

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of

each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Warning **Statements using this symbol are provided for additional information and to comply with regulatory and customer requirements.**

Related Documentation

Cisco UCS Documentation Roadmaps

For a complete list of all B-Series documentation, see the *Cisco UCS B-Series Servers Documentation Roadmap* available at the following URL: http://www.cisco.com/go/unifiedcomputing/b-series-doc.

Other Cisco UCS Documentation Resources

An ISO file containing all B and C-Series documents is available at the following URL: http://www.cisco.com/cisco/software/type.html?mdfid=283853163&flowid=25821. From this page, click **Unified Computing System (UCS) Documentation Roadmap Bundle**.

The ISO file is updated after every major documentation release.

Follow Cisco UCS Docs on Twitter to receive document update notifications.

Cisco UCS Integration Pack User Guide, Release 1.0 for Microsoft System Center 2012 and 2012 SP1 Orchestrator Resources

More information about Cisco UCS Integration Pack User Guide, Release 1.0 for Microsoft System Center 2012 and 2012 SP1 Orchestrator is available at the following URL: http://www.cisco.com/en/US/partner/products/ps11724/tsd_products_support_series_home.html

General Cisco UCS Resources

More information about Cisco UCS is available at the following URL: http://www.cisco.com/en/US/products/ps10265/index.html

Microsoft System Centre 2012 Orchestrator Resources

More information on Microsoft System Centre 2012 Orchestrator is available on the Microsoft TechNet site at the following URL: http://technet.microsoft.com

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to ucs-docfeedback@cisco.com. We appreciate your feedback.

Obtaining Documentation and Submitting a Service Request

For information on obtaining documentation, submitting a service request, and gathering additional information, see *What's New in Cisco Product Documentation* at: http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html.

Subscribe to *What's New in Cisco Product Documentation*, which lists all new and revised Cisco technical documentation, as an RSS feed and deliver content directly to your desktop using a reader application. The RSS feeds are a free service.

Overview

This chapter includes the following sections:

Cisco UCS Integration Pack for Microsoft System Center 2012 and 2012 SP1 Orchestrator

The Cisco UCS Integration Pack is an add-on for Microsoft System Center 2012 and 2012 SP1 Orchestrator (SCO) that enables you to automate Cisco UCS Manager tasks. You can use Cisco UCS Integration Pack to create workflows that interact with and transfer information to other Microsoft System Center products such as Microsoft System Center 2012 Operations Manager.

System Requirements

Note Refer to the *Release Notes for Cisco UCS Integration Pack, Release 1.0* for the most up-to-date information about Release 1.0 requirements.

Note You can find installation and upgrade information for Microsoft System Center Orchestator on the Microsoft website.

Before installing Cisco UCS Integration Pack, ensure the system meets or exceeds the minimum requirements:

Software Requirements

- Cisco UCS PowerTool, Release 1.0 or later
- Microsoft System Center 2012 or 2012 SP1 - Orchestrator
- Windows PowerShell 2.0 or later

Installing the Integration Pack

This chapter includes the following sections:

Note For installation information on Microsoft System Centre 2012 Orchestrator, refer to the Microsoft TechNet website.

Integration Pack Installation Overview

Installing and running the Integration Pack includes the following tasks:

Step 1 Close any running instances of Microsoft SCO 2012 Runbook Designer.

Step 2 Download and unzip the **Cisco_UCS_OIP_1.0.zip** file. The compressed **.zip** file includes the Integration Pack **.oip** installation file. The .zip file can be downloaded from the software download site on Cisco.com.

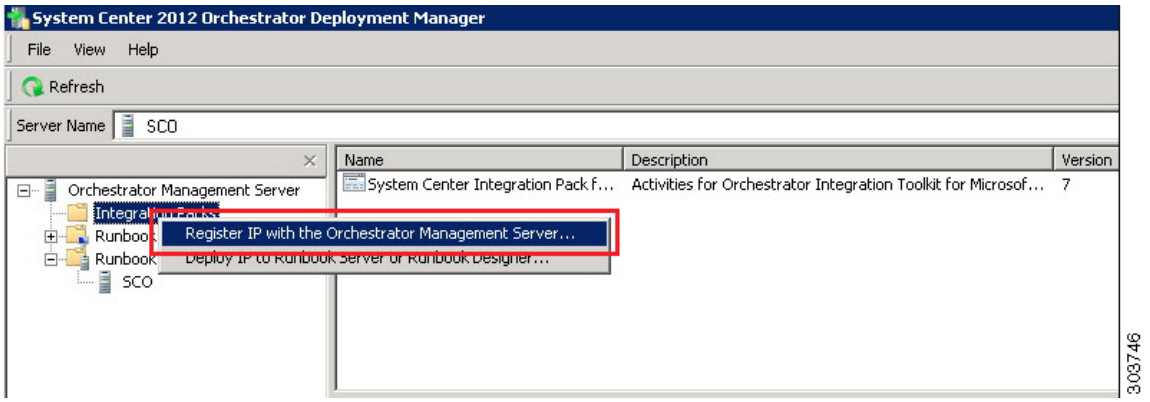
Step 3 Register the integration pack with the Orchestrator Management Server. See Registering the Integration Pack.

Step 4 Deploy the registered integration pack to the Runbook Servers and Runbook Designer. See Deploying the Integration Pack.

Registering the Integration Pack

Step 1 Launch the System Center 2012 Orchestrator Deployment Manager.

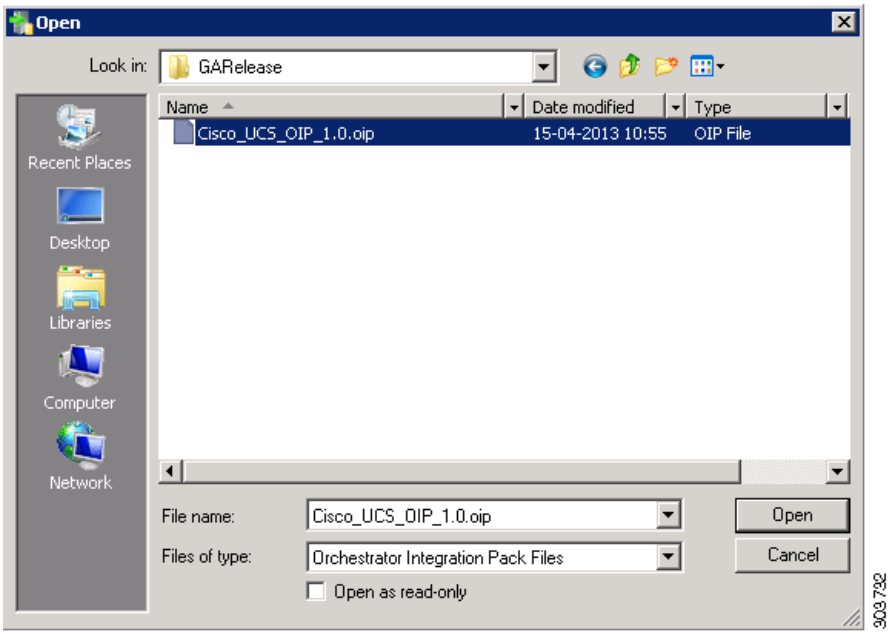
Step 2 In the left pane, right click **Integration Packs** and choose **Register IP with the Orchestrator Management Server**.



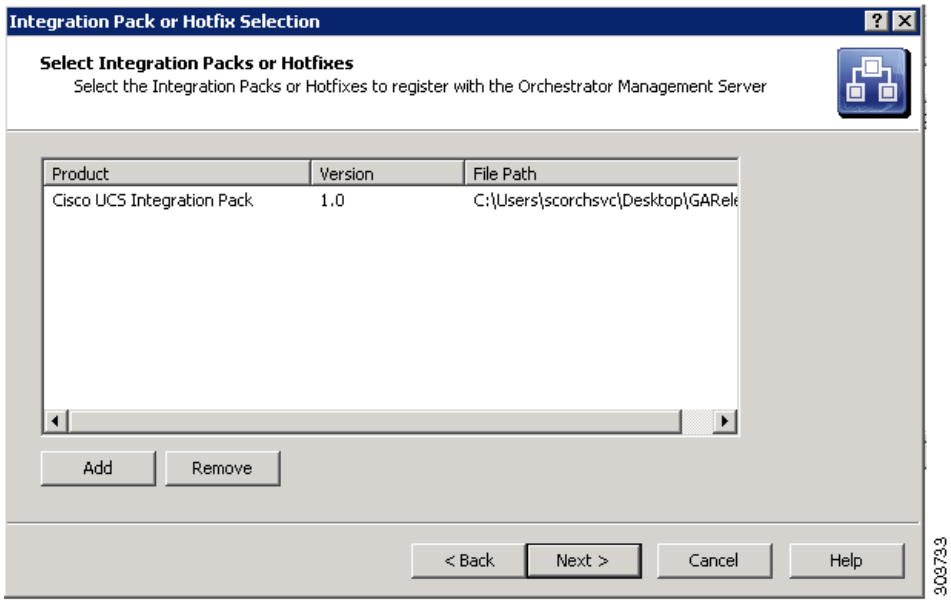
Step 3 In the **Integration Pack Registration Wizard** window, click **Next**.

Step 4 In the **Integration Pack or Hotfix Selection** window, click **Add**.

Step 5 In the **Open** window, locate and choose the **Cisco_UCS_OIP_1.0.oip** file that was downloaded earlier and click **Open**.



Step 6 In the **Integration Pack or Hotfix Selection** window, click **Next**.



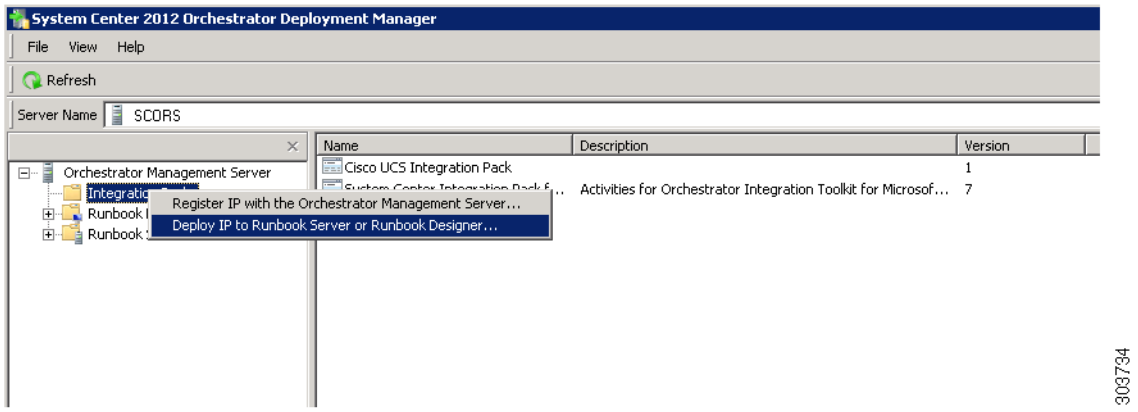
Step 7 In the **Completing Integration Pack Registration Wizard**, click **Finish** to complete the Integration Pack registration.

Step 8 In the **End User License Agreement** window, click **Accept** after reading the agreement. This completes the Integration Pack registration.

Deploying the Integration Pack

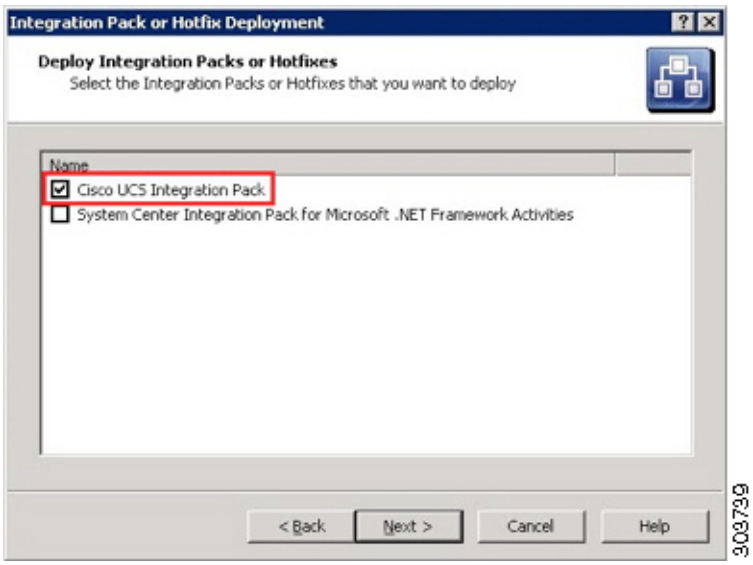
Step 1 Launch the System Center 2012 Orchestrator Deployment Manager.

Step 2 In the left pane, right-click **Integration Packs** and choose **Deploy IP to Runbook Server or Runbook Designer**.



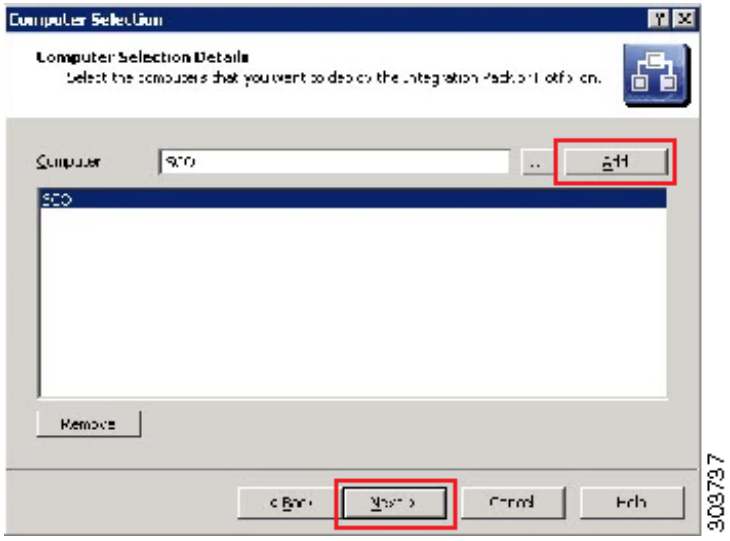
Step 3 In the **Integration Pack Deployment Wizard** window, click **Next**.

Step 4 In the **Integration Pack or Hotfix Deployment** window, select the **Cisco UCS Integration Pack** and click **Next**.

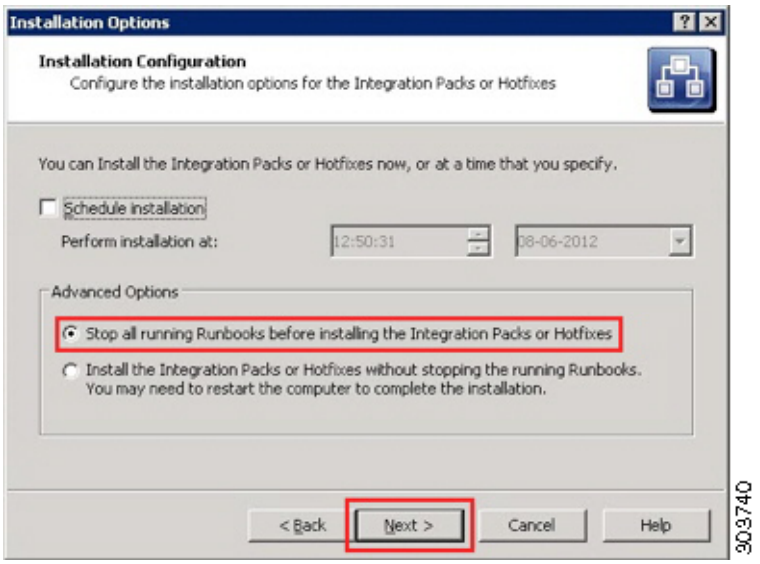


Step 5 In the **Computer** field of the **Computer Selection** window, choose the computer on which you wish to deploy the Integration Pack, click **Add**, and then click **Next**.

You can specify more than one computer on which to deploy the Integration Pack.



Step 6 In the **Installation Options** window, by default **Stop all running Runbooks before installing the Integration Packs or Hotfixes** is selected. Click **Next**.



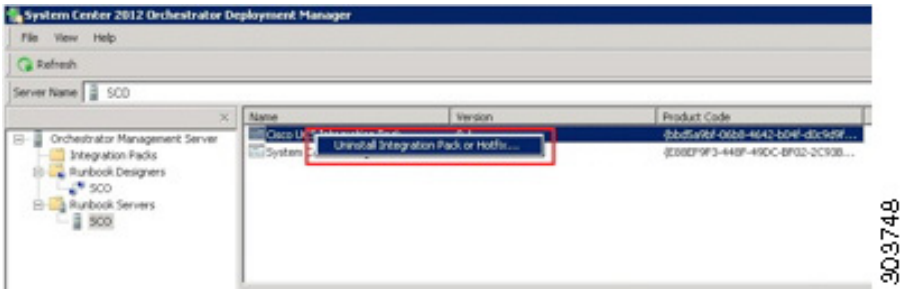
Step 7 In the **Completing Integration Pack Deployment Wizard**, click **Finish** to complete the process.

Uninstalling the Integration Pack

To completely remove the Integration Pack, first uninstall the Integration Pack and then unregister the Integration Pack.

Step 1 In the left navigation pane, expand **Runbook Servers** and select the server where the Integration Pack is deployed. The Integration Packs are listed in the work pane.

Step 2 Right-click a Cisco UCS Integration Pack and choose **Uninstall Integration Pack or Hotfix...** from the right-click menu.

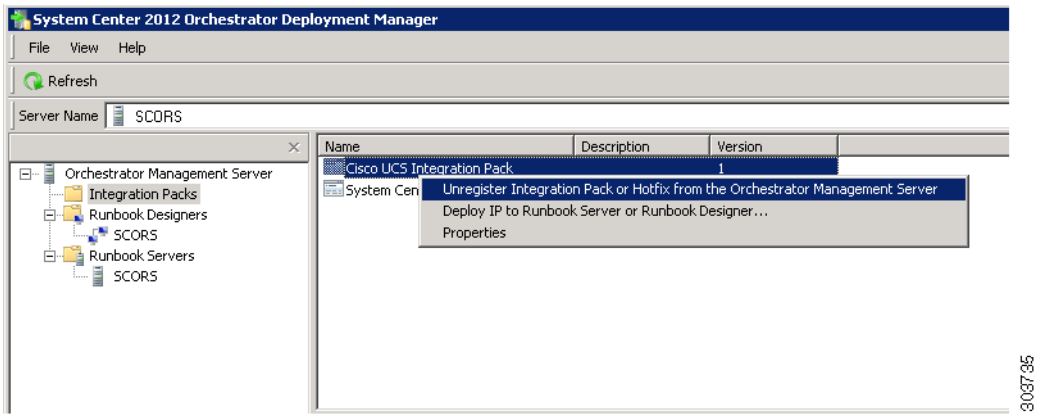


After uninstalling the Cisco UCS Integration Pack, continue to unregister the Integration Pack to completely remove the Integration Pack from the server.

Unregistering an Integration Pack

Step 1 In the left navigation pane, expand **Integration Packs.** The Integration Packs are listed in the work pane.

Step 2 Right-click a Cisco UCS Integration Pack and choose **Unregister Integration Pack or Hotfix from the Orchestrator Management Server**.



Configuring the Integration Pack

This chapter includes the following sections:

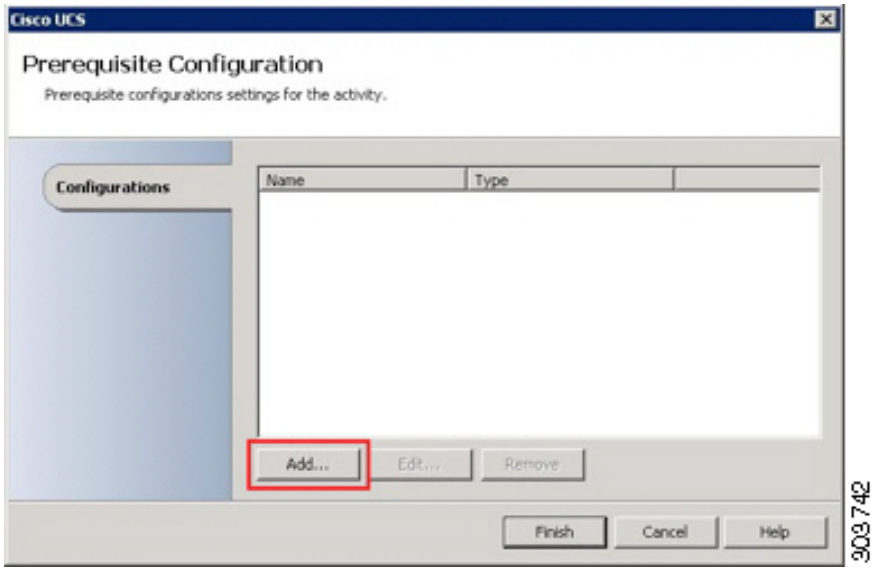
Configuring a PSModule Path

You can configure a PSModule path to establish a link between the Microsoft System Centre 2012 Orchestrator and the Cisco UCS PowerTool module.

Step 1 Launch the SCO 2012 Runbook Designer.

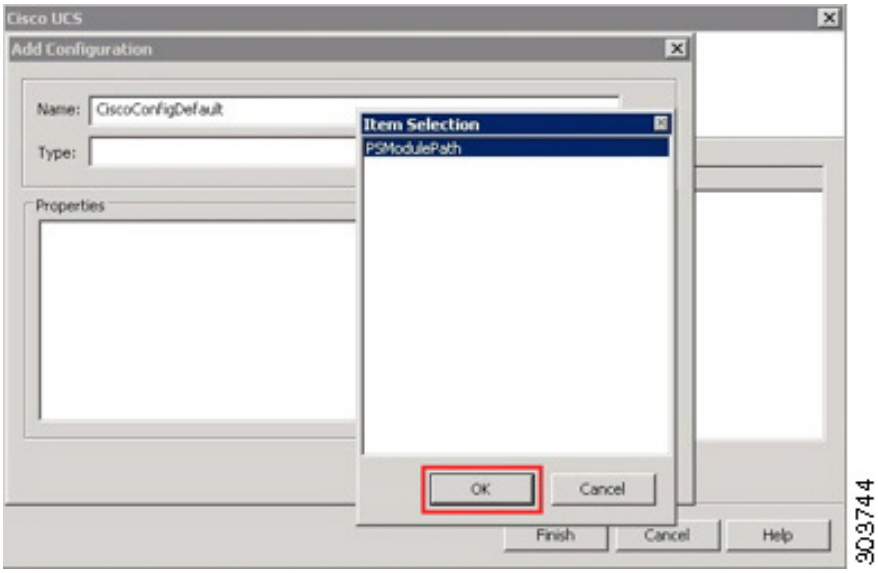
Step 2 Choose **Options** > **Cisco UCS**.

Step 3 In the **Prerequisite Configuration** window click **Add**.



Step 4 In the **Add Configuration** window, use the **Name** field to enter a name for the configuration. Click the Ellipsis (**...**) button to specify a configuration type.

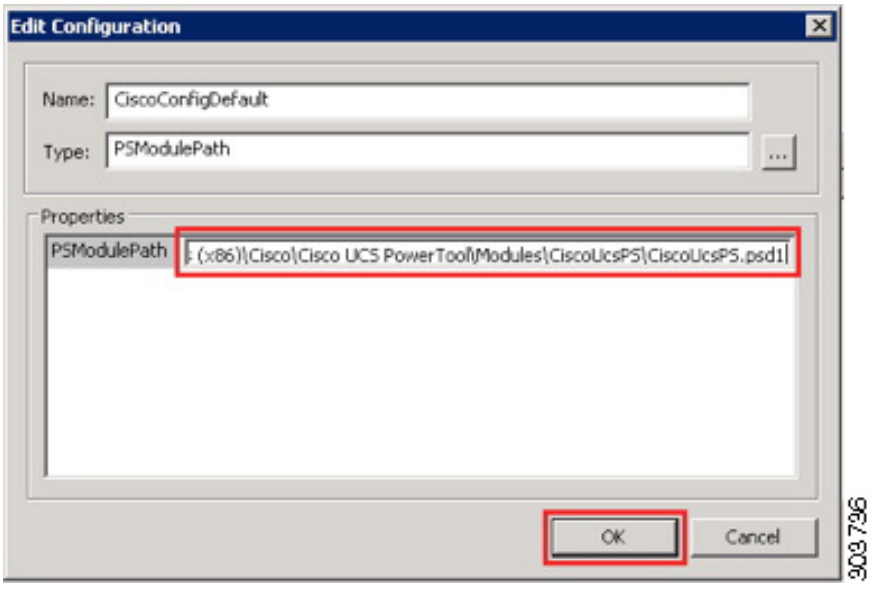
Step 5 In the **Item Selection** window, select **PSModulePath** and click **OK**.



Step 6 In the **Properties** field of the **Edit Configuration** window, specify the path for the Cisco UCS PowerTool module and click **OK**.

For example, you can enter a path such as: `C:\Program Files (x86)\Cisco\Cisco UCS PowerTool\Modules\CiscoUcsPS\CiscoUcsPS.psd1`).

You can leave the PSModulePath Properties field blank to use the default installation location.



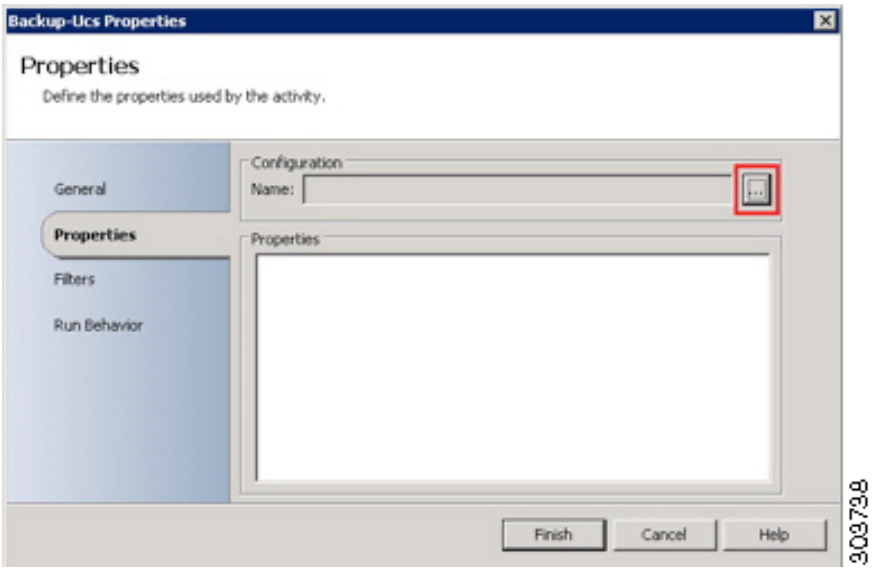
Step 7 In the **Prerequisite Configuration** window, click **Finish**.

Configuring Activity Properties

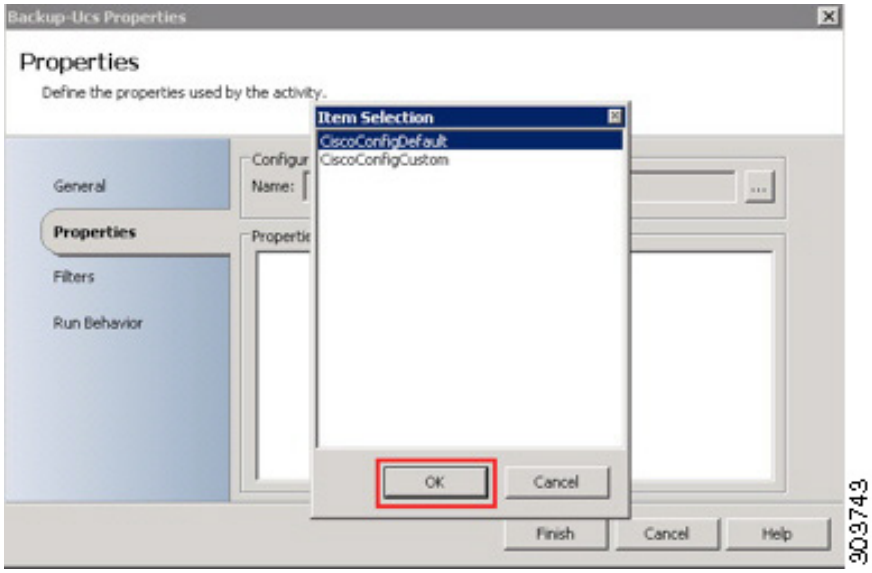
Each Cisco UCS activity has a set of required or optional configuration properties. The properties define the behavior of an activity. For example, the properties may include how to connect to other activities or how the activity performs its actions.

Step 1 1In the Runbook Designer window, double-click the activity.

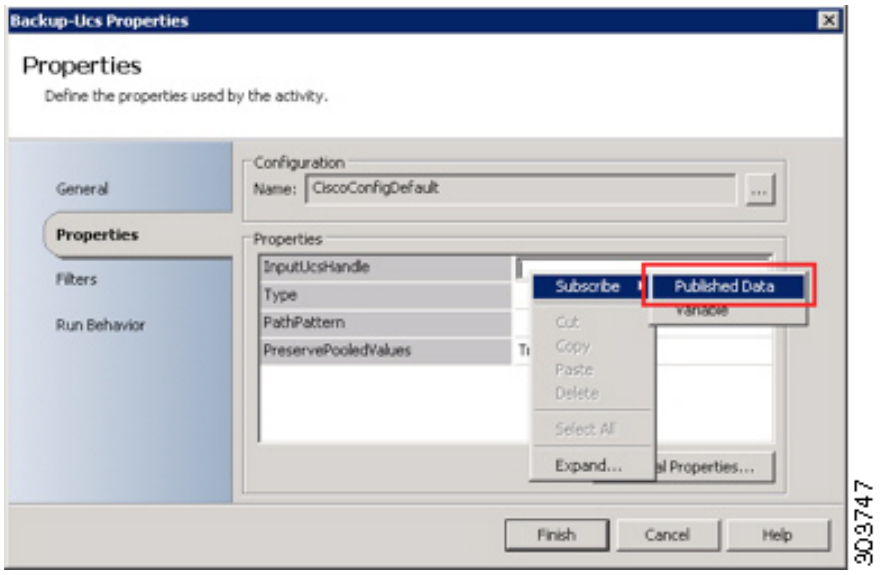
Step 2 In the **Backup-UCS Properties** window, click the Ellipsis (**...**) button.



Step 3 In the **Item Selection** window, choose the configuration and click **OK**.

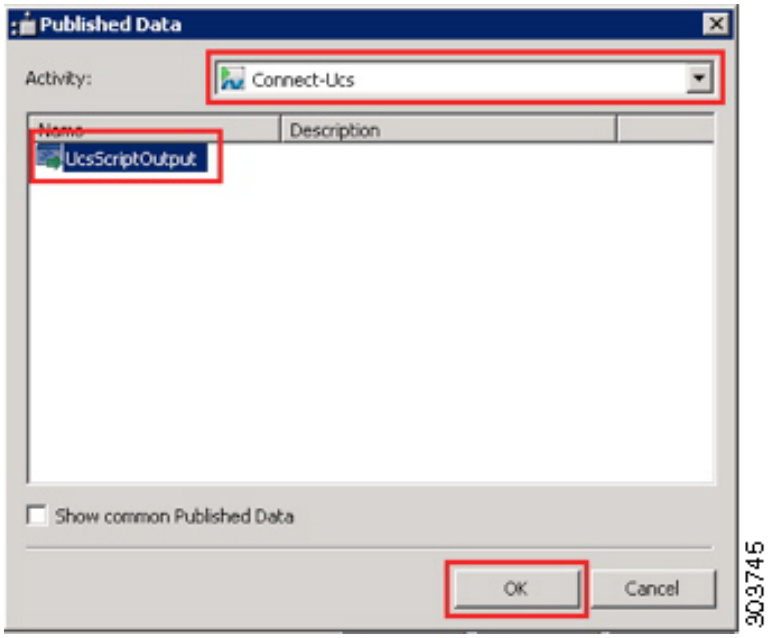


Step 4 In the **Properties** tab, right-click the **Input** field and choose **Subscribe > Published Data**.

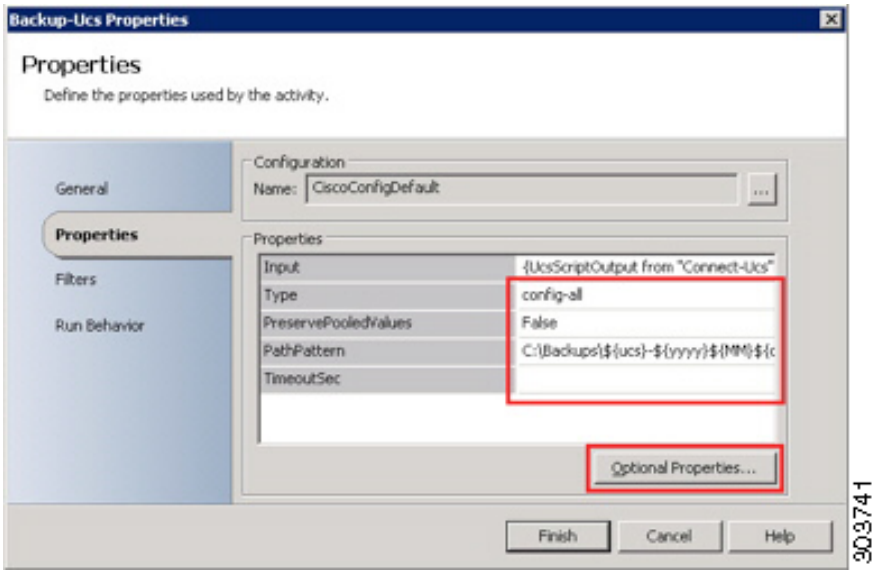


Step 5 From the **Activity** drop-down menu in the **Published Data** window, choose the activity whose output data you wish to subscribe to and click **OK**.

For example, select UcsScriptOutput.



Step 6 In the **Backup-UCS Properties** window, specify the other properties.



Step 7 Click **Optional Properties** to include any optional properties you wish to use.

Step 8 In the **Add/ Remove Property** window, do the following:

a. In the **Available** column, choose the properties you want to add.

b. Click the **>>** button to move the chosen properties to the **Selected** column.

c. Click **OK**.

Step 9 Click **Finish** to complete the activity configuration.

Cisco UCS Activities

The Cisco UCS Integration Pack activities described in this chapter are located in the Activity pane of the Runbook Designer.

Note All parameters in square brackets [ ] are optional parameters.

This chapter includes the following sections:

Connect-Ucs

Definition

This activity establishes a new connection with the Cisco UCS domains. This activity is required in a runbook if you want to perform an operation on one or more Cisco UCS domains.

Parameter Set

HostName <String>, Username <String>, Password <String>, [NoSsl <Boolean>], [Port <UInt16>]

Parameters

Table 4-1 provides details of the parameters available for this activity.

Table 4-1 Parameter Definitions for Connect-Ucs

Field	Description	Sample Values and Notes
Hostname	Host name/IP Address of Cisco UCS Manager	ucs-manager-01 209.165.201.3
Username	Valid login ID for Cisco UCS Manager	admin
Password	Password for the username	passw0rd
NoSsl	Use if you want to connect to Cisco UCS Manager with a non-secure (HTTP) connection	Default value: False Press the ellipsis (**...**) button to view the acceptable values.
Port	Port number for connections to Cisco UCS Manager	445 for HTTPS connections 88 for HTTP connections

Published Data

<UcsHandle>

After successful execution, this activity publishes the UcsHandle object and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Disconnect-Ucs

Definition

This activity disconnects from one or more Cisco UCS domains. This activity disconnects all connections to Cisco UCS (UcsHandle connections created by Connect-Ucs) in a runbook.

You must specify the name of the Cisco UCS domain if you want to only disconnect from one domain. If you do not specify the name of the Cisco UCS domain from which you want to disconnect, all UcsHandles are disconnected.

Parameter Set

[Ucs<String>]

Parameters

Table 4-2 provides details of the parameters available for this activity.

Table 4-2 Parameter Definitions for Disconnect-Ucs

Field	Description	Sample Values and Notes
Ucs	Comma separated names of Cisco UCS domains from which you want to disconnect. If you leave this value empty, this activity disconnects all handles created using Connect-Ucs in a runbook.	ucs-manager-01 You can find the name of the Cisco UCS domain in the published output of the Connect-Ucs activity.

Published Data

This activity does not publish output.

Get-UcsManagedObject

Definition

This activity is used to get one or more managed objects from a Cisco UCS domain. The output of this activity can be fed to other activities, such as Set-UcsManagedObject or Add-UcsManagedObject, to perform operations on that managed object. This activity supports filters that you can use to refine your search.

Parameter Set

Input <UcsHandle>, Dn <String>, [Hierarchy <Boolean>]

Input < UcsHandle >, ClassId <String>, [Filter <String>]

Input < UcsHandle >, XmlTag <String>, [Filter <String>]

Input <ManagedObject>, ClassId <String>, [Filter <String>]

Input <ManagedObject>, XmlTag <String>, [Filter <String>]

Parameters

Table 4-3 provides details of the parameters available for this activity.

Table 4-3 Parameter Definitions for Get-UcsManagedObject

Field	Description	Sample Values and Notes
Input	Managed Object (Parent) or UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle or Managed Object. A value of type UcsHandle gets the managed object according to the Dn, ClassId, or XmlTag parameters. A value of type Managed Object is considered to be the parent object. This gets the managed object under that subscribed parent object, according to the parameters ClassId or XmlTag.	{UcsScriptOutput from Connect-Ucs} {UcsScriptOutput from Get-UcsManagedObject} For example, you can subscribe from a Get-UcsManagedObject activity, which publishes a list of organizations, and also specify a value of lsServer for the ClassId parameter. With those values, this activity gets all service profiles under the subscribed organization list.
Dn	Distinguished name of the managed object.	org-root/org-subOrg/ls-finSrvcProf sys/chassis-1
Hierarchy	Includes children of the managed object as well as the object itself.	Default value: False True
ClassId	Class ID of the managed object	lsServer LsServer ComputeBlade
Filter	Filter string that you want to use to refine the search. **Note** Use a PowerTool compatible formatted filter and PowerTool compatible formatted syntaxes and wild cards in the filter strings.	(type -clike *-template) (name –ilike finSrvcProf)
XmlTag	Class ID (case sensitive) of the managed object	lsServer computeBlade

Published Data

<ManagedObject> or list<ManagedObject>

After successful execution, this activity publishes the managed object or the list of managed objects and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Set-UcsManagedObject

Definition

This activity is used to modify and configure managed objects. The property map can include values that modify more than one property of the managed object. You can use those modified.values to configure the managed object.

Parameter Set

Input <ManagedObject>, PropertyMap <KeyValuePair>

Input list<ManagedObject>, PropertyMap <KeyValuePair>

Input <UcsHandle>, ClassId <String>, PropertyMap <KeyValuePair>

Input <UcsHandle>, XmlTag <String>, PropertyMap <KeyValuePair>

Parameters

Table 4-4 provides details of the parameters available for this activity.

Table 4-4 Parameter Definitions of Set-UcsManagedObject

Field	Description	Sample Values and Notes
Input	Managed Object(s) or UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle or Managed Object. If you use a managed object or list of managed objects, such as a list of organizations, the property map includes the modifications that you wants to perform on the managed object or all managed objects in the list. If you use an input of type UcsHandle, the property map includes the key-value (modifications) for that UcsHandle.	{UcsScriptOutput from Connect-Ucs} {UcsScriptOutput from Get-UcsManagedObject}

Table 4-4 Parameter Definitions of Set-UcsManagedObject (continued)

Field	Description	Sample Values and Notes
PropertyMap	PropertyMap contains one or more key-value pairs separated by semi-colons. For example, a property map could be: propertyName1="propertyValue1";PropertyName2="propertyValue2"… **Note** If the input is of type UcsHandle, the property map must include the Dn for that UcsHandle. Property names are case insensitive. However, these names must match the properties of the managed object. To find the names for a managed object's properties, see the XML schema or the output of the get cmdlet in PowerTool for that managed object.	Dn="sys/chassis-1";Descr="modify"
ClassId	Class ID of the managed object	lsServer equipmentChassis
XmlTag	Class ID (case sensitive) of the managed object	lsServer orgOrg

Published Data

<ManagedObject> or list<ManagedObject>

After successful execution, this activity publishes the managed object or the list of managed objects and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Add-UcsManagedObject

Definition

This activity is used to add one or more managed objects. To create a managed object, you must provide the classId (type of managed object) and a property map that sets the values of the properties of that managed object.

Parameter Set

Input <UcsHandle>, ClassId <String>, PropertyMap <KeyValuePair>, [ModifyPresent <Boolean>]

Input <UcsHandle>, XmlTag <String>, PropertyMap <KeyValuePair>, [ModifyPresent <Boolean>]

Input <ManagedObject>, ClassId <String>, PropertyMap <KeyValuePair>, [ModifyPresent <Boolean>]

Input <ManagedObject>, XmlTag <String>, PropertyMap <KeyValuePair>, [ModifyPresent <Boolean>]

Input list<ManagedObject>, ClassId <String>, PropertyMap <KeyValuePair>, [ModifyPresent <Boolean>]

Input list<ManagedObject>, XmlTag <String>, PropertyMap <KeyValuePair>, [ModifyPresent <Boolean>]

Parameters

Table 4-5 provides details of the parameters available for this activity.

Table 4-5 Parameter Definitions of Add-UcsManagedObject

Field	Description	Sample Values and Notes
Input	Managed Object(s) or UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle or Managed Object. If you use a managed object or list of managed objects (parent), a new managed object of the specified classId/XmlTag is created under the subscribed managed object(s). If you use an input of type UcsHandle, a new managed object with the specified classId/XmlTag is created using the subscribed UcsHandle.	{UcsScriptOutput from Connect-Ucs} {UcsScriptOutput from Get-UcsManagedObject} For example, if you use a list of organizations as a parent managed object, you can create service profiles under that subscribed list of organizations.
ClassId	Class ID of the managed object	lsServer orgOrg
PropertyMap	PropertyMap contains one or more key-value pairs separated by semi-colons. These keys and values denote the properties and their values for the managed object that you want to create. For example, a property map could be: propertyName1="propertyValue1";PropertyName2="propertyValue2"… **Note** If the input is of type UcsHandle, the property map must include the Dn for that UcsHandle.	Dn="org-root/org-subOrg";Descr="modify"
ModifyPresent	If the managed object already exists and this parameter is set to True, the activity modifies the existing managed object.	True False Default Value: False
XmlTag	Class ID (case sensitive) of the managed object	lsServer orgOrg

Published Data

<ManagedObject> or list<ManagedObject>

After successful execution, this activity publishes the managed object or the list of managed objects and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Remove-UcsManagedObject

Definition

This activity is used to remove one or more managed objects. You can identify the managed object to be removed by providing the managed object or its Dn and classId.

Parameter Set

Input <ManagedObject>

Input list<ManagedObject>

Input <UcsHandle>, ClassId <String>, Dn <String>

Input <UcsHandle>, XmlTag <String>, Dn <String>

Parameters

Table 4-6 provides details of the parameters available for this activity.

Table 4-6 Parameter Definitions of Remove-UcsManagedObject

Field	Description	Sample Values and Notes
Input	Managed Object(s) or UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle or Managed Object. If you use a managed object or list of managed objects, this activity removes those managed object(s). If you use an input of type UcsHandle, the specified Dn and classId determines which managed object is removed.	{UcsScriptOutput from Connect-Ucs} {UcsScriptOutput from Get-UcsManagedObject} For example, if you use a list of organizations as a parent managed object, you can create service profiles under that subscribed list of organizations.
ClassId	Class ID of the managed object	orgOrg fabricVlan
Dn	Distinguished name of the managed object.	org-root/org-Finance sys/chassis-3
XmlTag	Class ID (case sensitive) of the managed object	orgOrg fabricVlan

Published Data

<ManagedObject> or list<ManagedObject>

After successful execution, this activity publishes the managed object or the list of managed objects and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Run-PowerTool

Definition

This activity is used to run a PowerTool script. You can subscribe a UcsHandle from the Connect-Ucs activity and use that handle in the PowerTool script. You can also connect or disconnect the UcsHandle in the script itself. The output of this activity can be passed to another activity if you want to perform additional operations.

Parameter Set

Input<object>, Script<String>

Parameters

Table 4-7 provides details of the parameters available for this activity.

Table 4-7 Parameter Definitions of Run-PowerTool

Field	Description	Sample Values and Notes
Input	Managed Object(s) or UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle or Managed Object. You can access the subscribed object/handle in the Script parameter directly using special '$ucsInput' variable. For more information, see the "Run Custom PowerTool Script" example in the Sample Runbooks. This parameter is optional. If you do not want to subscribe from any activity, leave this parameter empty. For example, if you do not want to subscribe from the Connect-Ucs activity, but instead you want to create a UcsHandle inside the PowerTool script, leave this parameter empty.	{UcsScriptOutput from Connect-Ucs} {UcsScriptOutput from Get-UcsManagedObject}

Table 4-7 Parameter Definitions of Run-PowerTool (continued)

Field	Description	Sample Values and Notes
Script	PowerTool script to run. You can paste an existing PowerTool script in this parameter and execute that script. **Note** In the PowerTool script, '$ucsInput' contains the input object(s) from a subscribed activity. Add any output object(s) to be published in '$ucsOutput', such as $ucsOutput.Add($obj).	orgOrg fabricVlan

Published Data

<ManagedObject> or list<ManagedObject>

After successful execution, this activity publishes the managed object or the list of managed objects and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Copy-UcsServiceProfile

Definition

This activity is used to make a copy of an existing service profile or service profile template.

Parameter Set

Input <LsServer>, NewName <String>, [DestinationOrgDn <String>], [Type <String>]

Parameters

Table 4-8 provides details of the parameters available for this activity.

Table 4-8 Parameter Definitions of Copy-UcsServiceProfile

Field	Description	Sample Values and Notes
Input	Service Profile or Service Profile Template Managed object of type lsServer. You can subscribe from an activity which publishes an output of type lsServer managed object.	{UcsScriptOutput from Get-UcsManagedObject}
NewName	Name of the service profile copy (new service profile name)	orgOrg fabricVlan

Table 4-8 Parameter Definitions of Copy-UcsServiceProfile (continued)

Field	Description	Sample Values and Notes
DestinationOrgDn	Distinguished name of the organization where the service profile copy is to be created. **Note** If a DestinationOrgDn is not provided, the service profile copy is created under the root organization.	org-root\org-Finance
Type	Type of service profile (copy). To choose a type, click on the ellipsis (**...**) button in the text box. You can choose one of the following types: • instance—The new copy will be a service profile. • initial-template—The new copy will be a service profile template of type initial. • updating-template—The new copy will be a service profile template of type updating. If you do not include a value for the type parameter, the type of service profile that is created depends on the Input parameter. If you subscribe a service profile template, the type will be initial-template. If you subscribe a service profile, the type will be instance.	

Published Data

<ManagedObject> or list<ManagedObject> (of type lsServer)

After successful execution, this activity publishes the managed object or the list of managed objects of type lsServer, and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Add-UcsServiceProfileFromTemplate

Definition

This activity is used to create one or more service profiles from a service profile template. To name the service profiles, you can either provide comma-separated names or “name prefix, count” values.

Parameter Set

InputManagedObject< ManagedObject >, NewName <String>, [DestinationOrgDn<String>]

InputManagedObject< ManagedObject >, Prefix <String>, Count <UInt32>, [DestinationOrgDn <String>]

Parameters

Table 4-9 provides details of the parameters available for this activity.

Table 4-9 Parameter Definitions of Add-UcsServiceProfileFromTemplate

Field	Description	Sample Values and Notes
InputManaged Object	Managed Object(s) of type service profile template. Managed object of type lsServer. You can subscribe from an activity which publishes an output of type lsServer (service profile template) managed objects. You can use filters to get one or more managed objects of type service profile templates. For more information, see the "Add Service Profile From Template" runbook in the Sample Runbooks.	{UcsScriptOutput from Get-UcsManagedObject}
NewName	Comma-separated names of the new service profiles that you want to create from the service profile template.	financeServiceProfile, eduServiceProfile
Prefix	Prefix for the service profile names.	finSP
Count	Number of service profiles that you want to create from the service profile template.	5
DestinationOrgDn	Distinguished name of the organization where the service profile copy is to be created. **Note** If a DestinationOrgDn is not provided, the service profile copy is created under the root organization.	org-root/org-fin org-root/org-fin/org-account

Published Data

<ManagedObject> or list<ManagedObject>

After successful execution, this activity publishes the managed object or the list of managed objects (service profile) and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Get-UcsTechSupport

Definition

This activity is used to create and download a tech support file for a Cisco UCS domain. If desired, you can delete the tech support file from Cisco UCS Manager after you downloading the file. You can also choose various options for downloading the tech support file.

Parameter Set

Input<UcsHandle>, PathPattern<String>, UcsManager<Boolean>, [RemoveFromUcs<Boolean>], [TimeOutSec<UInt32>]

Input<UcsHandle>, PathPattern<String>, ChassisId<UInt16>, [CimcId<String>], [IomId<String>], [RemoveFromUcs<Boolean>], [TimeOutSec<UInt32>]

Input<UcsHandle>, PathPattern<String>, FexId<UInt16>, [RemoveFromUcs<Boolean>], [TimeOutSec<UInt32>]

Input<UcsHandle>, PathPattern<String>, RackServerId<UInt16>, [RackAdapterId<UInt16>], [RemoveFromUcs<Boolean>], [TimeOutSec<UInt32>]

Parameters

Table 4-10 provides details of the parameters available for this activity.

Table 4-10 Parameter Definitions of Get-UcsTechSupport

Field	Description	Sample Values and Notes
Input	UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle.	{UcsScriptOutput from Connect-Ucs}
PathPattern	Full name (including location) of the file you want to create. **Note** The path may contain DateTime format specifiers, as well as a custom format specifier 'ucs' which relates to the Cisco UCS name. For example: `C:\Folder\${ucs}-${yyyy}${MM}${dd}-${HH}${mm}.tar`	C:/subfold/techSupp.tar
RemoveFromUcs	If this parameter is set to True, the tech support file is deleted from Cisco UCS Manager after it is downloaded. Use the ellipsis (**...**) to choose a value for this parameter.	True False Default Value: False
UcsManager	Creates a tech support file for the entire Cisco UCS domain. Use the ellipsis (**...**) to choose a value for this parameter.	
TimeOutSec	Timeout duration in seconds.	Default Value: 600 seconds
ChassisId	ID of the chassis for which you want to create a tech support file.	
CimcId	ID of the CIMC for which you want to create a tech support file.	
IomId	ID of the I/O module for which you want to create a tech support file.	

Table 4-10 Parameter Definitions of Get-UcsTechSupport (continued)

Field	Description	Sample Values and Notes
FexId	ID of the FEX for which you want to create a tech support file.	
RackServerId	ID of the rack-mount server for which you want to create a tech support file.	
RackAdapterId	ID of the adapter in the rack-mount server for which you want to create a tech support file.	

Published Data This activity does not publish output.

Backup-Ucs

Definition This activity is used to create and download a backup file from a Cisco UCS domain. The backup file can be a snapshot of all or part of the system configuration, and can be exported to a file.

Parameter Set Input<UcsHandle>, Type<String>, PathPattern<String>, [PreservePooledValues<Boolean>], [TimeOutSec<UInt32>]

Parameters Table 4-11 provides details of the parameters available for this activity.

Table 4-11 Parameter Definitions of Backup-Ucs

Field	Description	Sample Values and Notes
Input	UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle.	{UcsScriptOutput from Connect-Ucs}

Table 4-11 Parameter Definitions of Backup-Ucs (continued)

Field	Description	Sample Values and Notes
Type	The type of backup that you want to create. This can be one of the following types: • full-state—A binary file that includes a snapshot of the entire system. • config-all—An XML file that includes all system and logical configuration settings. • config-system—An XML file that includes all system configuration settings such as usernames, roles and locales. • config-logical—An XML file that includes all logical configuration settings such as service profiles, VLANs, VSANs, pools, and policies. Use the ellipsis (**...**) to choose a value for this parameter.	full-state
PathPattern	Full name (including location) of the backup file you want to create. **Note** The path may contain DateTime format specifiers, as well as a custom format specifier 'ucs' which relates to the Cisco UCS name. For example: `C:\Folder\${ucs}-${yyyy}${MM}${dd}-${HH}${mm}.tar`	C:/subfold/full-stateBackup.xml
PreservePooled Values	If this parameter is set to True, the backup preserves the values of vHBAs, WWPNs, vNICs, MAC addresses, WWNNs, and UUIDs that are derived from pools. Use the ellipsis (**...**) to choose a value for this parameter.	True False Default: False for backups of type full-state and config-system
TimeOutSec	Timeout duration in seconds.	Default Value: 600 seconds

Published Data

This activity does not publish output.

Import-UcsBackup

Definition This activity is used to import a Cisco UCS backup file.

Note You cannot import a full-state configuration file.

Parameter Set Input<UcsHandle>, LiteralPath<String>, [Merge<Boolean>]

Input<UcsHandle>, Path<String>, [Merge<Boolean>]

Parameters Table 4-12 provides details of the parameters available for this activity.

Table 4-12 Parameter Definitions of Import-UcsBackup

Field	Description	Sample Values and Notes
Input	UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle.	{UcsScriptOutput from Connect-Ucs}
LiteralPath	Exact path to the backup file that you want to import.	C:/subfold/All-ConfigBackup.xml
Merge	If this parameter is set to True, the activity merges the contents of the backup file into the existing configuration state.	True False Default: False
Path	Path to the backup file that you want to import, including wildcard characters.	C:\work\config?.xml

Published Data This activity does not publish output.

Get-UcsChild

Definition This activity is used to get the child objects of a managed object. If you subscribe a managed object in the InputManagedObject parameter, you get all the immediate children of that managed object.

Parameter Set InputManagedObject< ManagedObject >, [ClassId<String>], [Hierarchy<Boolean>]

Parameters

Table 4-13 provides details of the parameters available for this activity.

Table 4-13 Parameter Definitions of Get-UcsChild

Field	Description	Sample Values and Notes
Input	Managed Object You can use any value from a previous activity whose output (published data) is of type Managed Object.	{UcsScriptOutput from Get-UcsManagedObject}
classId	The classId of the child objects.	lsServer
Hierarchy	If this parameter is set to True, the activity includes the child objects at all levels of the hierarchy. Use the ellipsis (**...**) to choose a value for this parameter.	True False Default: False

Published Data

<ManagedObject> or list<ManagedObject>

After successful execution, this activity publishes the managed object or the list of managed objects (service profile) and makes it available for other activities to use through UcsScriptOutput. If an error occurs, an error message is populated in the default Error summary text field in the SCO.

Confirm-UcsFault

Definition

This activity is used to acknowledge faults.

Parameter Set

InputUcsHandle< UcsHandle >, Id<ulong>

Parameters

Table 4-14 provides details of the parameters available for this activity.

Table 4-14 Parameter Definitions of Confirm-UcsFault

Field	Description	Sample Values and Notes
InputUcsHandle	UcsHandle You can use any value from a previous activity whose output (published data) is of type UcsHandle.	{UcsScriptOutput from Connect-Ucs}
Id	The ID of the fault that you want to acknowledge.	3171250

Published Data

This activity does not publish output.

Sample Runbooks

To illustrate how to use the activities that are available as part of this integration pack, a set of sample runbooks is available on the Cisco Developer Network. You can import these sample runbooks into the SCO runbook designer and use it to understand how the activities can be used to publish data, use data, and execute an action on your Cisco UCS domains.

These sample runbooks provide examples of how to use the activities included in the Cisco UCS Integration Pack. For more information about the available activities, see Cisco UCS Activities, page 4-1.

This chapter includes the following sections:

Add Service Profile From Template

This sample runbook shows the steps required to create a new service profile from an existing service profile template.

Step 1 Initialize Data, which uses the following parameters:

- New Service Profile Name—The name of the service profile created by the successful execution of the runbook.
- Service Profile Template Name—The name of the service profile template used to create the service profile.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Get-UcsManagedObject(Org), which uses the following parameters to get the required org-root managed object:

- Input—Subscribes UcsHandle from the Connect-Ucs activity.
- Dn—Contains the distinguished name of the organization that you want to get.

Step 4 Get-UcsManagedObject(Service Profile), which uses the following parameters to get the service profile template managed object:

- Input—Subscribes organization from the previous activity.
- ClassId—Contains the class ID of the service profile.
- Filter—Specifies that the managed object should be of type template. The name of the managed object must be identical to the name you provided in the Initialize Data activity.

Step 5 Add-UcsServiceProfileFromTemplate, which use the following parameters to create the service profile according to the provided template:

- InputManagedObject—Subscribes from the previous activity which publishes the service profile template.
- NewName—Subscribes the New Service Profile Name variable from the Initialize Data activity.

Step 6 Disconnect-Ucs, which disconnects the UcsHandle.

Add VLAN

This sample runbook shows the steps required to add a VLAN to a fabric interconnect.

Step 1 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 2 Get-UcsManagedObject(Lan Cloud), which uses the following parameters to get the FabricLanCloud managed object:

- Input—Subscribes UcsHandle from the Connect-Ucs activity.
- Dn—Contains the distinguished name of the LAN cloud that you want to get.

Step 3 Add-UcsManagedObject, which uses the following parameters to create a VLAN under the fabricLanCloud managed object:

- InputManagedObject—Subscribes from the previous activity which publishes the fabricLanCloud object.
- PropertyMap—Specifies the necessary properties for creating a VLAN, such as the VLAN ID and name.
- ClassId—Specifies the type of object that you want to create, which is fabricVLAN.

Step 4 Disconnect-Ucs, which disconnects the UcsHandle.

Backup UCS

This sample runbook shows the steps required to create and download a backup file for Cisco UCS Manager.

Step 1 Initialize Data, which uses the following parameters:

- PathPattern—The name and full filepath (location) where you want to create the backup on local machine.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Backup-Ucs, which uses the following parameters to create and download the backup file:

- InputUcsHandle—Subscribes UcsHandle from the previous activity.
- Type—Specifies the type of backup, such as config-system.
- PathPattern—Subscribes the PathPattern variable from the Initialize Data activity.

Step 4 Disconnect-Ucs, which disconnects the UcsHandle.

Confirm Fault

This sample runbook shows the steps required to acknowledge a fault using the fault ID.

Step 1 Initialize Data, which uses the following parameters:

- Id—The ID of the fault you want to acknowledge.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Confirm-UcsFault, which uses the following parameters to acknowledge the fault whose ID is provided in Initialize Data:

- InputUcsHandle—Subscribes UcsHandle from the previous activity.
- Id—Subscribes the Id variable with the fault ID from the Initialize Data activity.

Step 4 Disconnect-Ucs, which disconnects the UcsHandle.

Copy Service Profile

This sample runbook shows the steps required to make a copy of an existing service profile.

Step 1 Initialize Data, which uses the following parameters:

- ServiceProfileDn—The distinguished name of the service profile that you want to copy.
- DestinationOrgDn—The distinguished name of the organization where the service profile copy is to be created.
- NewName—The name to be given to the service profile copy.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Get-UcsManagedObject (Service Profile), which uses the following parameter to get the service profile that you want to copy:

- Input—Subscribes UcsHandle from the Connect-Ucs activity.
- ServiceProfileDn—Subscribes this variable with the distinguished name of the service profile from the Initialize Data activity.

Step 4 Copy-UcsServiceProfile, which uses the following parameters to make a copy of an existing service profile:

- InputManagedObject—Subscribes the service profile from the previous activity.
- NewName—Specifies the name to be given to the new service profile.
- DestinationOrgDn—Specifies the distinguished name of the organization under which the new service profile will be created.

Step 5 Disconnect-Ucs, which disconnects the UcsHandle.

Get Child

This sample runbook shows the steps required to get one or more child objects of a managed object.

Step 1 Initialize Data, which uses the following parameter:

- OrgDn—The distinguished name of the managed object from which you want to get one or more child objects.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Get-UcsManagedObject(Org), which uses the following parameters to get the organization from which you want to get one or more child objects:

- Input—Subscribes UcsHandle from the Connect-Ucs activity.
- OrgDn—Subscribes this variable with the managed object's distinguished name from the Initialize Data activity.

Step 4 Get-UcsChild, which uses the following parameters to get one or more child objects of a managed object:

- InputManagedObject—Subscribes the organization from the previous activity.
- ClassId—Specifies the type of child object that you want to get from the subscribed managed object.

Step 5 Disconnect-Ucs, which disconnects the UcsHandle.

Get Tech Support

This sample runbook shows the steps required to create and download a tech support file.

Step 1 Initialize Data, which uses the following parameters:

- PathPattern—The full name and filepath (location) of the tech support file you want to create.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Get-UcsTechSupport, which uses the following parameters to create and download a tech support file for a Cisco UCS domain:

- InputUcsHandle—Subscribes UcsHandle from the previous activity.
- PathPattern—Subscribes the PathPattern variable from the Initialize Data activity to specify the name and file path for the tech support file.
- RemoveFromUcs—Specifies whether the file should be deleted after it is downloaded. Set to True to delete the file.
- UcsManager—Specifies whether the tech support file should include all components in the Cisco UCS domain. Set to True for a tech support file that includes all components.

Step 4 Disconnect-Ucs, which disconnects the UcsHandle.

Import Backup

This sample runbook shows the steps required to import a Cisco UCS backup file.

Step 1 Initialize Data, which uses the following parameters:

- LiteralPath—The exact name and filepath (location) of the backup file you want to import.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Import-UcsBackup, which uses the following parameters to import a Cisco UCS backup file:

- InputUcsHandle—Subscribes UcsHandle from the previous activity.
- LiteralPath—Subscribes the LiteralPath variable from the Initialize Data activity to specify the name and file path for the backup file.

Step 4 Disconnect-Ucs, which disconnects the UcsHandle.

Modify Service Profile

This sample runbook shows the steps required to modify an existing service profile. This runbook gets the service profile from an organization and then modifies it.

Step 1 Initialize Data, which uses the following parameters:

- Service Profile Name—The name of the service profile that you want to modify.
- Modify PropertyMap—The property map with the key-value pairs that you want to modify.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Get-UcsManagedObject(Org), which uses the following parameters to get the required org-root managed object:

- Input—Subscribes UcsHandle from the Connect-Ucs activity.
- Dn—Contains the distinguished name of the organization that you want to get.

Step 4 Get-UcsManagedObject(Service Profile), which uses the following parameters to get the service profile managed object:

- Input—Subscribes organization from the previous activity.
- ClassId—Contains the class ID of the service profile.
- Filter—Specifies that the managed object should be of type instance. The name of the managed object must be identical to the name you provided in the Initialize Data activity.

Step 5 Set-UcsManagedObject, which uses the following parameters to modify the service profile:

- Input—Subscribes the service profile that you want to modify from the previous activity.
- PropertyMap—Subscribes the Modify PropertyMap variable from the Initialize Data activity, which contains the changes that you want to make to the service profile.

Step 6 Disconnect-Ucs, which disconnects the UcsHandle.

Remove Service Profile

This sample runbook shows the steps required to delete an existing service profile. This runbook gets the service profile according its DN (distinguished name) and then removes it.

Step 1 Initialize Data, which uses the following parameter:

- Service Profile DN—The distinguished name of the service profile that you want to remove.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Get-UcsManagedObject(Service Profile), which uses the following parameters to get the service profile managed object:

- Input—Subscribes UcsHandle from the Connect-Ucs activity.
- Dn—Specifies the distinguished name of the service profile you want to remove.

Step 4 Remove-UcsManagedObject, which uses the following parameter to delete the service profile:

- Input—Subscribes the service profile that you want to delete from the previous activity.

Step 5 Disconnect-Ucs, which disconnects the UcsHandle.

Run Custom PowerTool Script

This sample runbook shows the steps required to use the Run-PowerTool activity to get the value of the user label of a service profile and set that as the user label for all server objects. This runbook uses the following three instances of the Run-PowerTool activity:

- The first instance fetches the service profile that has the given name in the specified Cisco UCS domain.
- The second instance fetches all the server objects in the specified Cisco UCS domain.
- The third instance copies the value of the user label from the fetched service profile to the user label of the fetched server managed objects.

This sample runbook is intended to help you understand how data from the PowerTool script of one Run-PowerTool activity instance can be passed and used in the script of another Run-PowerTool activity instance.

Step 1 Initialize Data, which uses the following parameters:

- ServiceProfileName—The name of the service profile whose user label you want to copy to the user label property of the server objects.
- HostName—The hostname of the Cisco UCS domain.
- UserName—A valid username with the required privileges that can be used to access the Cisco UCS domain.
- Password—The password for the username.

This is a standard SCO activity that is used to initialize input parameters and define runtime variables. When the runbook is executed, you are asked to enter values for these input parameters. Other activities which need these values get them by referencing these input parameters.

Step 2 Connect-Ucs, which uses the following parameters to create UcsHandle:

- hostname
- username
- password

This activity uses the UcsScriptOutput variable to create UcsHandle for other activities to use.

Step 3 Run-PowerTool (Get a Service Profile), which uses the following parameter to get the service profile managed object.

- Input—Subscribes the UcsHandle from the previous activity, which will be deserialized and available in the PowerTool script (the script parameter of this activity) as $ucsInput.
- Script—Contains the PowerTool script that gets the service profile and passes it to SCO for publishing. The script shows the following special parameters: $ucsInput and $ucsOutput.

Step 4 Run-PowerTool (Get all Server Objects), which uses the following parameters to get the server objects:

- Input—Subscribes UcsHandle from the Connect-Ucs activity.
- Script—Contains the PowerTool script that gets the server objects and passes them to SCO for publishing.

Step 5 Run-PowerTool (Set Usr Label of Blade to Usr Label of SP), which uses the following parameters to modify the user label of the server objects:

- Input—Subscribes server objects from the previous activity.
- Script—Contains the PowerTool script that modifies the server objects and passes them to SCO for publishing.

This script also shows how to subscribe from another activity, such as from Run-PowerTool (Get a Service Profile) and how to recreate data from the subscribed value.

Step 6 Disconnect-Ucs, which disconnects the UcsHandle.