



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, page 5-1](#)
- [Add VLAN, page 5-2](#)
- [Backup UCS, page 5-3](#)
- [Confirm Fault, page 5-4](#)
- [Copy Service Profile, page 5-4](#)
- [Get Child, page 5-5](#)
- [Get Tech Support, page 5-6](#)
- [Import Backup, page 5-6](#)
- [Modify Service Profile, page 5-7](#)
- [Remove Service Profile, page 5-8](#)
- [Run Custom PowerTool Script, page 5-8](#)

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 to 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.
-

