



## **Cisco UCS Director Orchestration Guide, Release 6.8**

**First Published:** 2021-03-17

**Last Modified:** 2022-03-14

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <https://www.cisco.com/c/en/us/about/legal/trademarks.html>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2021–2022 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### PREFACE

<b>Preface</b>	<b>ix</b>
Audience	<b>ix</b>
Conventions	<b>ix</b>
Related Documentation	<b>xi</b>
Documentation Feedback	<b>xi</b>
Communications, Services, and Additional Information	<b>xi</b>

---

### CHAPTER 1

<b>New and Changed Information</b>	<b>1</b>
New and Changed Information for This Release	<b>1</b>

---

### CHAPTER 2

<b>Overview</b>	<b>3</b>
Cisco UCS Director Orchestrator	<b>3</b>
Going Further	<b>3</b>
Common Icons	<b>4</b>

---

### CHAPTER 3

<b>Orchestration Concepts</b>	<b>7</b>
Introduction to UCS Director Orchestration	<b>7</b>
Tasks	<b>7</b>
Workflows	<b>8</b>
Service Requests	<b>8</b>
Input and Output	<b>8</b>
Workflow Validation	<b>9</b>
Workflow Versioning	<b>10</b>
Approvals	<b>10</b>
Rollback	<b>10</b>
Libraries and Catalogs	<b>11</b>

Activities 11  
 Managing the Workflow Lifecycle 12

---

**CHAPTER 4**

**Creating Workflows 15**

Viewing the Task Library 15  
 The Workflow Designer 15  
 Creating a Workflow 17  
     Defining a Workflow 17  
     Defining Workflow Input and Output 20  
     Dynamic Workflow Inputs 22  
         Adding an Input Rule 24  
         Example: Using Basic Workflow Input Processing Rules 25  
         Example: Using Cascading Workflow Input Processing Rules 26  
         Example: Using More Advanced Input Processing Rules 27  
         Example: Combining Advanced Input Processing Rules 29  
     Admin Input Filters 31  
     Configuring a Task in a Workflow 33  
     Connecting a Task to a Workflow 36  
     Cloning a Task 37  
     Validating a Workflow 38  
 Example: Creating a Workflow 39  
 Resolving Workflow Validation Issues 41  
 Changing Flow of Control in Workflows 42  
     Looping in Workflows 42  
     Example: Looping in a Workflow 44  
     Adding an If-Then Branch to a Workflow 46  
     Using Conditional Branching in a Workflow 47  
     Creating Approvals 49  
 Locking Resources 51  
 File Uploads 53  
     Permanently Uploading an Admin-Selected File 53  
     Temporarily Uploading a User-Selected File 54

---

**CHAPTER 5**

**Executing Workflows 57**

Workflow Execution	57
Pre- and Post-Provisioning Workflows	58
Executing a Workflow	58
Example: Executing a Workflow	59
Example: Updating the Default Template for Sending Email through Template	60
Creating a Workflow Trigger	61
Example: Creating a Trigger to Execute a Workflow	63
Creating a Trigger Template	66
Scheduling a Service Request	68
Example: Scheduling Workflow Execution	69
Creating a Context Workflow Mapping	70
Using VM Action Policies	72

---

**CHAPTER 6**
**Using Service Requests 75**

Viewing Service Requests	75
Viewing Service Request Input and Output	76
Rolling Back a Service Request	76
Resubmitting a Service Request	77
Approving and Denying Service Requests	78
Deleting Service Requests	79
Locking a Service Request	80

---

**CHAPTER 7**
**Managing Workflows 81**

Workflow Editing	81
Renaming Workflows	82
Reordering Inputs	82
Deleting Workflow Inputs	83
Renaming Workflow Inputs	83
Changing Input Optional or Required Status	84
Editing a Task in an Existing Workflow	85
Opening Workflow Designer from a Header Search	86
Deleting a Workflow	87
Deleting Multiple Workflows	87
Running a Quick Export on Workflows	88

- Exporting and Importing Cisco UCS Director Artifacts **89**
  - Exporting Workflows, Custom Tasks, Script Modules, and Activities **89**
  - Importing Workflows, Custom Tasks, Script Modules, and Activities **91**
  - Workflow Templates **92**
    - Exporting Workflows as Templates **92**
    - Importing a Workflow Template **93**
    - Creating a Workflow from a Template **93**
    - Predefined Templates **94**
  - Workflow Version History **95**
    - Creating a New Version of a Workflow **95**
    - Choosing the Default Version of a Workflow **96**
    - Managing Versions of a Compound Task **97**
  - Cloning a Workflow **98**
  - Searching a Workflow **99**
  - Saving a Picture of a Workflow **100**

---

**CHAPTER 8**

- Customizing Workflow Components 101**
  - Creating a Compound Task **101**
    - Example: Creating a Compound Task **102**
  - Creating Custom Approvals **102**
  - Creating Custom Inputs **103**
  - Generating a Task Using a OpenAPI Specification File **104**
  - Orchestration Global Variables **106**
    - Orchestration Variables **107**
      - Input and Output System Variables **107**
      - Service Request Global Variables **110**
      - Virtual Machine System Variables **110**
    - User-Defined Global Variables **110**
      - Creating Global Variables **110**
      - Cloning Global Variables **111**
      - Editing Global Variables **112**
      - Deleting Global Variables **112**
  - VM Annotations **113**
    - List of VM Macros and VM Annotations **113**

---

**CHAPTER 9****Using Activities 117**

- Activities 117
- Creating an Activity 117
- Associating an Activity with a Workflow 118
- Adding an Activity to a Workflow 120
- Importing and Exporting Activities 121
- Deleting an Activity 121

---

**CHAPTER 10****Using Script Modules 123**

- Using Script Modules 123
- Adding Script Modules 124
- Adding Libraries 124
  - Accessing Libraries 125
- Jar Files 125
  - Adding Jar Files 125
- Lists of Values 126
  - Adding a List of Values 126
  - Editing a List of Values 127
  - Deleting a List of Values 128
- Tabular Reports 128
  - Adding a Tabular Report 128
  - Editing a Tabular Report 131
  - Deleting a Tabular Report 131
- Context Mapping 131
  - Enabling Metadata 132
  - Adding a Context Mapping 132
  - Editing a Context Mapping 134
  - Deleting a Context Mapping 135
- Importing and Exporting Script Modules 135





## Preface

---

- [Audience, on page ix](#)
- [Conventions, on page ix](#)
- [Related Documentation, on page xi](#)
- [Documentation Feedback, on page xi](#)
- [Communications, Services, and Additional Information, on page xi](#)

## Audience

This guide is intended primarily for data center administrators who use Cisco UCS Director and who have responsibilities and expertise in one or more of the following:

- Server administration
- Storage administration
- Network administration
- Network security
- Virtualization and virtual machines

## Conventions

Text Type	Indication
GUI elements	GUI elements such as tab titles, area names, and field labels appear in <b>this font</b> . Main titles such as window, dialog box, and wizard titles appear in <b>this font</b> .
Document titles	Document titles appear in <i>this font</i> .
TUI elements	In a Text-based User Interface, text the system displays appears in <i>this font</i> .
System output	Terminal sessions and information that the system displays appear in <i>this font</i> .

Text Type	Indication
CLI commands	CLI command keywords appear in <b>this font</b> . Variables in a CLI command appear in <i>this font</i> .
[ ]	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.
< >	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 document.

---




---

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

---




---

**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.

---




---

**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

---

## Related Documentation

### Cisco UCS Director Documentation Roadmap

For a complete list of Cisco UCS Director documentation, see the *Cisco UCS Director Documentation Roadmap* available at the following URL: [http://www.cisco.com/en/US/docs/unified\\_computing/ucs/ucs-director/doc-roadmap/b\\_UCSDirectorDocRoadmap.html](http://www.cisco.com/en/US/docs/unified_computing/ucs/ucs-director/doc-roadmap/b_UCSDirectorDocRoadmap.html).

### 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>.

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



---

**Note** The *Cisco UCS B-Series Servers Documentation Roadmap* includes links to documentation for Cisco UCS Manager and Cisco UCS Central. The *Cisco UCS C-Series Servers Documentation Roadmap* includes links to documentation for Cisco Integrated Management Controller.

---

## Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to [ucs-director-docfeedback@cisco.com](mailto:ucs-director-docfeedback@cisco.com). We appreciate your feedback.

## Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at [Cisco Profile Manager](#).
- To get the business impact you're looking for with the technologies that matter, visit [Cisco Services](#).
- To submit a service request, visit [Cisco Support](#).
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit [Cisco Marketplace](#).
- To obtain general networking, training, and certification titles, visit [Cisco Press](#).
- To find warranty information for a specific product or product family, access [Cisco Warranty Finder](#).

### Cisco Bug Search Tool

[Cisco Bug Search Tool](#) (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.





# CHAPTER 1

## New and Changed Information

This chapter contains the following sections:

- [New and Changed Information for This Release, on page 1](#)

## New and Changed Information for This Release

The following table provides an overview of the significant changes to this guide for this current release. The table does not provide an exhaustive list of all changes made to this guide or of all new features in this release.

### New and Changed Information in Cisco UCS Director Release 6.8(3.0)

Feature	Description	Where Documented
Support for Searching Workflows	You can search the workflows by tasks, activities, inputs, outputs, and global variables.	<a href="#">Searching a Workflow, on page 99</a>
Enhancement to Export Service Request Log	You can download the service request log in your local system.	<a href="#">Viewing Service Requests, on page 75</a>

### New and Changed Information in Cisco UCS Director Release 6.8(1.0)

Feature	Description	Where Documented
Support of Global Variables in Workflows	You can export and import the Global variables in the workflows.	<a href="#">Running a Quick Export on Workflows, on page 88</a> <a href="#">Exporting Workflows, Custom Tasks, Script Modules, and Activities</a> <a href="#">Importing Workflows, Custom Tasks, Script Modules, and Activities, on page 91</a>

Feature	Description	Where Documented
Support for Static Key	You can encrypt dynamic key. Static key will not be supported	<a href="#">Running a Quick Export on Workflows, on page 88</a> <a href="#">Exporting Workflows, Custom Tasks, Script Modules, and Activities</a> <a href="#">Importing Workflows, Custom Tasks, Script Modules, and Activities, on page 91</a>

### New and Changed Information in Cisco UCS Director Release 6.8

Feature	Description	Where Documented
Create a trigger template	You can create trigger templates using user interface.	<a href="#">Creating a Trigger Template, on page 66</a>
Lock or unlock a service request	You can lock a service request to prevent archiving of the service request. You can use the <b>Lock/Unlock</b> option to lock or unlock one or more service requests.	<a href="#">Defining a Workflow, on page 17</a> <a href="#">Locking a Service Request, on page 80</a>



## CHAPTER 2

# Overview

---

This chapter contains the following sections:

- [Cisco UCS Director Orchestrator, on page 3](#)
- [Going Further, on page 3](#)
- [Common Icons, on page 4](#)

## Cisco UCS Director Orchestrator

The Cisco UCS Director Orchestrator (also called the Cisco UCS Director Orchestration Engine, or just Orchestration) enables IT administrators to automate cloud deployment and provisioning and to standardize IT services.

At the scale of cloud computing, manually performing actions such as creating VMs and provisioning networks is prohibitively time-consuming. In Cisco UCS Director, these actions, or *tasks*, are executable elements that can be run from the GUI. Cisco UCS Director Orchestrator further automates these complex tasks by organizing them into *workflows*.

Orchestrator works by executing a series of scripted actions called *tasks*. Each task performs one action. By connecting tasks so that the input of one task is the output of a previous task, you build up a *workflow* to automate administrative processes such as creating VMs, provisioning bare metal servers, setting up storage, compute, and network resources, and many others.

## Going Further

The following documentation describes how to go further with Cisco UCS Director Orchestrator by using advanced scripting capabilities not covered in this basic guide to Orchestration.

- For a description of scripting technologies available in Cisco UCS Director and help choosing the right solution for your application, see the [Cisco UCS Director API Customization and Integration Guide](#).
- For an introduction to developing custom tasks, see the [Cisco UCS Director Custom Task Getting Started Guide](#).
- For examples of scripts that can be used to customize tasks, see the [Cisco UCS Director Cloupiascript Cookbook](#).

For more information about installing, configuring, and administering Cisco UCS Director, see the guides available at the [Cisco UCS Director Product Support Home website](#).

## Common Icons

The following table provides information about the common icons used in the user interface. You can see the name of an icon when you hover over it with your mouse. Some icons may have a different name, depending upon the context in which they're used.

Icon	Name	Description
	Search	Search is available on the header and on individual screens. Click <b>Search</b> on the header to find a report in the user interface. Click <b>Search</b> on an individual screen to find one or more items in the report.
	Alert	Alert is available on the header. Click <b>Alert</b> to view your diagnostic system messages.
	User	User is available on the header. Click <b>User</b> to access your profile, or to log out of the user interface.
	Export	Export is available on individual screens. Click <b>Export</b> to export the content of the report that is visible on the screen.
	Import	Import is available on individual screens. Click <b>Import</b> to import a file.
	Refresh	Refresh is available on individual screens. Click <b>Refresh</b> to refresh the data that is visible on the screen.
	View Details	View Details is available on individual screens. Click <b>View Details</b> to see details about the selected row in the table.
	Table View	Table view is available for your application containers and catalogs. Click <b>Table View</b> to view your application containers or catalogs in a table with details about each application container or catalog displayed.
	Tile View	Tile view is available for your catalogs and application containers. <b>Tile View</b> to view your catalogs and application containers in a tiled view of icons. With this view, you must click on an icon to see details about that catalog item or application container.
	Create	Create is available on individual screens. Click <b>Create</b> to create a new object, such as a VM disk.
	Add	Add is available on individual screens. Click <b>Add</b> to add an item to an existing object, such as adding a catalog item to an existing catalog folder. The name of this icon may also include the item that you want to add, such as <b>Add Catalog</b> .

Icon	Name	Description
	Expand a list of values	<p>Lists of values are available in forms when you must select one (single select) or more (multi select) items, for example IP addresses or VMs.</p> <p><b>Tip</b>      A <i>multi-select</i> list displays a check box in the upper left corner of the table next to the first column label. Click this check box to select all items in the table.</p> <p>The item label appears to the right of the <b>Expand</b> icon. Click the <b>Expand</b> icon to display the list, then select an item or items.</p> <p>Once an item is selected, its value appears in parentheses to the right of the <b>Expand</b> icon and label.</p>
	Collapse a list of values	Once a list of values is expanded, the <b>Expand</b> icon changes to a <b>Collapse</b> icon. Click the <b>Collapse</b> icon to hide the list of values, for example to see what is beneath the list.
	Edit	Edit is available on individual screens. Click <b>Edit</b> to modify an existing object, such as a catalog item or a VM disk.
	Delete	Delete is available on individual screens. Click <b>Delete</b> to delete an object, such as a catalog item or a VM disk.
	Custom Actions	This icon represents additional tasks that do not have an associated icon. It also represents the default icon available in the user interface.
	Favorites	Adds a page to the <b>Favorites</b> menu. You can use this option to view frequently accessed pages more quickly.
	Filter	Provides filtering parameters on the page.





## CHAPTER 3

# Orchestration Concepts

---

This chapter contains the following sections:

- [Introduction to UCS Director Orchestration, on page 7](#)
- [Tasks, on page 7](#)
- [Workflows, on page 8](#)
- [Service Requests, on page 8](#)
- [Input and Output, on page 8](#)
- [Workflow Validation, on page 9](#)
- [Workflow Versioning, on page 10](#)
- [Approvals, on page 10](#)
- [Rollback, on page 10](#)
- [Libraries and Catalogs, on page 11](#)
- [Activities, on page 11](#)
- [Managing the Workflow Lifecycle, on page 12](#)

## Introduction to UCS Director Orchestration

The concepts described in this chapter are essential to understanding and using Cisco UCS Director Orchestrator. Even if you are familiar with orchestration in general, review this chapter for concepts specific to Cisco UCS Director.

We recommend that you become familiar with the functionality available in Cisco UCS Director before using Cisco UCS Director Orchestrator. If you are unfamiliar with Cisco UCS Director, a thorough introduction is available in the *Cisco UCS Director Fundamentals Guide*.

## Tasks

A task is the atomic unit of work in Cisco UCS Director Orchestrator. A task is a single action or operation with inputs and outputs (except in rare cases where the task operation does not require inputs or outputs). A task cannot be decomposed into smaller operations (exception: a compound task is made up of atomic tasks; see [Creating a Compound Task, on page 101](#). But for now, think of a task as an indivisible unit of orchestration work).

Cisco UCS Director has a task library containing hundreds of predefined tasks that cover most of the actions an administrator must perform using Orchestration. In cases where there is no suitable predefined task, you can create custom tasks; see the [Cisco UCS Director Custom Task Getting Started Guide](#).

Some examples of predefined tasks are:

- SSH Command task - Execute a command in a secure shell (SSH) session.
- Collect Inventory Task - Gathers information about available devices.
- Create New User - Add a user to the system.
- New VM Provision - Create a new VM (hypervisor-specific).

## Workflows

A workflow is a series of tasks arranged to automate a complex operation. The simplest possible workflow contains a single task, but workflows can contain any number of tasks.

Workflows are the heart of the Cisco UCS Director Orchestrator; they enable you to automate processes of any level of complexity on your physical and virtual infrastructure.

You build workflows using the **Workflow Designer**, a drag-and-drop interface. In the **Workflow Designer**, you arrange tasks in sequence and define inputs and outputs to those tasks. Outputs from earlier tasks are available to use as inputs to any subsequent task.

Looping and conditional branching can be implemented using special flow-of-control tasks.

## Service Requests

Service Requests are closely related to workflows. You create service requests by running workflows; a service request is generated every time you execute a workflow in Cisco UCS Director. A service request is a process under the control of Cisco UCS Director.

You can schedule a workflow for later execution, and Cisco UCS Director stores details of completed service requests. Thus a service request can have one of several states depending on its execution status: it can be scheduled, running, blocked (for example, awaiting an approval; see [Approvals, on page 10](#)), completed, or failed (a service request can fail when one of its component tasks fails to execute properly; see [Rollback, on page 10](#) and [Resubmitting a Service Request, on page 77](#)).

## Input and Output

Both tasks and workflows can have any number of input and output variables (*inputs* and *outputs*).

Any task or workflow input can be either mandatory or optional. A task or workflow cannot run without all of its mandatory inputs. You define whether an input is mandatory or optional when you create the task or workflow.

### Input and Output Types

There are many input types defined in Cisco UCS Director representing a broad selection of categorical, numeric, and text parameters. For example, some existing data types are:

- Generic Text Input (the least restricted input type)
- IPv4 Address
- OS Type
- Memory Size Selector

You choose existing input types from a list that displays a name, type, and category for each variable. The list can be filtered to make finding a given data type manageable.

If none of the existing data types serves your need in a particular application, it is possible to create custom data types by defining restrictions on existing data types.

### Connecting Inputs and Outputs

When you construct workflows, you connect an output of one task to the input of another task. For example, consider the following two tasks:

1. A *Create User* task that produces a *user ID* as an output.
2. An *Add User to Group* task that takes a *group ID* and a *user ID* as input.

In this instance, you would position Task 1 before Task 2, feeding the *user ID* output of Task 1 to the *user ID* input of Task 2.

A workflow's inputs and outputs connect to the inputs and outputs of one or more of its tasks.

### User and Administrator Inputs

Administrator inputs (*admin inputs*) are default values specified at the time a workflow is defined. When defining the workflow the administrator can also allow users to override a default value.

Workflow *user inputs* are inputs with values specified by a human user at runtime. User inputs can have default values. A user has the option of accepting a default input or overriding it. If the input is mandatory and no default value is specified, the user must input a value.

Instead of dictating a specific input value, an administrator can place restrictions on a user input value when creating a workflow. For example, an admin can restrict the values of an IP address input to a certain range. In this case the IP address is still a user input, but with a restricted range of allowable values.

Some workflows must have admin inputs defined for all of their mandatory inputs. This is the case when workflows are run without human intervention—For example, when a system is configured to run a scheduled workflow at a specific time.

## Workflow Validation

Orchestrator provides a mechanism for validating the flow of data from one task to the next in a workflow. Workflow validation checks the data bindings and connections between tasks. Some common issues detected during validation are:

- Missing mandatory values—A required value is not supplied to a task.
- Mapping mismatch—A connected task input/output pair do not have the same data type.
- Missing inputs—This can happen especially after an import or upgrade.
- Task handler not found—An underlying class required to run the task is missing. This message appears if you try to use an unlicensed feature, or if you try to validate a workflow template.

Orchestrator supplies a wizard-based issue resolver. When you validate a workflow, the wizard presents a list of issues along with suggestions for fixing those issues. Some issues require additional information or input from you. Other issues are quick fixes that are resolved for you.

## Workflow Versioning

All Orchestrator workflows have a version history. With the version history, you can revert a workflow to an earlier version or create a new version.

The *default* version of a workflow is the one that appears on the **Workflows** page. You can set any version in the workflow's history to be the default workflow. When you modify a workflow, you modify only the default version; other versions are unchanged.

Workflows are deleted on a version basis. That is, you can delete one or more versions of a workflow without affecting the remaining versions.

## Approvals

An approval is a "gate" task that requires the intervention of a Cisco UCS Director user to allow a workflow to run to completion. This user is typically an administrator who has go-or-no-go authority over the workflow process. See [Approving and Denying Service Requests, on page 78](#).

You can create *custom approvals* that allow users to enter input values when they approve a workflow. For example, you can create a custom approval to enable an IT administrator to approve provisioning of a VM, then specify the memory size of the VM before the workflow creates the VM. See [Creating Custom Approvals, on page 102](#).

When an approval task is used within a loop task of a workflow, the approval request will be initiated to the mapped user for each iteration. The progress of the approval task can be seen under the **Show Details Status** tab of the **Service Request Submit Status** screen. The approval task behaviour remains same during rollback of the workflow.

## Rollback

Workflows can be "rolled back" to a state identical or similar to the state before the workflow was executed. You can do this, for example, to remove virtual components that were created in error.

The term "rollback" often implies that a process is transactional, especially in systems using relational technology. However, it is important know that workflows are *not* transactional in a relational database sense. Instead, rollbacks work like this:

- Each task consists of two scripts. One script performs the work that the task was designed for. The other is a rollback script designed to "undo" the task. For example, if a task creates a VM, the rollback script deletes the VM.
- When you run a workflow, the scripts of the tasks in that workflow are executed in the order indicated by the workflow.
- When you roll back a workflow, the task rollback scripts are run in the reverse of the order they are run during normal workflow execution.
- A request to roll back a workflow creates a new service request, unconnected to the original service request.
- State information can be saved and used to roll back a task. For example, if you run a task to resize a VM, the pretask size can be saved to enable a rollback. This state persistence is a feature of the API that is used to write tasks; see the *Cisco UCS Director Custom Task* documentation for details.
- Tasks are atomic; workflows are not. You can roll back a workflow starting with any task in the workflow, but you cannot partially roll back a task.
- It is possible for a workflow to partially succeed; that is, for some but not all of its tasks to execute. Similarly, it is possible for a rollback to completely or partially fail; that is, that some or all of the rollback scripts could fail to execute.
- It is possible for a task to be written with a defective rollback script, or without a rollback script altogether. (Omitting the rollback script is not recommended.)

See [Rolling Back a Service Request, on page 76](#) for instructions on how to roll back a workflow.

## Libraries and Catalogs

Libraries and catalogs are collections of predefined tasks and workflows, respectively, from which you can build workflows specific to your needs. For example, you can:

- Copy a predefined workflow that performs a process you require and modify it with parameters particular to your installation.
- Copy a predefined workflow that performs a process similar but not identical to your needs and modify it by adding, modifying, or removing tasks.
- Build a workflow to suit your unique needs entirely from predefined tasks.

## Activities

An activity is a placeholder for a type of workflow—A kind of generic front-end that abstracts workflows from their implementation details. You can create an activity for a generic task then associate one or more workflows with the activity to actually perform the required work.

For example, consider a situation in which you need to create two different types of datastore—Say a NetApp datastore and an EMC datastore. You can define one activity called "Create Datastore," then associate both workflows with it. The activity matches input conditions at runtime to determine which storage type is being used, and then run the correct workflow.

In addition, an activity can be used as a workflow task, providing more flexibility to perform context-dependent activities within workflows.

Activities are described in [Activities, on page 117](#).

## Managing the Workflow Lifecycle

Workflow management consists of the organization, storing, updating, creation, and deletion of catalogs of workflows. Cisco UCS Director provides a complete set of actions to enable workflow management.

Operations used to manage workflows are described in the following table, with references to their respective task descriptions in this guide.

Operation	Description
<b>Add a Workflow</b>	Create a workflow from scratch. See <a href="#">Creating a Workflow, on page 17</a> .
<b>Add a Workflow to a Catalog</b>	Create a catalog of workflows for subscribers, for example application administrators. See the <a href="#">Cisco UCS Director Administration Guide</a> for information about publishing and managing catalogs.
<b>Add and Arrange Tasks in a Workflow</b>	Use the <b>Workflow Designer</b> . See <a href="#">Configuring a Task in a Workflow, on page 33</a> and <a href="#">Connecting a Task to a Workflow, on page 36</a> .
<b>Choose a Workflow Version</b>	Choose the active version of a workflow. The active version is called the <i>default</i> version and appears on the <b>Workflows</b> page. See <a href="#">Choosing the Default Version of a Workflow, on page 96</a> .
<b>Clone a Workflow</b>	Create a renamed copy of a workflow. The copy has a new version history. See <a href="#">Cloning a Workflow, on page 98</a> .
<b>Create a New Version of a Workflow</b>	Workflows have a version history. You can use any version of a workflow from its history at any time. See <a href="#">Creating a New Version of a Workflow, on page 95</a> .
<b>Delete a Workflow</b>	Remove one or more versions, or all versions, of a workflow from Cisco UCS Director. See <a href="#">Deleting a Workflow, on page 87</a> .
<b>Edit a Workflow</b>	Modify the name, location, inputs, and outputs of a workflow (but not the tasks). See <a href="#">Workflow Editing, on page 81</a> .
<b>Execute a Workflow</b>	Create a service request immediately from the selected workflow. See <a href="#">Executing a Workflow, on page 58</a> .

Operation	Description
<b>Export a Workflow</b>	Save a workflow in a format that can be loaded in another Cisco UCS Director appliance. See <a href="#">Exporting Workflows, Custom Tasks, Script Modules, and Activities</a> , on page 89.
<b>Export a Workflow as a Template</b>	Export the selected workflow as a template in XML-based format. See <a href="#">Exporting Workflows as Templates</a> , on page 92.
<b>Import a Workflow</b>	Load a workflow created elsewhere into Cisco UCS Director. See <a href="#">Importing Workflows, Custom Tasks, Script Modules, and Activities</a> , on page 91.
<b>Lock or Unlock a Workflow</b>	Lock a workflow to prevent any modifications. Once locked, the workflow cannot be deleted or edited.
<b>Move a Workflow</b>	Move a workflow to a new directory. See <a href="#">Workflow Editing</a> , on page 81.
<b>Validate a Workflow</b>	Analyze the workflow to determine that the task inputs and outputs are properly connected. See <a href="#">Validating a Workflow</a> , on page 38.
<b>View an Entire Workflow</b>	View the entire structure of a workflow, in which you can pan a magnified section to view details. See <a href="#">Saving a Picture of a Workflow</a> , on page 100.





## CHAPTER 4

# Creating Workflows

---

This chapter contains the following sections:

- [Viewing the Task Library](#), on page 15
- [The Workflow Designer](#), on page 15
- [Creating a Workflow](#), on page 17
- [Example: Creating a Workflow](#), on page 39
- [Resolving Workflow Validation Issues](#), on page 41
- [Changing Flow of Control in Workflows](#), on page 42
- [Locking Resources](#), on page 51
- [File Uploads](#), on page 53

## Viewing the Task Library

Cisco UCS Director contains web-based documentation on all predefined tasks. To view the documentation, do the following:

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Click **Task Library**.
  - Step 4** Check **Regenerate document** to include new tasks and tasks created using Cisco UCS DirectorOpen Automation.
  - Step 5** Click **Submit**.  
The **Orchestration Task Library** appears in your default web browser.

**Note** There are over 2000 entries. You can use your browser's search function to find a particular task.

- Step 6** Click an entry to see detailed information about a task.
- 

## The Workflow Designer

The **Workflow Designer** is a graphical tool for building workflows. In the **Workflow Designer**, you add and arrange tasks to build up a workflow. Following is a brief description of the parts of the **Workflow Designer**.

To bring up the **Workflow Designer**:

1. Select a workflow from the **Workflow** report or create a new workflow. See [Defining a Workflow, on page 17](#).
2. From the **More Actions** drop-down list, choose **Workflow Designer**.

On the **Workflow Designer** screen, observe the following components:

### Work Area

The large area with grid lines is the **Work Area** where you add, arrange, and connect tasks to build a workflow. Depending on your zoom settings, the **Work Area** might show only a portion of the current workflow.

### Workflow Title

Across the top of the **Workflow Designer** are the name of the workflow and the **Close Window (X)** control.

### Action Menu

Across the top of the **Work Area** from the left are controls to **Edit**, **Create a New Version**, **Validate**, **Execute** the workflow in the **Workflow Designer**, and **Export as Image** (export the workflow diagram as a graphic image file). On the right is a count of the tasks in the workflows. The count does not include the default **Start**, **Completed (Success)**, or **Completed (Failed)** tasks. Conditional flow-of-control tasks (**Conditional Task** and **If Else**), compound tasks, and custom tasks are also broken out separately.




---

**Tip** Hover over a task count icon to display a list of the tasks names.

---

### Available Tasks

To the left of the **Work Area** is the **Available Tasks** directory. This directory structure contains every available orchestration task, including both built-in and custom tasks. Type in the **Search for ...** field to quickly find a named task. The available tasks narrow to tasks containing your search text as you type. To add a task to a workflow, drag-and-drop the task from the **Available Tasks** directory to the **Work Area**.

Documentation for the last selected task is displayed at the bottom of the **Available Tasks** area. Click **Read More ...** at the end of the documentation to bring up the selected task in the **Task Library** in your browser. See [Viewing the Task Library, on page 15](#).

### View Controls

In the upper right corner of the **Work Area** are the **View Controls**. From top to bottom, the **View Controls** are:

#### Auto Layout

Click to automatically arrange the tasks in the **Work Area** in a vertical stack with consistent spacing and alignment.

#### Compact View Mode

Click to toggle the height of the tasks (one line of text or two) and simultaneously activate **Auto Layout**.

#### Fit to Window

Click to rescale the entire workflow diagram to fit in the **Work Area**. Depending on the size of the workflow, this might enlarge or reduce the diagram.

#### Zoom

Use the zoom controls to change the size of the displayed workflow:

- + Zoom in

- - Zoom out

### Show / Hide Navigator

Click to toggle the **Navigator** view on and off. The **Navigator** is a map of the entire workflow in the lower right corner of the **Work Area**. The **Navigator** shows the portion of the workflow visible in the **Work Area** framed in a blue border. You can pan over the entire workflow to bring different parts of the workflow into the visible **Work Area**. Use the **Navigator** to examine large workflows at a scale at which text is legible.

## Creating a Workflow

Creating a workflow can be broken down into five overall steps:

1. Define the workflow. To define a workflow, you name the workflow and select a few other details.
2. Define the workflow inputs and outputs. You specify which input parameters the workflow requires to execute, and what output results from its execution.
3. Once the workflow inputs and outputs are defined, use the **Workflow Designer** to add a workflow task to the workflow.
4. Still in the Workflow Designer, connect the task into the workflow. Repeat these two steps, defining and adding a task, until you have incorporated all the tasks needed for the workflow to carry out its function.
5. Validate the workflow. Validating a workflow ensures that there are no errors in connecting task inputs and outputs.

These five steps are described in the following sections.

## Defining a Workflow

To define a workflow, do the following:

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Click **Add (+)**.
- Step 4** In the **Add Workflow Details** screen, complete the following fields:

Name	Description
<b>Workflow Name</b>	The workflow name.  <b>Note</b> Workflow names cannot contain the following characters: <code>\'%&amp;'*+./:;&lt;=&gt;?^/</code>
<b>Version</b>	Displays which version of the workflow you are creating or editing.
<b>Description</b>	A description of the workflow. (Optional.)

Name	Description
<b>Workflow Context</b>	<p>The workflow context. Cisco UCS Director Orchestrator supports the following options:</p> <ul style="list-style-type: none"> <li>• <b>Any</b>—Enables you to use the workflow in any context.</li> <li>• <b>Selected VM</b>—Enables you to execute the workflow only when you choose a VM.</li> </ul>
<b>Save as Compound Task</b>	<p>Check <b>Save as Compound Task</b> to save this version of the workflow as a compound task. See <a href="#">Creating a Compound Task, on page 101</a>.</p>
<b>Publish Task outputs as Compound Task outputs</b>	<p><b>Publish Task outputs as Compound Task outputs</b> appears if you check <b>Save as Compound Task</b>.</p> <p>Check <b>Publish Task outputs as Compound Task outputs</b> to expose all the task outputs as outputs of the compound task. Leaving it unchecked suppresses the outputs of the included tasks so that the compound task exposes only the workflow outputs.</p>
<b>Always execute during System initialization</b>	<p>Check <b>Always execute during System initialization</b> to execute the workflow as part of the startup sequence when Cisco UCS Director is initialized.</p> <p><b>Note</b> Because the workflow is executed before the Cisco UCS Director GUI starts, workflows that require user input cannot be executed at system startup.</p>
<b>Disable Auto-population of inputs</b>	<p>Check this check box to disable auto-population of input from previous service requests.</p>
<b>Place in New Folder</b>	<p>Create a new folder in which to place the workflow. If you check <b>Place in New Folder</b>, the <b>Select Folder</b> drop-down list becomes a text field. Type the name of the new folder.</p>
<b>Select Folder</b>	<p>The folder in which to place the workflow.</p> <p><b>Note</b> “Effective with Cisco UCS Director 6.8, the <b>Default Folder</b> option is not available.</p>
<b>Rollback Workflow on Failure</b>	<p>Check this check box to automatically initiate a rollback of the workflow in the event of a failure.</p> <p>Selecting the <b>Rollback Workflow on Failure</b> option ensures that the service request is automatically rolled back when the service request fails. In this scenario, a rollback service request is initiated.</p>

Name	Description
<b>Abort Rollback on Failure</b>	<p>Check this check box to abort the rollback of the workflow if the rollback request results in errors.</p> <p>Selecting the <b>Abort Rollback on Failure</b> check box terminates the rollback of a service request if the execution of any task within the rollback service request fails.</p>
<b>Lock</b>	<p>Check this check box to lock a service request.</p> <p>Selecting the <b>Lock</b> check box prevents archiving of the service request.</p>
<b>Notify status of execution to initiating user</b>	<p>Email the result status to the user who invoked the workflow. If checked, you can notify more users by entering their email addresses in the <b>Additional User(s) to send Email Notification</b> field.</p> <p>The result status for the workflow can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Completed status</b></li> <li>• <b>Failed execution status</b></li> <li>• <b>Cancelled execution status</b></li> </ul> <p>Leave <b>Notify status of execution to initiating user</b> unchecked to use an email policy.</p>
<b>Email policy</b>	<p>Instead of specifying email recipients individually, you can use an orchestration notification email policy. Default policies exist to notify:</p> <ul style="list-style-type: none"> <li>• The user that starts execution of the workflow</li> <li>• The user and the user's group</li> <li>• The user's group and that group's administrator</li> <li>• External email IDs</li> </ul> <p>You can define your own policies as well by navigating to <b>Administration &gt; System &gt; Orchestration Policy</b> and selecting <b>Add</b>. Policies you create are available in the <b>Email policy</b> drop-down list.</p>

### What to do next

Click **Next** to define input and output for the workflow.

## Defining Workflow Input and Output

You can define a set of inputs for a workflow. You can assign values for these inputs or restrict the values available to users at runtime. You can also choose whether to allow the user to change the administrator values at runtime.

To define workflow inputs and outputs, complete the following steps.

### Before you begin

Define the workflow in the **Workflow Add Details** screen as described in the previous topic.

**Step 1** You can define any number of inputs for a workflow.

Rather than define inputs for a workflow, you can instead associate the workflow with an activity and use the activity's inputs as the workflow's inputs. See [Activities, on page 117](#).

- To associate the workflow with an activity:
  - a. Check **Associate to Activity**.
  - b. In the **Activity** drop-down list, choose the activity with which to associate the workflow. The activity's inputs populate the input list.
  - c. Skip the rest of this step.
- To define inputs for the workflow without an activity, continue with this step:
  - a) To allow a user to copy inputs from a previously run service request (SR) of the same workflow, check **Load Inputs from Previous SR**. See [Executing a Workflow, on page 58](#). The default is to allow SR input loading.
  - b) On the **Workflow User Inputs** screen, click **Add (+)**.
  - c) On the **Add Entry To** screen, complete the following fields:

Name	Description
<b>Input Label</b>	The name or label of the input parameter.
<b>Input Description</b>	A description of the input parameter.
<b>Optional</b>	Make the input parameter optional (not mandatory) at the time of workflow execution.
<b>Input Type</b>	<p>Click to choose the data type of the input parameter.</p> <p><b>Note</b> Reduce the number of input types by typing a match string in the filter field at the top of the list.</p> <p><b>Note</b> If you later edit a workflow, the <b>Input Type</b> field is not editable.</p>
<b>Validate</b>	Appears after you select a data type. Click <b>Validate</b> to continue.

Name	Description
<b>Multiline/Multiple Value Input</b>	<p>The effect of this option depends on the input type.</p> <p>This option is available only for the Generic Text input type. Check the box to allow the user or administrator to enter multiple lines for the input value.</p> <p><b>Note</b> You cannot map a multiline workflow input to a single-line task input. When creating or editing a task, no multiline workflow inputs appear in the <b>User Input</b> drop-down list on the <b>User Input Mapping</b> page.</p>
<b>Input Field Size</b>	This list appears if you choose <b>Multiline Input</b> . Choose a size for the text area that is displayed to enter the input value.
<b>Value Restrictions</b>	<p>Depending on the input type that you have chosen, one or more of the following are available:</p> <ul style="list-style-type: none"> <li>• <b>Admin Input</b>—Check to populate inputs with default values. User input is then optional at run time.</li> <li>• <b>Admin Input Filter</b>—Check to define a rule-based filter to restrict available input values. User input is required at run time from the restricted range of values. See <a href="#">Admin Input Filters, on page 31</a>.</li> <li>• <b>Admin Input List</b>—Check to define an item-by-item list of available input values. User input is required at run time from the admin-selected list of values.</li> </ul>
<b>Admin Input</b>	Available when you check <b>Admin Input</b> . Click <b>Select</b> to choose one or more admin input values.
<b>Override Admin Input Value</b>	Available when you check <b>Admin Input</b> . Check <b>Override Admin Input Value</b> to enable the user to replace the administrator value at runtime. Leaving unchecked hides the user's input control for this value at runtime.
<b>Input Filter Criteria</b>	Available when you click <b>Admin Input Filter</b> . Type a filter expression to limit the values available at run time. See <a href="#">Admin Input Filters, on page 31</a> .
<b>Admin Input Value</b>	Available when you click <b>Admin Input Filter</b> . Displayed for reference—Use the column headings as attribute names in your filter expression.
<b>Select</b>	Available when you click <b>Admin Input List</b> . Click <b>Select</b> , then choose a value or values. All chosen values are available as user input choices at run time.

d) If necessary, add more inputs by repeating the previous two substeps.

**Step 2** Click **Next**.

- Step 3** On the **Define Workflow Input Processing Rules** page, decide whether you want to use input rules to control input display to users:
- To opt out of workflow input rules, ensure that **Enable Input Rules** is unchecked. All user inputs are displayed in full when the workflow is run.
  - To define input rules, add rules to the **Workflow Input Rules** table. Input rules are applied at runtime only if **Enable Input Rules** is checked. See [Adding an Input Rule, on page 24](#).

**Step 4** Click **Next**.

**Step 5** To enter outputs for the workflow, do the following.

- In the **Workflow User Outputs** screen, click **Add (+)**.
- On the **Add Entry To** screen, complete the following fields:

Name	Description
<b>Output Label</b>	The name or label of the output parameter.
<b>Output Description</b>	A description of the output parameter.
<b>Optional</b>	Make the output parameter optional (not mandatory) at the time of workflow execution.
<b>Type</b>	The data type of the output parameter. During the editing and saving of a workflow task, the <b>Type</b> field is no longer editable.

- If needed, add more outputs by repeating the previous two substeps.

**Step 6** Click **Submit**.

The new workflow opens in the **Workflow Designer**.

### What to do next

Add tasks to the workflow using the **Workflow Designer**. See [Configuring a Task in a Workflow, on page 33](#).

## Dynamic Workflow Inputs

You can create workflow input forms in which display of an input is dependent on the selected value of a previous input. This feature is called *dynamic inputs* or *dynamic forms*. It is sometimes called *progressive disclosure* in user experience (UX) terminology.

By default, all inputs of a workflow are displayed when a workflow is executed. To enable dynamic inputs and potentially hide some inputs in a workflow, you must explicitly turn on dynamic inputs and create input rules for that workflow.

Dynamic inputs are controlled by *workflow input processing rules*, or simply *input rules*. You define input rules when you create a workflow. See [Adding an Input Rule, on page 24](#). You can create any number of input rules for an input, and any number of input rules in a single workflow.

When a user submits the workflow as a service request, the input rules determine the inputs that the user is prompted for. Dynamic input rules are processed in the same order as the workflow inputs.

Dynamic inputs have no further effect once inputs are selected. When the service request is submitted, it runs no differently than if the inputs had been supplied by a user or administrator from a "static" input form.

The logic of a dynamic input rule is essentially an if-then construct:

```
If Field select input Is operation select value Then Show target inputs
```

The components of the construct are:

#### **select input**

A selectable input type. Selectable input types include tabular report columns, lists of values (LOVs), and Boolean inputs. Selectable input types do *not* include free-form input types such as generic text, date, datetime, and so on. The input must be a user (not admin) input. The user-selected value of this input, along with the input rule logic, controls which subsequent inputs are displayed.

#### **operation**

The operation used to compare *select input* and **select value** to yield a Boolean value for the `IF` condition. All string-matching operations are case-sensitive.

There are six possible operations:

##### **ANY**

Returns true for any value of the select input; that is, as long as the select input is not null.

##### **CONTAINS**

Returns true if the select value is a substring of the select input.

##### **DOES-NOT-CONTAIN**

Returns true if the select input is *not* a substring of the select input.

##### **EQUAL-TO**

Returns true if the select value exactly matches the select input.

##### **NOT-EQUAL-TO**

Returns true if the select value does *not* exactly match the select input.

##### **Regular Expression**

The select value must be a valid regular expression. Then the rule returns true if the select input matches the regular expression.

The regular expression must conform to Java regular expression syntax. If the regular expression is invalid, an error message is displayed when you try to save the rule.

#### **select value**

The value to be compared, using *operator*, to *select input* to determine the result of the `IF` condition. The **select value** is a text string. (The select value can contain variables – see below.) If the operator is **Regular Expression**, the string must represent a valid Java-style regular expression.

The **select value** can contain any combination of plain text and variables. Variables are referenced with the `${}` notation. See [Orchestration Global Variables, on page 106](#).

The variables must be available at the time the rule is applied. Thus for example, previous inputs in the form can be used.

#### **target inputs**

A list of inputs that are to be displayed when the `IF` condition evaluates to true. The *target inputs* can be any of the inputs defined for the workflow, of any input type.

When the `IF` condition evaluates to false, the target inputs are not displayed.

Target inputs can be optional or mandatory inputs. If a mandatory target input is not displayed because of an input rule, its value is set to an empty string.




---

**Note** If the rule fails to display a mandatory target input, the mandatory input is set to an empty string. It is up to the rule creator to ensure that the inputs behave as expected.

---

The *selection input* must appear in the workflow input form before any of the *target inputs*.

For example, you could use the following logic to require that two IP address inputs be shown only if they are statically defined:

If Field **NAT Type** Is **Equals Static** Then Show **Real IP Address, Masked IP Address**

Input rules are defined on the **Add Input Rules** page during workflow creation. See [Adding an Input Rule, on page 24](#).



**Note** The order of the input rules on the **Add Input Rules** page is important. If there are multiple rules for the same input, the rules are evaluated in the order shown on the **Add Input Rules** page. The last rule to be evaluated controls the inputs that are shown or not shown during execution. For an example of this behavior see [Example: Combining Advanced Input Processing Rules, on page 29](#).

## Adding an Input Rule

You can show (or, indirectly, hide) workflow inputs in a user input form based on a value selected for a previous input. You do this by defining *workflow input processing rules*, or simply *input rules*. See [Dynamic Workflow Inputs, on page 22](#). To create an input rule, do the following:

### Before you begin

Begin creation of a new workflow as described in [Creating a Workflow, on page 17](#). On the **Add User Inputs** page, create at least two user inputs. One of the inputs, the *select input*, must be a selectable input type such as a list of values (LOV) or tabular report column. One or more other inputs, the *target inputs*, are optional and are visible or hidden at the time of workflow execution depending on the rules and the value selected for the select input. The select input must appear before any of its target inputs that will be shown or hidden.

Once the user inputs are defined, advance to the **Add Input Rules** page in the **Add Workflow** window.

- Step 1** On the **Add Input Rules** page, choose **Enable Input Rules**.
- Step 2** Click [+] to expand the **Rules for Input Fields** list.
- Step 3** Click + to add a new rule.
- Step 4** On the **Add Entry to Rules for Input Fields** page, complete the following fields:

#### Rule Name

Give the rule a descriptive name.

#### If Field

Choose the input on which the rule depends. Only selectable inputs – lists of values (LOVs), tabular report columns, and Booleans – are available.

#### Column

If the selected **If Field** value is a tabular report, choose one of its columns to use as the selection input.

#### Match all rows selected?

Available if the select input is a table. Because column values in a table are not necessarily unique, two or more column values can be identical. Check this option to cause the rule to evaluate as true only if the operation is true for the selected column across all selected rows. Default (not checked) is that only one row match is needed to make the rule true.

#### Is

Choose the comparison operator, or relationship, that defines the rule. The rule is triggered if **If Field** (for LOVs) or **Column** (for tabular inputs) bears this relationship with **Value**. See [Dynamic Workflow Inputs, on page 22](#).

**Value**

Type a value to compare with **If Field** or **Column**. **Value** can contain any combination of text and variables. Any undefined variable that appears in **Value** is not replaced at selection time and will probably cause the rule to fail. See [Orchestration Global Variables, on page 106](#).

**Select input fields to be shown to user when this rule is satisfied**

Select one or more inputs that are displayed when the rule evaluates to true.

**Step 5** When you are satisfied with the rule, click **Submit**.

**Step 6** Repeat these steps to create another input rule. You can create any number of input rules in a workflow.

---

**What to do next**

Click **Next** to advance to the **Add User Outputs** page and complete the workflow definition. See [Defining a Workflow, on page 17](#).

**Example: Using Basic Workflow Input Processing Rules**

This example discusses workflow processing rules to control the display of user inputs during workflow execution. It includes the following workflow rule features:

- Using a list input as a selection criterion
- Using a simple regular expression in a selection criterion

The example assumes you are entering NAT settings for a firewall. You first select the NAT type from a list. Depending on the NAT type (static, source, or destination), you then see inputs for the different IP addresses required.

**Before you begin**

Know how to create a workflow.

---

**Step 1** Open the first example workflow containing input rules. Do the following:

- a) Choose **Orchestration**.
- b) On the **Orchestration** page, click **Workflows**.
- c) Open the **Examples** folder.
- d) Select the **Example 1 - Workflow Rules - Firewall Config** workflow.
- e) Click the **Edit** action to edit the workflow.
- f) Click **Next** to advance to the **User Workflow Inputs** page.

Note that the first input is **Select NAT Type**, followed by several IP address inputs. The workflow input rules displays different IP inputs based on what NAT Type is selected.

**Step 2** Click **Next** to advance to the **Workflow Input Rules** page.

Note that there are three input rules, all dependent on the same select input:

- a. **NAT-Selection-Rule-Static** displays the Real and Masked IPs if NAT Type **Static** is selected. Admin input values has been assigned to the two IPs that are displayed when **Static** is selected. These values can be overridden by the user submitting the service request.

**NAT-Selection-Rule-Destination** displays the Outside IPs using a regular expression to match any NAT type that begins with "**Destination**".

b. **NAT-Selection-Rule-Source** displays the Inside IPs if the selected NAT Type contains "**Source**".

**Step 3** Click **Cancel** to go back to the **Workflows** page.

**Step 4** Demonstrate how the rules work:

a) Again choose the **Example 1 - Workflow Rules - Firewall Config** workflow.

**Tip** The workflow remains selected after you exit the **Edit Workflow** page.

b) Choose the **Execute Now** action.

c) On the **Executing Workflow** page, choose each of the NAT types in turn from the Select NAT Type drop-down menu.

Note that the IP input fields change as defined by the rules. The inputs change immediately every time you change the selected value. This is an important behavior to be aware of.

**Step 5** Click **Cancel** to quit without actually submitting the service request or click **Submit** to submit the service request. (The example service request does nothing but wait ten seconds and then exit.)

### What to do next

To learn more about workflow input rules, see [Example: Using Cascading Workflow Input Processing Rules, on page 26](#).

## Example: Using Cascading Workflow Input Processing Rules

This example discusses workflow processing rules to control the display of user inputs, including following features:

- Using a Boolean input as a selection criterion
- Cascading workflow input rules
- Overrideable admin inputs used as selection criteria
- Filtering admin input values using lists and expressions

This example enables the user to choose whether or not to select a workflow specific to a geographic region. The user first chooses whether or not to select a workflow at all; if so, they then select the region.

Selecting the region exposes the Select Workflow input. This is a simple example of *cascading inputs*. Notice that control of which workflows to present is controlled by an admin input filter on the **Select Workflow** input.

### Before you begin

Familiarize yourself with the previous Input Processing Rules example. See [Example: Using Basic Workflow Input Processing Rules, on page 25](#).

**Step 1** Open the second example workflow containing input rules. Do the following:

- a) Choose **Orchestration**.
- b) On the **Orchestration** page, click **Workflows**.
- c) Open the **Examples** folder.

- d) Select the **Example 2 - Workflow Rules - Regional Workflow Selection** workflow.
- e) Click the **Edit** action to edit the workflow.
- f) Click **Next** to advance to the **User Workflow Inputs** page.

Note that the first input is a Boolean user input, **Show Workflow**, followed by two list inputs, **Select Region** and **Select Workflow**.

**Step 2** Click **Next** to advance to the **Workflow Input Rules** page.

Note that there are two input rules **Workflow Y-N** and **Region Select**.

The first rule, Workflow Y-N, shows the input **Select Region** if the Boolean value is equal to **true** (that is, the box is checked).

The second rule, Region Select, displays the Select Workflow input if the selected region is not "**No Region**". Note that:

- The **Select Region** input is not shown initially. This is because the default (admin) value of the **Show Workflow** input is **false** (the box is unchecked).
- Input visibility cascades in this example: the **Workflow Y-N** rule exposes the **Select Region** input, which in turn is used by the **Region Select** rule to expose the **Select Workflow** input. Any number of inputs can be progressively exposed in this manner. It is up to the workflow creator to ensure that the input rules are consistent.

Note also that an admin input filter on the **Select Workflow** input restricts selections to the example workflows in the selected region using the value of **Select Region** as a variable.

**Step 3** Click **Cancel** to go back to the **Workflows** page.

**Step 4** Demonstrate how the rules work:

- a) Again choose the **Example 2 - Workflow Rules - Regional Workflow Selection** workflow.
- b) Choose the **Execute Now** action.
- c) On the **Executing Workflow** page, check **Show Workflow?**.  
The **Select Region** input is displayed.
- d) Select one of the regions from the Region Select list.  
The **Select Workflows** input appears.
- e) Click **Select Workflows**. Notice that the available workflows are all specific to the selected region.

**Step 5** Click **Cancel** to quit without actually submitting the service request.

---

### What to do next

To see more advanced features of workflow input rules, see [Example: Using More Advanced Input Processing Rules, on page 27](#).

## Example: Using More Advanced Input Processing Rules

This example discusses workflow processing rules to control the display of user inputs, including the following features:

- Using a table input as a selection criterion
- Requiring that all selected values of a selection criterion match in order to trigger the rule logic (multiple selection is possible only with tabular input)

For this example, suppose you set up a workflow that periodically "nags" approvers via emails and waits for a response. Depending on the workflow selected, you want to put in either a count (maximum number of emails) or a duration (total time before giving up), but not both.

All workflows are managed in a tabular input type containing an ID, name, description, and so on. When using a table as a selection criterion, you must choose which column to use for comparison. In this example, **Name** is used.

Also when using tables, multiple rows can be selected (since the column you select from may not be a unique across rows). You specify whether a rule must match all rows or if a match in any row fires the rule. In this example, if both count and duration workflows are selected, you want to specify a count. Therefore, you require **Duration** to match all rows but do not similarly restrict **Count**.

### Before you begin

Familiarize yourself with the previous Input Processing Rules examples. See [Example: Using Basic Workflow Input Processing Rules, on page 25](#) and [Example: Using Cascading Workflow Input Processing Rules, on page 26](#).

**Step 1** Open the third example workflow containing input rules. Do the following:

- Choose **Orchestration**.
- On the **Orchestration** page, click **Workflows**.
- Open the **Examples** folder.
- Select the **Input Rules 3 - Workflow Rules - Notification Timeouts** workflow.
- Click the **Edit** action to edit the workflow.
- Click **Next** to advance to the **User Workflow Inputs** page.

Note that the first input, **Select Workflow**, is a tabular input. The next input, **Count**, is a numerical input of type **resendMailCount**, and the last input, **Duration**, is a list input of type **timeDuration**.

**Step 2** Click **Next** to advance to the **Workflow Input Rules** page.

Note that there are two input rules. The rules have been designed to expose different inputs depending on the names of the selected **Select Workflow** input values.

#### Workflow Count Select

Exposes the **Count** field if one or more of the **Select Workflow** values contains "**Count**".

Note that **Match All Rows** is **No**, so a match on one row is sufficient to trigger the rule.

#### Workflow Duration Select

Note that **Match All Rows** is **Yes** for this rule. This causes the rule to expose the Duration field only if *all* of the selected **Select Workflow** values contain "**Duration**".

Note also that because the workflow selector is a table-based input, the rule requires you to choose a **Column** to specify which column the rule match is based on. Any column in the table can be used, though only some of the inputs are useful. In the **Select Workflow** table, the choices are **Name**, **Description**, **Folder**, **Version**, and **ID**.

**Step 3** Click **Cancel** to go back to the **Workflows** page.

**Step 4** Demonstrate how the rules work:

- Again choose the **Input Rules 3 - Workflow Rules - Notification Timeouts** rule.
- Choose the **Execute Now** action.
- On the **Executing Workflow** page, click **Select** to open the workflow selection table.
- Select two Workflows, one of which ends has a workflow name ending with "By Count". Click **Select**.

**Tip** Click the **Search** (magnifying glass) icon and type "Example" in the search field to narrow the search to example workflows only.

The **Count** input appears after the **Select Workflow** button on the **Executing Workflow** page.

- e) Clear your previous selection, then select both of the example Workflows that end with "Duration". The **Duration** input appears.
- f) Clear your previous selection, then select one of the example Workflows that end with "Duration" and one workflow that contains neither "Duration" nor "Count" in the name. The **Duration** input disappears. If one of your selected workflows happens to be a "Count" example, the **Count** input appears.

**Step 5** Click **Cancel** to quit without actually submitting the service request.

---

## Example: Combining Advanced Input Processing Rules

This example discusses workflow processing rules to control the display of user inputs, including the following features:

- Using an input variable in an input rule
- Using a global variable in an input rule
- Developing more complex cascading inputs using workflow input rules
- Writing two or more rules for the same input

In this example, assume that you have a collection of workflows, each of which is usually used in a particular region, that generate reports for entire regions. Further, you can set a global variable that specifies your default region; in that region, the reports are scheduled and a user does not usually have to run the reporting workflows manually.

You do not want to run these large reporting workflows any more than you have to. Running a report against the default region, or running it against a different region from the one it was designed for, is unusual, but should be allowed. This workflow uses checkboxes to alert the user to one of these conditions and to get the user's acknowledgement. In these cases, the checkbox input can be discarded (it is always "yes" if the user confirms the action).

There are several rules in this example workflow, creating several execution paths through the inputs.

### Before you begin

Familiarize yourself with the previous Input Processing Rules examples. See [Example: Using Basic Workflow Input Processing Rules, on page 25](#), [Example: Using Cascading Workflow Input Processing Rules, on page 26](#).

---

**Step 1** Open the third example workflow containing input rules. Do the following:

- a) Choose **Orchestration**.
- b) On the **Orchestration** page, click **Workflows**.
- c) Open the **Examples** folder.
- d) Select the **Input Rules 4 - Workflow Rules - Regional Workflows** workflow.
- e) Click the **Edit** action to edit the workflow.
- f) Click **Next** to advance to the **User Workflow Inputs** page.

Examine the inputs:

### Select Region

All of the rules in this example depend directly or indirectly on this input.

#### Default Region selected. Continue?

This input is displayed if the **Region** matches a global variable called *Example Default Region*.

**Note** You must create the global variable *Example Default Region* as a user-defined global variable for this example. It does not exist when you first install Cisco UCS Director.

### Select Workflow

This input presents a table of workflows to select from.

#### Regions are different. Continue?

This input is displayed if one or more selected workflows are for regions other than the selected one.

#### Execute Workflows

This is the final confirmation to run the workflows. You should only arrive here if:

- The selected workflows match the region *and* the region is not the default region. Or:
- One or more selected regions do not match the region, but the user has checked **Regions are different. Continue?**. Or:
- One or more selected workflows match the default region, but the user has checked **Default Region selected. Continue?**.

**Note** Workflow input rules cannot control the availability of the **Submit** button on the **Execute Workflow** page. In cases where an input is used to control workflow execution, the workflow must contain logic to stop execution if the checkbox input value is **no** (not checked).

**Step 2** Click Next to advance to the **Workflow Input Rules** page.

Note that there are six input rules:

#### Region Select Other

This rule shows the **Select Workflow** input if the selected region exactly matches one of **AMER**, **EMEA**, or **APAC**. Note the regular expression used to perform the match.

#### Region Select Default

This rule shows the **Default Region selected. Continue?** checkbox if the selected region is the same as the global variable *Example Default Region*. The rule does not expose the checkbox if the global variable contains anything else or is undefined.

#### WF Same Region Selected

This rule shows the final **Execute Workflows** checkbox if all the selected workflows are for the selected region.

#### WF Different Region Selected

This rule shows the **Regions are different. Continue?** checkbox if one or more of the selected workflows is for a region other than the one selected in the **Region** input.

#### Different Regions Acknowledged

This rule shows the **Execute Workflows** checkbox if **Regions are different. Continue?** is checked.

#### Default Region Acknowledged

This rule shows the **Execute Workflows** checkbox if **Default Region selected. Continue?** is checked.

**Step 3** Click **Cancel** to go back to the **Workflows** page.

**Step 4** Demonstrate how the rules work:

- a) Again choose the **Input Rules 4 - Workflow Rules - Regional Workflows** rule.
- b) Choose the **Execute Now** action.

- c) On the **Executing Workflow** page, choose a region in the **Select Region** list input. The **Select Workflow** checkbox appears (unless you have previously defined the *Example Default Region* global variable to be the name of the selected region).
- d) Click **Cancel** to exit workflow execution.
- e) Define a user-defined global variable named *Example Default Region* to be a generic text string with the value **APAC**. See [Creating Global Variables, on page 110](#).
- f) Choose the **Input Rules 4 - Workflow Rules - Regional Workflows** rule.
- g) Choose the **Execute Now** action.
- h) On the **Executing Workflow** page, choose **APAC** in the **Select Region** list input. The **Default Region selected. Continue?** checkbox appears.
- i) Check the **Default Region selected. Continue?** checkbox. The **Select Workflow** input appears.
- j) Choose one or more workflows. Ensure that at least one of the workflows does not contain "APAC". The **Regions are different. Continue?** checkbox appears.
- k) Check the **Regions are different. Continue?** checkbox. The **Execute Workflow** checkbox appears.

**Step 5** Click **Cancel** to quit without actually submitting the service request.

## Admin Input Filters

When you create a workflow, you can limit the set of entries from a table or list of values seen by the user at run time. You do this by either choosing the values explicitly using an **Admin Input List**, or by using a filter expression (**Admin Input Filter**). This section describes the syntax for writing admin input filters.

There are two types of data in Cisco UCS Director Orchestrator that you can filter:

- Tabular data—A table with two or more columns. You filter the columns to choose the rows to display.
- Lists of values (LOVs)—A list of items, all of the same type. You filter the values in the list.

In addition, some inputs are numeric, for example, disk storage sizes. You must include units when you filter numeric inputs.

### Filtering Tabular Data

To filter a tabular input, the input filter criteria have this form:

```
column_label operator expression [ AND column_label operator expression ]
```

where the elements of the expression are defined as follows:

Field	Entry
<i>column_label</i>	The label at the top of the column to be filtered. Type the name exactly as it appears. The label is case-sensitive and must include any spaces or special characters. Do not enclose the label in quotes.

Field	Entry
operator	<p>One of:</p> <ul style="list-style-type: none"> <li>• EQUALS</li> <li>• NOT_EQUALS</li> <li>• CONTAINS</li> <li>• NOT_CONTAINS</li> <li>• IS_SUBSET</li> <li>• NOT_SUBSET</li> </ul> <p>The operator name is case-sensitive; it must be all uppercase. IS_SUBSET and NOT_SUBSET are operators for list type inputs.</p>
[ ] (brackets)	The brackets denote that the enclosed expression is optional and can be repeated. Do not include the brackets themselves in the filter criteria expression.
AND	<p>Indicates extra criteria to be met. If one or more AND criteria are appended, all of them must be met to match a record.</p> <p>The criteria are applied serially. All records that meet the first criterion are subjected to the second criterion, then those results to the third, and so on.</p> <p>No operators besides AND can be used to append extra criteria. In particular, OR cannot be used. In other words, more criteria can be used only to narrow the search further, not to expand it.</p>
<i>expression</i>	<p>The syntax for <i>expression</i> is:</p> <pre>match_expression [ OR match_expression ]</pre>
OR	Indicates a union of any of the specified <i>match_expressions</i> . In other words, only one of the <i>match_expressions</i> has meet the criterion for that part of the expression to be true.
<i>match_expression</i>	<p>A simple regular expression that uses a subset of standard wildcard search characters and escape characters. The full POSIX regular expression syntax is not supported.</p> <p>You can match literal values provided they do not contain regex special characters or that you escape the special characters.</p> <p>For more information about regular expressions, consult any regular expression reference on the Internet; there are many.</p> <p>If <i>match_expression</i> is enclosed in quotes, a case-sensitive comparison is made, if applicable.</p>

Following is an example of a table-type filter:

```
Cloud CONTAINS vcenter21 OR vcenter98 AND Power Status EQUALS off OR unknown
```

The expression `Cloud CONTAINS vcenter21 OR vcenter98` matches names containing `vcenter21` and `vcenter98`, such as `vcenter211` and `TESTvcenter98`. If you are interested only in `vcenter21` and `vcenter98`, a better expression is `Cloud EQUALS vcenter21 OR vcenter98`.

### Filtering Lists of Values

LOV types are lists of single values. To filter an LOV you therefore omit the column name and the AND operator. The top-level search syntax is:

```
operator match_expression [ OR match_expression ]
```

The elements of the syntax are as described for tabular data, with the exception that `IS_SUBSET` and `NOT_SUBSET` operators are not permitted.

For example, if the power status values from the tabular data example were in an LOV, the search would look like this:

```
EQUALS off OR unknown
```

For example, you can filter from the list of values using expression.

```
CONTAINS '*vcenter21'
```

### Filtering Numeric Data

Certain numeric data such as disk size, LUN size, and volume size are specified with units--megabytes (MB), gigabytes (GB), or terabytes (TB). If the units are given in a table column, the units are part of the column label and must be included in your criteria; for example:

```
LUN Size (GB) EQUALS 10.0
```

Numeric expressions in filter criteria are regex matches and not numerical comparisons. For example, consider this filter criterion:

```
LUN Size (GB) EQUALS 10
```

The expression does *not* match records with LUN size given as `10.0`.

## Configuring a Task in a Workflow

Cisco UCS Director Orchestrator contains predefined tasks that you can use to create workflows. These tasks are organized in folders based on their functionality and can be accessed in the left panel of the **Workflow Designer** interface.

Documentation for a selected predefined task appears at the bottom of the task list in the **Workflow Designer**. To expand the documentation and open the entire **Task Library** in your browser, click **Read More ...** at the bottom left of the **Workflow Designer**. See [Viewing the Task Library, on page 15](#).



**Note** Cisco UCS Director Orchestrator's predefined tasks cover many if not most of the common operations required in a typical environment. For operations not represented in the predefined tasks, you can create custom tasks. See the [Cisco UCS Director Custom Task Getting Started Guide](#).

To configure a predefined task for use in a workflow, do the following:

**Before you begin**

Define a workflow and its inputs and outputs as described in the previous sections.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Choose a workflow.
- Step 4** Click **Workflow Designer**. The Cisco UCS Director Orchestrator **Workflow Designer** screen appears.
- Step 5** On the left side of the **Workflow Designer**, choose a category from the **Available Tasks** area by expanding folders in the task hierarchy.
- Step 6** Drag and drop your chosen task onto the **Workflow Designer** work area on the right side of the screen.
- Step 7** On the **Task Information** screen, complete the following fields:

Name	Description
<b>Task Name</b>	A name for this instance of the task. You can accept the automatically generated default or enter your own name for the task.
<b>Task Category</b>	The category (also the folder name) in which the task appears. Cannot be changed.
<b>Task Type</b>	The type (or generic name) of the task. Cannot be changed.
<b>Comment</b>	An optional comment about this use of the task in this workflow.
<b>Retry Execution</b>	Check <b>Retry Execution</b> to retry the task if it fails. The <b>Retry Count</b> and <b>Retry Frequency</b> controls appear.  <b>Retry Execution</b> is disabled for a workflow executed in a previous release of Cisco UCS Director. This is true even if retry is enabled for the workflow.
<b>Disable Rollback</b>	Check <b>Disable Rollback</b> to disable the task's rollback script. If you check this box, the task is unchecked in the <b>Rollback Service Request</b> dialog and cannot be rolled back; see <a href="#">Rolling Back a Service Request, on page 76</a> .  If the task does not support rollback, checking <b>Disable Rollback</b> has no effect.
<b>Retry Count</b>	The number of times (up to 5) to retry the task on failure.  The retry count is also assigned to the task's rollback. When a task is rolled back, the same number of retries are attempted as with the original script.  Rollback settings apply only for workflows executed with retry options in the current release of Cisco UCS Director.

Name	Description
<b>Retry Frequency</b>	A comma-separated list of integers specifying how many seconds to wait between retries. For example, if you selected a retry count of three, enter <b>1, 10, 60</b> to retry after one, 10, or 60 seconds, respectively.

**Step 8** Click **Next**.

**Step 9** On the **User Input Mapping** screen, complete the following fields:

Name	Description
<b>Manage Workflow User Inputs</b>	Click <b>Manage Workflow User Inputs</b> to add to or edit the workflow user inputs.
<b>Task Input</b>	Each task input is displayed as a heading on the remainder of this page. The input is labeled ( <b>Mandatory</b> ) if it required.
<b>Map to User Input</b>	Check <b>Map to User Input</b> to populate the task input from a workflow user input at runtime.
<b>User Input</b>	This drop-down list appears if you check <b>Map to User Input</b> . Select the workflow user input to map to the task input.  <b>Note</b> You cannot map a multiline workflow input to a single-line task input. The multiline workflow input does not appear in the <b>User Input</b> drop-down when creating or editing a task. See <a href="#">Defining Workflow Input and Output, on page 20</a> .

**Step 10** Click **Next**.

**Step 11** Any task input that was not mapped to a user input cannot be entered at runtime. Instead, specify the input value now. On the **Task Inputs** screen, complete the following fields:

Name	Description
<b>Task Input</b>	Each task input that was not mapped to a user input on the previous page is displayed with an appropriate input control (text field, drop-down list, and so on). Use the input controls to specify the value. Optional inputs can be left blank.

**Step 12** Click **Next**.

**Step 13** On the **User Output Mapping** screen, complete the following fields:

Name	Description
<b>Task Output</b>	Each task output is displayed as a heading on this page. The output is labeled ( <b>Mandatory</b> ) if it required.

Name	Description
Map to User Output	Check <b>Map to User Output</b> to populate a workflow user output from the task output at runtime.
User Output	This drop-down list appears if you check <b>Map to User Output</b> . Select the workflow user output to map to the task output.

**Step 14**

Click **Submit**.

If this is the first task configured in this workflow, the **Workflow Designer** automatically connects it to the default tasks. If the task is not the first, you must connect it manually.

**What to do next**

Connect the task to the workflow.

## Connecting a Task to a Workflow

To be part of a workflow, a task must be connected to other tasks. You can insert a task anywhere in a workflow by dragging and dropping the flow arrows between tasks.

There are three default tasks in every workflow: **Start**, **Completed (Success)**, and **Completed (Failed)**. When you add the first task to a workflow, it automatically connects to the default tasks.

**Before you begin**

You have created a task in the **Workflow Designer**. The task is in the **Workflow Designer** work area but is not yet part of the workflow.

**Step 1**

Connect a task from the existing workflow to the new task. To connect the tasks:

- The existing task is already connected to a different task, perhaps the **Completed (Success)** task. Mouse over the connecting line.  
Two large arrowheads and a red **x** icon appear superimposed on the line.
- Mouse over the arrowhead attached to the top of the **Completed (Success)** task (or other task that your source task is connected to).
- Click and hold your left mouse button.
- Drag the cursor to the input (the colored square at the top) of the task to which you are connecting and release the mouse button. The colored arrow now connects the source task to your new task, indicating the flow of operation from one task to the other.

**Note** Alternatively, you can click inside the task for a dialog box containing choices for both **On Success** and **On Failure** results. Click the arrow next to either choice for a menu of tasks that you can connect to. Choose a task, and the connection is made automatically with no drag-and-drop.

**Note** Typically you connect a task's **On Success** result to the next task, and connect the **On Failure** result to the **Completed (Failed)** task. There are exceptions, however. For example, you could insert a task to do cleanup or error handling, in which case you would connect **On Failure** to the cleanup task.

The **Workflow Designer** does not permit you to connect a task's **On Success** drop-down to the **Completed (Failed)** task.

**Note** To remove a connection between tasks, mouse over the connection line until the red **x** icon appears over the connecting line, then click the icon.

**Step 2** Similarly, drag and drop from the **On Success** connector of the new task to **Completed (Success)** or to another existing task in the workflow.

**Step 3** Drag and drop from the **On failure** connector of the new task to **Completed (Failed)** or to another existing task in the workflow.

---

The task is connected into the workflow.



**Note** Click **Auto Layout** at the top right of the **Workflow Designer** to arrange the workflow tasks in the work area with uniform spacing and alignment.



**Note** To delete a task from the workflow, mouse over the workflow and click the red **Delete (X)** icon that appears.

### What to do next

Add more tasks if necessary to complete the workflow. When you are finished, validate the workflow. See [Validating a Workflow, on page 38](#).

## Cloning a Task

You can clone tasks directly within the **Workflow Designer**, enabling you to quickly create multiple instances of a task that are identical or nearly identical within the same workflow.

For example, you can create duplicate instances of a task that accept the same inputs but which are run based on a conditional statement. Or you can repeat a task with many inputs multiple times in a workflow, varying only one or two inputs each time.

In such cases, cloning the workflow saves you the effort of duplicating the inputs manually each time. You clone the workflow, make minor changes, and connect the new task to your workflow. The cloned task is initially identical to the original. You can then change any or all editable parameters in the new task.

Once created, cloned workflows work exactly as they would if they were created from scratch by dragging and dropping the task from the **Available Tasks** directory.

### Before you begin

Define a workflow and its inputs and outputs as described in the previous sections. Define one or more tasks in the **Workflow Designer**.

You must have Orchestration write privileges to clone a task.



---

**Note** Dynamic tasks cannot be cloned. Dynamic tasks are tasks that create objects and have variable numbers of inputs. The following are dynamic tasks:

- Execute Generic Api task
- Get Resource Group Objects
- Get Tenant Resource Objects

To add a dynamic task to a workflow, you must drag it from the task list and enter its parameters.

---

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Workflows**.

**Step 3** Choose a workflow.

**Step 4** Click **Workflow Designer**. The Cisco UCS Director Orchestrator **Workflow Designer** screen appears.

**Step 5** Define one or more tasks and add them to the workflow. See [Configuring a Task in a Workflow, on page 33](#) and [Connecting a Task to a Workflow, on page 36](#).

**Step 6** Hover over the upper right corner of the task you want to clone and click the **Clone** (the "double page") icon next to the **Delete** icon (the red X).  
The **Clone Task** page comes up.

**Step 7** On the **Clone Task** page, change the Task Name.

**Note** You must change the name of the new task. Every task in a workflow must have a name that is unique within the workflow.

**Step 8** Change any other parameters of the task that you want. See [Configuring a Task in a Workflow, on page 33](#).

**Step 9** Click through to the **User Output Mapping** page and click **Submit**.

**Step 10** Connect the task to the workflow. See [Connecting a Task to a Workflow, on page 36](#).

**Note** While cloning a task, if a task is having password as input, the password will not be cloned. You must manually enter the password.

---

## Validating a Workflow

Validating a workflow ensures that the inputs and outputs of its component tasks are connected with no major errors. It also checks workflow input rules for errors.

You can validate a workflow without leaving the **Workflow Designer**.

### Before you begin

Open a workflow in the **Workflow Designer**.

Click the **Validate Workflow** button.

Task connection errors are displayed in red above the affected tasks. If no errors are detected, a success message appears.

### What to do next

If connection errors are displayed in the **Workflow Designer**, use the **Workflow Validation** tool to resolve the issues. See [Resolving Workflow Validation Issues, on page 41](#).

## Example: Creating a Workflow

This simple example walks you through the creation of a workflow that power cycles (turns off and back on) a VM.

### Before you begin

You have a virtual machine (VM) in Cisco UCS Director.

**Step 1** Define the workflow. Do the following:

- a) Choose **Orchestration**.
- b) On the **Orchestration** page, click **Workflows**.
- c) Click **Add**.
- d) Complete the fields on the **Add Workflow Details** screen as follows:

Field	Entry
<b>Workflow Name</b>	Type <b>PowerCycleVM</b> .
<b>Description</b>	Type <b>Shut down and restart a VM</b> .
<b>Workflow Context list</b>	Select <b>Any</b> .
<b>Save as Compound Task</b>	Leave this unchecked.
<b>Always execute during System initialization</b>	Leave this unchecked.
<b>Place in New Folder</b>	Check <b>Place in New Folder</b> .
<b>Folder Name</b>	Type <b>Demo</b>
<b>Notify status of execution to initiating user</b>	Leave this unchecked.
<b>Email policy</b>	Choose <b>No email</b> .
<b>Version Label</b> text field	Leave the default value, <b>0</b> .

- e) Click **Next**.

**Step 2** Define an input to specify the name of the VM. Do the following:

- a) In the **Workflow User Inputs** screen, click **Add (+)** at the top of the list of values.
- b) Complete the fields on the **Add Entry To** screen as follows:

Field	Entry
<b>Input Label</b>	Type <b>VM Name</b> .
<b>Input Description</b>	Type <b>VM Name</b> .
<b>Optional</b>	Leave this unchecked.
<b>Input Type</b>	Click <b>Select ...</b> . The <b>Select</b> screen comes up.

- c) On the **Select** screen, type **vm selector** in the search box.
- d) Check the VM Selector entry.
- e) Click **Select**.
- f) Define an administrator (fixed) value for the VM name by completing the next set of fields as follows:

Field	Entry
<b>Admin Input</b>	Check <b>Admin Input</b> .
<b>Admin Input Filter</b> and <b>Admin Input List</b>	<b>Admin Input Filter</b> and <b>Admin Input List</b> disappear when you check <b>Admin Input</b> . Both of these enable the administrator to filter the choices available to the user when the workflow is run. Instead, for this example, the <b>Admin Input Value</b> supplies a single fixed value at runtime.
<b>Override Admin Input Value</b>	Leaving <b>Override Admin Input Value</b> unchecked hides the user's input control for this value at runtime. For this example, leave it unchecked.
<b>Admin Input Value</b>	Click <b>Select</b> to choose a value for the administrator input. The <b>Select</b> list comes up.  Click to select an existing VM instance, then click <b>Select</b> .

- g) Click **Next**.

**Step 3** On the **Add Input Rules** page, click **Next**. Do not define any workflow input processing rules.

**Step 4** Do not define an output. Click **Submit**.

The **Workflow Designer** appears.

**Step 5** Add a task to power off the VM by doing the following:

- a) In the **Available Tasks** search field, type **power**.  
The **Available Tasks** area now displays only tasks containing the string *power*.
- b) From the **Generic VM Tasks** folder, drag and drop the **VM Power Action** icon into the work area.  
The **Add Task** screen appears.
- c) On the **Task Information** screen, type *Power Off 1* in the **Task Name** field.
- d) Click **Next**.
- e) On the **User Input Mapping** screen, complete these fields:

Field	Entry
Map to User Input	Check <b>Map to User Input</b> . You use this screen to map workflow inputs to task inputs.
User Input	Since there is only one user input of the correct type (or of any type, in this case), the user input name that you created ( <i>VM Name</i> ) is already selected in the drop-down list.

- f) Click **Next**.
- g) In the **Task Inputs** screen's **VM Action** drop-down list, select **Power OFF**.
- h) Click **Next**.
- i) Click **Submit**.  
The task is automatically connected in the new workflow.

**Step 6** Add a task to power on the VM. The procedure is similar to the previous step, with the following differences:

- On the **Task Information** screen, type **Power On 1** in the **Task Name** text field.
  - In the **Task Inputs** screen's **VM Action** drop-down list, select **Power ON**.
- Note** You could create this workflow with a single **Reboot** task rather than powering off and back on. However, for purposes of this example go ahead and create the workflow with the two power management tasks.
- When you click **Submit**, the task does not automatically connect to the existing workflow. Instead, connect it by hand:
    - a. Mouse over the lower left corner of the **Power Off 1** task and click the red **x** icon to delete the **Power Off 1 On Success** connection.
    - b. Mouse over the lower left corner of the **Power Off 1** task again. The **On Success** cursor appears.
    - c. Drag and drop the **On Success** arrow to the input connector at the top of the **Power On 1** task.
    - d. Using the same drag and drop technique, connect the **Power On 1 On Success** arrow to the **Completed (Success)** task.
    - e. Similarly, connect the **Power On 1 > On Failure** drop-down to the **Completed (Failed)** task.

**Step 7** Validate the workflow by clicking **Validate Workflow**.

The system displays the message `This workflow is valid`, confirming that the workflow and task inputs and outputs are connected properly.

**Step 8** Click the **Close Screen** control in the upper right to leave the **Workflow Designer**.

## Resolving Workflow Validation Issues

Cisco UCS Director supplies a wizard-based tool to aid you in troubleshooting workflow validation errors.

### Before you begin

A workflow has failed validation in the **Workflow Designer**.

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Choose the workflow that failed validation.
  - Step 4** Click **Validate Workflow**.
  - Step 5** In the **Workflow Validation** dialog box, double click the description under **Resolution**.
  - Step 6** In the **Validation Errors** dialog box, view the summary for the issue detected. Complete the remaining prompts to resolve the validation error.
- 

## Changing Flow of Control in Workflows

Normally, a workflow executes as a linear series of tasks. However, Cisco UCS Director Orchestrator provides tasks that enable changes to the flow of control within a workflow. The following sections describe how to create these flow-of-control constructs:

- Loops.
- If-then branches.
- Conditional switches with an arbitrary number of branches.




---

**Note** Conditional tasks, including **Conditional Task** and **If Else**, are denoted in the **Workflow Designer** with a diamond icon in their upper left corner. See [The Workflow Designer, on page 15](#).

---

## Looping in Workflows

You can create a loop to perform a selected series of tasks multiple times. A loop can be configured to iterate one of two ways:

- By count—Iterate a specified number of times.
- By item—Iterate over a list of items.

### Before you begin

Open a workflow in which you want to include a loop in the **Workflow Designer**.

- 
- Step 1** In **Available Tasks** area in the **Workflow Designer**, open the **Procedural Tasks** folder.
  - Step 2** Drag the **Start Loop** task into the **Workflow Designer** work area.
  - Step 3** In the **Add Task** screen, complete the following fields:

Name	Description
Task Name	The name of the task.

Name	Description
Comment	Comments for the task.
Retry Execution	Check <b>Retry Execution</b> to retry later if the task fails.
Retry count	Choose the number of retry attempts.
Retry Frequency	Choose the duration between retry attempts.

**Step 4** Click **Next**.

**Step 5** In the **User Input Mapping** screen, click **Next**.

**Step 6** In the **Task Inputs** screen, complete the following fields:

Name	Description
List based iteration	Check <b>List based iteration</b> to perform the loop tasks on every item in a list. The following two fields appear.
Input for list-based iteration	Choose a list of input values for the list-based iteration.
User Input to assign iterated values	The user input for the list-based iteration. Create a workflow input with the same type as your list elements and map it to the <b>Start Loop</b> task here.
Count based iteration check box	Check <b>Count based iteration</b> to perform loop tasks based on a count. The following fields appear.
Number of times to loop	The number of times to perform the loop.
Starting Index	The starting value of a counter that can be passed as an output of the task. Enter any integer (including negatives). The value defaults to zero.
Step Increment	This value is added to the counter on every iteration. Enter any integer (including negatives). The value defaults to one.

**Step 7** On the **User Output Mapping** screen, click **Submit**.

**Step 8** From the **Available Tasks** area, drag the tasks that you want as part of the loop into the work area of the **Workflow Designer**.

**Step 9** Create links between the task icons in the loop using the **On Success** connector.

**Step 10** Create a link between the **Start** task icon (or whichever task is immediately before the loop) and the **Start Loop** task icon.

**Step 11** In the **Available Tasks** area in the **Workflow Designer**, click the **Procedural Tasks** folder.

**Step 12** Drag the **End Loop** task into the **Workflow Designer** work area.

**Step 13** Complete the remaining screens in the **Add Task (End Loop)** wizard.

**Step 14** Click **Submit**.

**Step 15** Connect the **On Success** connector of the last task in the loop to the **End Loop** task icon.

- Step 16** In the **End Loop** task icon, connect the **On Success** connector to **Completed (Success)** or to another task to execute after the loop.
- Step 17** In the **End Loop** task icon, connect the **On Failure** connection to the **Completed (Failed)** (or another task) task icon.

## Example: Looping in a Workflow

This example demonstrates how to repeat workflow tasks for elements in a list. You will modify the PowerCycleVM workflow to restart a list of VMs instead of a single VM.

### Before you begin

Create the example workflow as described in [Example: Creating a Workflow, on page 39](#).

- Step 1** Open the PowerCycleVM workflow.
- Choose **Orchestration**.
  - On the **Orchestration** page, click **Workflows**.
  - Locate and select the PowerCycleVM workflow you created in [Example: Creating a Workflow, on page 39](#).
  - Click on **Workflow Designer**.
- Step 2** Add the **Start Loop** task to the workflow.
- From the **Available Tasks** in the **Procedural Tasks** folder, drag and drop **Start Loop** onto the work area.
  - On the **Task Information** screen, complete the following fields:

Field	Entry
<b>Task Name</b>	Type <b>StartVMLoop</b> .
<b>Comment</b>	Type <b>Loop through a list of VMs</b> .
<b>Task Details</b>	Make a note of the message under <b>Task Details</b> regarding the task output <b>START_LOOP_OUTPUT</b> .

- Click **Next**.
- Replace the workflow input to accept a list of VMs.

On the **User Input Mapping** screen, click **Manage Workflow User Inputs**.

- Click the **Add** icon.
- In the **Add Entry To** dialog, complete the following fields:

Field	Entry
<b>Input Label</b>	Type <b>VM Names</b> .
<b>Input Description</b>	Type <b>List of VMs to restart</b> .
<b>Optional</b>	Leave <b>Optional</b> unchecked.

Field	Entry
Input Type	Click <b>Input Type</b> to access the <b>Select</b> list. Locate and select the user input type <b>multiVM</b> . Click <b>Select</b> .
Value Restrictions	Check <b>Admin Input</b> . Click <b>Select</b> . Choose several VMs , then click <b>Select</b> .

- g) Click **Submit**.
- h) On the **User Input Mapping** page, under the **Input for list based iteration** heading, check **Map to User Input**. The **User Input** list appears.
- i) In the **User Input** list, select **VM Names**.
- j) Click **Next**.
- k) Check **List based iteration**. The **User input to assign iterated values** list appears.

**Note** This field is a feature of the **Start Loop** task. Since the output variable *START\_LOOP\_OUTPUT* is of type **generic text**, it cannot (usually) be mapped to the input of the tasks inside the loop. Instead, the **Start Loop** task assigns each item of the input list to the variable as it iterates.

- l) Select **VM Name** in the **User input to assign iterated values** list.

**Note** There is nothing special about the **VM Name** workflow input variable; you are reusing it since you no longer need it as input to the workflow. (The workflow input is now **VMNames**, which is a list of VMs.)

- m) Click **Next**.
- n) There is no output mapping, so click **Submit** to save the task.
- o) Drag and drop to connect the **Start** task to the **StartVMLoop** task. The **Workflow Designer** connects the **StartVMLoop > On Success** dropdown to the **Power Off 1** task (because it was the previous target of the **Start** task).
- p) Drag and drop to connect the **StartVMLoop > On failure** dropdown to the **Completed (Failed)** task.

**Step 3**

The **End Loop** task is just a placeholder with no inputs or outputs. To add the **End Loop** task to the workflow, do the following:

- a) Drag and drop the **End Loop** task from the **Procedural Task** folder onto the work area.
- b) On the Task Information screen, complete the following fields:

Field	Entry
Task Name	Type <b>EndVMLoop</b> .
Comment	Type <b>End of VM loop</b> .

- c) Click **Next**.
- d) On the **User Input Mapping** screen, click **Next**.
- e) On the **Task Inputs** screen, click **Next**.
- f) On the **User Output Mapping** screen, click **Next**.
- g) Drag and drop to connect the **Power On 1** task to the **EndVMLoop** task.

h) Drag and drop to connect the **EndVMLoop > On failure** dropdown to the **Completed (Failed)** task.

**Step 4** Click **Close** to close the **Workflow Designer**.

---

#### What to do next

Execute the task to restart the VMs specified in the workflow.

## Adding an If-Then Branch to a Workflow

You can create a two-way branch in a workflow using an if-then construct.

#### Before you begin

You have a workflow open in the **Workflow Designer**.

**Step 1** In the **Available Tasks** area of the **Workflow Designer**, open the **Procedural Tasks** folder.

**Step 2** Drag the **If Else** task into the **Workflow Designer** work area.

**Step 3** On the **Add Task** screen, complete the following fields:

Name	Description
<b>Task Name</b>	The name of the task.
<b>Comment field</b>	Comments for the task.
<b>Retry Execution</b>	Check <b>Retry Execution</b> to retry later if the task fails.
<b>Retry count</b>	Choose the number of retry attempts.
<b>Retry Frequency</b>	Choose the period between retry attempts.

**Step 4** Click **Next**.

**Step 5** On the **User Input Mapping** screen, click **Next**.

**Step 6** On the **Task Inputs** screen, complete the following fields:

Name	Description
<b>Specify the condition</b>	<p>Enter the condition evaluated to determine the course of the workflow. The flow of control depends on whether the condition evaluates to True or to False.</p> <p>The conditions that you define must contain only the following operators:  <b>==, !=, &lt;, &lt;=, &gt;, &gt;=,   , &amp;&amp;, contains, startsWith, endsWith</b></p> <p>Following are some example conditions:</p> <ul style="list-style-type: none"> <li>• <code>TaskName.OUTPUT_ATTRIBUTE_NAME == "2"</code></li> <li>• <code>WORKFLOW_USERINPUT_LABEL_NAME contains "xyz"</code></li> <li>• <code>TaskName.OUTPUT_ATTRIBUTE_NAME == WORKFLOW_USERINPUT_LABEL_NAME    WORKFLOW_USERINPUT_LABEL_NAME != "123"</code></li> </ul>

**Step 7** On the **User Output Mapping** screen, click **Submit**.

**Step 8** In the **If-Else** task icon, click and drag the **True** connector and the **False** connector to other task icons.

**Step 9** In the **If-Else** task icon, click and drag the **On Failure** connector to the **Completed (Failed)** task icon (or another task icon).

**Note** Do not confuse the **False** and **On Failure** connectors. The **False** connector defines the path if the condition statement evaluates to false. The **On Failure** connector defines the path for failure of the task, for example if the task cannot evaluate the condition statement.

## Using Conditional Branching in a Workflow

You can create a multiple-path branch in a workflow using a conditional construct. A single conditional task can have any number of branches.

### Before you begin

You have a workflow open in the **Workflow Designer**.

**Step 1** In the **Available Tasks** area of the **Workflow Designer**, expand the **Procedural Tasks** folder.

**Step 2** Drag the **Conditional** task into the **Workflow Designer** work area.

**Step 3** In the **Add Task** screen, complete the following fields:

Name	Description
<b>Task Name</b>	The name of the task.
<b>Comment</b>	Comments for the task.
<b>Retry Execution</b>	Check <b>Retry Execution</b> to retry later if the task fails.
<b>Retry count</b>	Choose the number of retry attempts.

Name	Description
Retry Frequency	Choose the duration between retry attempts.

**Step 4** Click **Next**.

**Step 5** In the **User Input Mapping** screen, click **Next**.

**Step 6** In the **Task Inputs** screen, add a condition statement for each execution path of the workflow. For example, if your workflow has three possible execution paths, add three entries into the conditions table, one entry for each condition. Click the default entry in the table to specify the condition when the workflow takes the default execution path.

**Note** The condition statements do not have to be mutually exclusive. At run time, the statements are evaluated in the order they are listed. Execution continues with the path of the first statement to evaluate to True.

**Step 7** In the **Add Entry to** screen, complete the following fields:

Name	Description
Label field	The label for the condition
Condition field	<p>Enter the condition to be evaluated to determine the course of the workflow. The conditions that you define must contain the following operators:</p> <p><b>==, !=, &lt;, &lt;=, &gt;, &gt;=,   , &amp;&amp;, contains, startsWith, endsWith</b></p> <p>Following are some examples that you can use while specifying conditions:</p> <ul style="list-style-type: none"> <li>• TaskName.OUTPUT ATTRIBUTE NAME=="2"</li> <li>• WORKFLOW USERINPUT LABEL NAME contains "xyz"</li> <li>• TaskName.OUTPUT ATTRIBUTE NAME == WORKFLOW USERINPUT LABEL NAME    WORKFLOW USERINPUT LABEL NAME != "123"</li> </ul>

Click **Submit**.

**Step 8** In the **User Output Mapping** screen, click **Submit**.

**Step 9** From the **Available Tasks** area, click and drag the tasks that you want to add to the various execution paths of the workflow.

**Step 10** Create a link between the **Start** task icon (or whichever task is immediately before the branch) and the **Conditional task** icon.

**Step 11** For each condition in the conditional task, create a link between the connector in the **Conditional Task** icon to the next task for the condition's execution path. For example, link the **default** connector in the **Conditional task** icon to the task you want to execute if none of the other conditions are met.

**Step 12** Link the **On failure** connector to the **Completed (Failed)** (or another task) task icon.

# Creating Approvals

To require approval of a user, you add an approval task to the workflow.

### Before you begin

Open a workflow in the **Workflow Designer**.

**Step 1** In the **Workflow Designer Available Tasks** window, open **Cloupia Tasks > General Tasks**.

**Step 2** Drag the **User Approval** task onto the work area.

**Step 3** Configure the task. See [Configuring a Task in a Workflow, on page 33](#). Complete the following fields:

Name	Description
<b>Task Name</b>	The task name.
<b>Comment</b>	A comment about the approval task (Optional.)
<b>Retry Execution</b>	Leave this unchecked. If the administrator rejects the approval, the workflow stops with the approval task in a failed state. If you enable execution retries, the workflow presents the administrator with an approval multiple times. This is probably not the behavior you want from the workflow.

**Step 4** Click **Next**.

**Step 5** In the **User Input Mappings** click **Next**. You define all the task inputs in the **Task Inputs** window.

**Step 6** In the Task Inputs window, complete the following fields:.

**Step 7**

Name	Description
<b>User ID list</b>	From the <b>User ID</b> list, choose one or more users whose approval is required to execute this workflow. Then click <b>Submit</b> .
<b>Approval required from all the users</b>	Check <b>Approval required from all the users</b> if running the workflow requires the approval of all the selected users. Leave unchecked if approval of any one of the selected users is sufficient.
<b>Number of approval request Reminders</b>	Enter the number of times to remind the user to approve this workflow. Entering 0 causes the reminder to be sent until the workflow approval has been approved or rejected.
<b>Reminder Interval (Hours)</b>	Enter the number of hours between approval reminders.
<b>Use Approval Email Template</b>	Check <b>Use Approval Email Template</b> to select a template for the approval emails. Leave unchecked to use a default Approval template.

Name	Description
<b>Select Email Template</b>	<p>Visible if <b>Use Approval Email Template</b> is checked. Select an email template to use.</p> <p>You can add, delete, and update email templates. See <a href="#">Working with Email Templates in the Cisco UCS Director Administration Guide</a>.</p> <p><b>Note</b> The <b>Select Email Template</b> list contains all the email templates, including those for email tasks other than approvals. If a selected email template causes an error when the workflow runs, the task uses the default Approval template instead.</p>
<b>Subject</b>	Visible if <b>Use Approval Email Template</b> is checked. The email subject line is displayed. You can edit this text.
<b>Body</b>	Visible if <b>Use Approval Email Template</b> is checked. The HTML text to output the email body text is displayed. You can edit this text.
<b>Use Approval Email Template for Rollback</b>	Check <b>Use Approval email template during Rollback</b> to select a template for emails requesting approval for rollback of the workflow. Leave unchecked to use a default Approval template.
<b>Select Email Template</b>	<p>Visible if <b>Use Approval Email Template during Rollback</b> is checked. Select an email template to use.</p> <p>You can add, delete, and update email templates. See <a href="#">Working with Email Templates in the Cisco UCS Director Administration Guide</a>.</p> <p><b>Note</b> The <b>Select Email Template</b> list contains all the email templates, including those for email tasks other than approvals. If a selected email template causes an error when the workflow runs, the task uses the default Approval template instead.</p>
<b>Subject</b>	Visible if <b>Use Approval Email Template during Rollback</b> is checked. The email subject line is displayed. You can edit this text.
<b>Body</b>	Visible if <b>Use Approval Email Template during Rollback</b> is checked. The HTML text to output the email body text is displayed. You can edit this text.

Name	Description
<b>Reset to Default</b>	<p>Visible beneath the <b>Body</b> text area in either of <b>Use Approval Email Template</b> or <b>Use Approval Email Template for Rollback</b>. <b>Reset to Default</b> has two effects:</p> <ul style="list-style-type: none"> <li>• It immediately discards any changes you have made to the <b>Subject</b> and <b>Body</b> text and restores the selected template to its system default.</li> <li>• It ensures that the system supplied version of the selected template is used for the approval email in the workflow you are editing.</li> </ul> <p>To save edits to a template, make changes in the <b>Subject</b> or <b>Body</b> text areas and be careful <i>not</i> to check <b>Reset to Default</b>.</p>

**Step 8** Click **Next**.

**Step 9** In the **User Output Mapping** window, click **Submit**.

## Locking Resources

You can lock a named resource to prevent other tasks from accessing that resource. To lock a resource, you acquire a lock on a resource name in a workflow and release the lock later in the same workflow. While the workflow holds the lock, other service requests (Workflows) block until the workflow that is currently holding the resource lock releases the lock.



**Note** The resource name is a generic character string with no programmatic connection to the resource being accessed. The **Acquire Lock on a Named Resource** task blocks workflow execution based only on the name of the resource that you specify. For purposes of locking, you can name a resource anything, but we recommend that you use the name of the resource as shown in Cisco UCS Director. This ensures that the name is unique, can be looked up easily, and is difficult to mistakenly assign to the wrong resource.

A workflow (within a single service request) can acquire a lock multiple times (this is called *reentrant locking*). If a workflow task has acquired a lock, then a successive attempt by any task in the workflow to acquire the same lock also succeeds. A lock acquired multiple times within a workflow must be released the same number of times before the resource is free.

A compound task (see [Creating a Compound Task, on page 101](#)) always succeeds in acquiring a lock if that lock is already held by its parent workflow.

When rolling back a service request that contains resource locking tasks, the roles of the `AcquireLock` and `ReleaseLock` tasks are reversed. This is to protect the resource when the rollback executes the workflow's tasks in reverse order. For more information see the `AcquireLock` and `ReleaseLock` entries in the Cisco UCS Director Orchestrator Task Library.

**Before you begin**

Open a workflow containing a reference to a resource that you want to lock.

- Step 1** In the **Available Tasks** area in the **Workflow Designer**, open the **Procedural Tasks** folder.
- Step 2** Drag the **Acquire Lock on a Named Resource** task into the **Workflow Designer** work area.
- Step 3** In the **Add Task** screen, complete the following fields:

Name	Description
Task Name	The name of the task.
Comment	Comments for the task.
Retry Execution	<p>There is no need to check <b>Retry Execution</b> when acquiring a lock. The task blocks until the named resource is available.</p> <p>If the named resource is not locked, the task acquires a lock on the name, regardless of whether the resource actually exists.</p>
Disable Roll Back	Check to prevent rollback of this task in a failed service request.

- Step 4** Click **Next**.
- Step 5** On the **User Input Mapping** screen, click **Next**.
- Step 6** On the **Task Inputs** screen, enter the name of the resource you want to lock.

**Note** The resource lock is based only on the name of the resource, which is a generic character string. It is vital that the name be exactly the same everywhere that the resource is referenced. The **Acquire Lock on a Named Resource** task does not have any connection to the resource being accessed; it simply controls access to the resource name that you specify. Therefore, misspelling the resource name in the **Acquire Lock on a Named Resource** task, for example, fails to lock the intended resource.

- Step 7** On the **User Output Mapping** screen, click **Submit**.
- Step 8** Create a link between the task immediately before the resource lock and the **Acquire Lock on a Named Resource** task icon.
- Step 9** From the **Available Tasks** area, drag the task that you want within the duration of the resource lock into the work area of the **Workflow Designer**.
- Step 10** Complete the **Add Task** wizard for the new task.
- Step 11** Link the task within the lock duration into the workflow using the **On Success** connector.
- Step 12** Repeat steps 9 to 11 if there are more tasks that you want to include in the duration of the resource lock.
- Step 13** In the **Available Tasks** area in the **Workflow Designer**, click the **Procedural Tasks** folder.
- Step 14** Drag the **Release Lock on a Named Resource** task into the **Workflow Designer** work area.

**Note** When a service request ends, whether successful or not, its workflow automatically releases all locks held by the service request. We recommend that you nonetheless explicitly release all locks as soon as they are no longer needed.

- Step 15** Complete the **Task Information** screen as you did in step 3.
- Step 16** Click past the **User Input Mapping** screen.
- Step 17** On the **Task Inputs** screen, enter the name of the locked resource.
- Step 18** Click **Next**.
- Step 19** Click **Submit**.
- Step 20** Connect the **On Success** connector of the last task added to the workflow to the **Release Lock on a Named Resource** task icon.
- Step 21** In the **Release Lock on a Named Resource** task icon, connect the **On Success** connector to the next task in the workflow.
- Step 22** In the **Release Lock on a Named Resource** task icon, connect the **On Failure** connection to the **Completed (Failed)** (or another task) task icon.

## File Uploads

You can upload files to Cisco UCS Director. The file is made available to the workflow task. To upload a file, you set an input to the **File Upload** input type.



**Note** Cisco UCS Director does not perform any validation of the contents of the file nor performs scans for viruses and malware. Extreme caution must be exercised to prevent virus infected files from being uploaded.

The file to upload must meet the following specifications:

- The file size is less than 100 MB.
- The file is in zip format.

If a task has a user input of type **File Upload**, then either a **File Upload** or **Generic Text** input can be mapped to it.

Uploaded files should not be deleted by tasks because the file may be used in other workflows or in subsequent tasks within the same workflow.

The behavior of the system with respect to the file is different depending on the source of the input:

### **Administrator Input or Task Input**

To upload the file at design time, that is, when you create the workflow, define an administrator input or a task input of type **File Upload**. Files uploaded at design time are stored permanently on the Cisco UCS Director instance and are available to be read by any task.

### **User Input or User Override of Admin Input**

To upload the file at the time that the service request is run, define a user input or enable a user to override an administrator input. Files uploaded at runtime are deleted when the workflow's service request ends. The file is available only to the workflow containing task that uploaded the file.

## Permanently Uploading an Admin-Selected File

To upload a file to Cisco UCS Director and make it available in a workflow, follow these steps:

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Add a new workflow. See [Creating a Workflow, on page 17](#).
- Step 4** Complete the fields on the **Add Workflow Details** page. See [Defining a Workflow, on page 17](#)
- Step 5** On the **Add User Input** page, add a new input. See [Defining Workflow Input and Output, on page 20](#).

Define an input as follows:

- a) At the top of the **Workflow User Inputs** table, click **Add**.
- b) Enter an **Input Label**.
- c) Enter a **Description** of the input (optional).
- d) Check **Optional** if the input is not required.
- e) Choose **fileUpload** as the input type.
- f) If the system requests that you validate the input, click **Validate**.
- g) Click **Admin Input**.
- h) In the File area, add a file to upload by doing one of the following:
  - Drag a file from your desktop to the **File** area.
  - Click **Select a File** and use the file browser to select a file.
- i) To enable a user to select a different file at runtime, click **Override Admin Input Value**. The filename appears at the top of the **File** area.
- j) Click **Submit**.
- k) Enter other inputs for the workflow, if any, and click **Next**.

**Note** The file must be a zipped (.zip) file. The zip file can contain any number of files with a total size of less than 100 MB. The filenames are then concatenated (file1;file2;file) and made available to the task. If your zip file contains multiple sub-directories, the directory structure will be flattened when the file is uploaded.

- Step 6** On the **Add User Outputs** page, define the workflow's outputs if necessary. See [Defining Workflow Input and Output, on page 20](#).
- 

### What to do next

Continue to construct your workflow. See [Connecting a Task to a Workflow, on page 36](#). The value of the workflow `File Upload` input that you defined is the path to the file on Cisco UCS Director. The pathname provided on Cisco UCS Director is a directory path ending in your filename combined with a date and time stamp. You can access the file from within any workflow using that path name.

If the zip file contains more than one file, the value of the `File Upload` input is a semicolon-separated list of the filenames.

## Temporarily Uploading a User-Selected File

To give a workflow access to a file that is selected by a user at runtime, follow these steps:

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Add a new workflow. See [Creating a Workflow, on page 17](#).
- Step 4** Complete the fields on the **Add Workflow Details** page. See [Defining a Workflow, on page 17](#)
- Step 5** On the **Add User Input** page, add a new input. See [Defining Workflow Input and Output, on page 20](#).
- Define an input as follows:
- At the top of the **Workflow User Inputs** table, click **Add**.
  - Enter an **Input Label**.
  - Enter a **Description** of the input (optional).
  - Check **Optional** if the input is not required.
  - Choose **fileUpload** as the input type.
  - Do not* click **Admin Input**.
  - If the system requests that you validate the input, click **Validate**.
  - Click **Submit**.
  - Enter other inputs for the workflow, if any, and click **Next**.
- Step 6** On the **Add User Outputs** page, define the workflow's outputs if necessary. See [Defining Workflow Input and Output, on page 20](#).
- Note** The file must be a zipped (.zip) file. The zip file can contain any number of files with a total size of less than 100 MB. The filenames are then concatenated (file1;file2;file) and made available to the task. If your zip file contains multiple sub-directories, the directory structure will be flattened when the file is uploaded.

---

### What to do next

Continue to construct your workflow. See [Connecting a Task to a Workflow, on page 36](#). At runtime, a user must specify a filepath to upload from their workstation. The file is uploaded to Cisco UCS Director and the workflow input becomes the local pathname of the file, assigned in the same manner as if it were an admin input. See [Permanently Uploading an Admin-Selected File, on page 53](#). The workflow can then access the file until the service request ends. When the service request ends, the file is deleted from Cisco UCS Director.





## CHAPTER 5

# Executing Workflows

---

This chapter contains the following sections:

- [Workflow Execution](#), on page 57
- [Pre- and Post-Provisioning Workflows](#), on page 58
- [Executing a Workflow](#), on page 58
- [Example: Executing a Workflow](#), on page 59
- [Example: Updating the Default Template for Sending Email through Template](#), on page 60
- [Creating a Workflow Trigger](#), on page 61
- [Example: Creating a Trigger to Execute a Workflow](#), on page 63
- [Creating a Trigger Template](#), on page 66
- [Scheduling a Service Request](#), on page 68
- [Example: Scheduling Workflow Execution](#), on page 69
- [Creating a Context Workflow Mapping](#), on page 70
- [Using VM Action Policies](#), on page 72

## Workflow Execution

When you execute a workflow, a service request is created. You can execute a workflow directly, schedule a workflow to run later, or create a trigger to run a workflow when certain conditions are met. You can also create a policy to execute workflows on VMs.

Each of these options is briefly outlined here.

- *Execute a Workflow Directly* - You can execute a selected workflow immediately. Navigate to **Policies > Orchestration > Workflows** or open the **Workflow Designer**, then choose **Execute Now**.
- *Schedule a Service Request to be Executed Later* - You can schedule execution of a selected workflow. Navigate to **Policies > Orchestration > Workflows**, then choose **Schedule**.
- *Trigger Conditional Workflow Execution* - You can create a trigger to execute a workflow when a set of conditions is met. Navigate to **Policies > Orchestration > Workflows**, then choose **Triggers**.
- *Create an Action Policy* - You can create a policy to execute workflows on VMs within a virtual data center (vDC). Navigate to **Policies > Orchestration > Workflows**, choose **User VM Action Policy**, then add the policy to a virtual data center (vDC).

Detailed instructions and examples are given in the following sections.

## Pre- and Post-Provisioning Workflows

When you publish a standard service request in a catalog, you can associate workflows to run before (pre-provisioning) and after (post-provisioning) the VM provisioning. See "Managing Catalogs" in the [Cisco UCD Director Administration Guide](#). The following features and rules apply to pre- and post-provisioning workflows.

- Pre- and post-provisioning workflows must be saved as compound tasks. See [Creating a Compound Task, on page 101](#).
- If pre-provisioning fails, the service request stops and the provisioning does not occur.
- Similarly, if the service request fails during provisioning, post-provisioning is not executed.
- No inputs are explicitly passed between the pre-provisioning, provisioning, and post-provisioning workflows.
- The tasks within the pre- and post-provisioning workflows are expanded as they are run, and their individual tasks are visible in the service request. This is in contrast to running a "normal" workflow containing a compound task, where the compound task is not expanded.

## Executing a Workflow

You can execute a workflow immediately from either the **Workflows** page or the **Workflow Designer**.

- 
- Step 1** Choose the workflow you want to execute by navigating to it on the **Workflows** page (it might be in a directory or subdirectory).  
Optionally, open the workflow in the **Workflow Designer** by clicking **Workflow Designer**.
- Step 2** Click the **Execute Workflow** action.
- Step 3** On the **Executing Workflow** screen, choose the version of the workflow you want to run.
- Step 4** If the workflow has the **Load Inputs from Previous Service Request** option enabled, you can optionally fill the input fields by duplicating the inputs from a previous service request:
- From the **Select an SR to Load Inputs** pull-down list, choose a service request from which to copy input values. Only the last 10 service requests appear in the list.
- All of the input fields are filled with the corresponding input values from the selected SR, with the following exceptions:
- Password fields are left blank.
  - Form Managed lists remain unpopulated.
- Step 5** Complete the input fields on the page. You can change any input field, including inputs that were populated by checking **Load inputs from previous SR**.
- The input fields show the workflow inputs that you are allowed to change as a user.
- Note** Workflow inputs can be defined with values (*admin inputs*) that cannot be overridden at runtime. See [Defining Workflow Input and Output, on page 20](#). Admin inputs are not displayed in the **Executing Workflow** dialog.
- If an input is mandatory and has no default input, you must provide a user input.

If workflow input rules are active, values you supply to input fields might cause other inputs to appear or disappear. See [Dynamic Workflow Inputs, on page 22](#).

**Note** Workflow input rules affect only the visibility of input fields, not their values. If you loaded inputs from a previous service request, all inputs, including hidden fields, contain the values copied from the selected SR.

**Step 6** Click **Submit**.  
The **Service Request Submit Status** screen appears.

---

#### What to do next

On the **Service Request Submit Status** screen, click **Show Detail Status** to see the progress of the service request.

## Example: Executing a Workflow

This example demonstrates executing a workflow directly.

#### Before you begin

Create the example workflow as described in [Example: Creating a Workflow, on page 39](#).

---

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Locate and select the PowerCycleVM workflow you created in [Example: Creating a Workflow, on page 39](#).
- Step 4** On the **Executing Workflow** screen, click **Submit**.
- Step 5** On the **Service Request Submit Status** screen, click **Show Detail Status**.
- Step 6** On the **Service Request** screen, click **Workflow Status**.  
On the status page is a graphic of the service request's progress.
- Step 7** Click **Log** to view the service request log.
- Note** Click **Refresh** to view the latest updates to the log.
- 

#### What to do next

Click **Workflow Status** and record the **Request ID** from the report. You can view the status of a service request by ID at any time. See [Viewing Service Requests, on page 75](#).

Click **Close** to leave the **Service Request** screen.

# Example: Updating the Default Template for Sending Email through Template

This example demonstrates updating the default template for sending emails through template.

## Before you begin

Make sure to configure the mail setup.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Create a new workflow. For more information, see [Example: Creating a Workflow, on page 39](#).
- Step 4** Add **Send Email Through Template** workflow task in your workflow and on the **Task Input** screen, specify the email addresses and select an email template, for example: Password Modify Notification.
- Step 5** Update the default template script in the body as per the sample body script.

The following displays the sample body script of Password Modify Notification template.

```
<Cloupia:ReportScript>
importClass (com.cloupia.model.cIM.ServiceRequest);
importClass (com.cloupia.service.cIM.inframgr.InfraPersistenceUtil);
importClass (com.cloupia.service.cIM.inframgr.workflowmgr.WorkFlowPersistenceUtil);
importPackage (java.lang);
importClass (com.cloupia.lib.util.time.SystemTimeFormat);
</Cloupia:ReportScript>

<!-- Generates email for Password Modification -->

<body bgcolor=#ffffff>

<Cloupia:ReportScript>
var loginName = "${INPUT_USER}";
var time = "";
var passwordChangedBy = "";
var sr = InfraPersistenceUtil.getServiceRequest (${SR_ID});
if (sr != null ) {
    time = SystemTimeFormat.formatTime (System.currentTimeMillis());
    passwordChangedBy = sr.getInitiatingUser();
}
print (rbutil.getString ("System.string.ET.Password.user"));
print (" "+loginName);
out.print (rbutil.getString ("System.string.ET.Password.Modify"));
print (" "+passwordChangedBy);
print (rbutil.getString ("System.string.ET.Password.user.time") + " "+ time);

</Cloupia:ReportScript>

</body>
```

where `${INPUT_USER}` is the login user name for whom the password is modified which can be workflow user input or output of the previous task.

- Step 6** Click **Next** and provide user output mappings if needed.
- Step 7** Click **Submit**.

**Step 8** Validate the workflow and execute it to check if the emails are sent.

## Creating a Workflow Trigger

You can define a trigger to execute a workflow when specific conditions are met. These conditions are typically the states of various system components.



**Note** If you want a workflow to execute at a predetermined time, you can create a schedule for workflow execution. See [Scheduling a Service Request, on page 68](#).

When you enable a trigger, Cisco UCS Director monitors the system state and when the trigger conditions are met executes the workflow you specify.

To create a workflow trigger, follow these steps:

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Triggers**.

**Step 3** Click **Add**.

**Step 4** On the **Add Trigger: Trigger Information** screen, complete the following fields:

Name	Description
<b>Enable Template</b>	Check this check box to create a trigger template.
<b>Trigger Name</b>	A unique name for the trigger.
<b>Is Enabled</b>	Enables the trigger. (You can create and save a trigger, then enable it later.)
<b>Description</b>	A description of the trigger.
<b>Frequency</b>	<p>The period between checks for the trigger rule. The choices for this period span three minutes to one month.</p> <p><b>Note</b> Internally, Cisco UCS Director polls for triggers every two minutes. Therefore, the actual period might be up to two minutes longer than specified. This can be significant for small trigger periods.</p> <p>For example, with <b>Frequency</b> set to three minutes, the trigger might activate up to three + two = five minutes after the trigger condition is satisfied.</p>

Name	Description
<b>Trigger Type</b>	Choose the trigger type: <ul style="list-style-type: none"> <li>• <b>Stateful</b>—The last trigger state is recorded and actions are executed when there is a change in the trigger state. (The trigger state is Active when the trigger conditions are met, and Clear otherwise.) You choose workflows for both trigger state transitions: one for when the trigger state changes from Active to Clear, and one for when the trigger state changes from Clear to Active. The trigger state is checked at the frequency specified in the Frequency drop-down list.</li> <li>• <b>Stateless</b>—The trigger is executed anytime the trigger conditions are met, at a frequency specified in the Frequency drop-down list.</li> </ul>

During execution of a triggered service request, additional requests by the same trigger are ignored. Only after the triggered service request has completed does the next polling event, as defined by the Frequency parameter, trigger another service request.

For example: Stateful Trigger T has a Frequency parameter of 5 minutes. T polls parameter P at time 13:00 and finds that P meets the trigger criterion, so T executes a service request SR that runs for 8 minutes. At 13:05, T polls again but, because SR is still running, T does not execute another service request, regardless of the state of P. At 13:08, SR ends. When T polls again at 13:10, it executes a new service request if the criterion for P is met.

**Step 5** Click **Next**.

**Step 6** Specify the trigger conditions.

- a) On the **Add Trigger: Specify Conditions** screen, do the following:
  - Expand the **Conditions** list.
- b) In the **Add Entry to Monitor** dialog box, complete the following fields:

Name	Description
<b>Type of Object to Monitor</b>	Choose which type of object to monitor with this condition.
<b>Object</b>	Choose the particular object to monitor. Entries in this list vary depending on the type of object. You might need to specify additional information (a pod, for example) to filter for the object.
<b>Parameter</b>	Choose an operational parameter to monitor. Entries in this list vary depending on the object.
<b>Operation</b>	Choose the relational operation for the trigger comparison. Entries in this list vary depending on the possible states of the parameter.

Name	Description
Value	Choose the value with which to compare the parameter. When the relation defined by Parameter, Operation, and Value is true, the trigger state is Active; otherwise, the trigger state is Clear.

- c) Click **Submit**.
- d) Repeat the previous two steps to add more trigger conditions.
- e) Choose one of the two options in the **Trigger When** drop-down list:
  - Choose **All Condition(s) Satisfied** to specify that the trigger state is true only when all of the trigger conditions are met.
  - Choose **Any Condition(s) Satisfied** to specify that the trigger state is true if one or more of the trigger conditions are met.

**Step 7** Click **Next**.

**Step 8** On the **Add Trigger: Specify Workflow** screen, choose the workflow or workflows to execute on the trigger:

- a) In the **Select Workflow** drop-down list under **When Trigger State Becomes Active**, choose a workflow. This workflow is executed:
  - Whenever the trigger state is Active (polled at a frequency specified in the **Frequency** drop-down list), if the trigger is stateless.
  - If and only if the trigger state has changed from Clear to Active since last checked one period ago (as specified in the **Frequency** drop-down list), if the trigger is stateful.
- b) If you chose the stateful trigger type, in the **Select Workflow** drop-down list under **When Trigger State Becomes Clear**, choose a workflow. This workflow is executed if and only if the trigger state has changed from Active to Clear since last checked one period ago (as specified in the **Frequency** drop-down list).

**Step 9** Click **Next**.

**Step 10** On the **Add Trigger: Specify Workflow Inputs** screen, enter all the required inputs for the workflow, and any needed optional inputs.

**Note** You must enter all inputs that are needed by the workflow. There is no opportunity for user input in a triggered workflow.

**Step 11** Click **Submit**.

**What to do next**

You can enable or disable a trigger by selecting the trigger on the **Policies > Orchestration > Triggers** screen and checking or unchecking **Is Enabled**.

## Example: Creating a Trigger to Execute a Workflow

This example demonstrates creating a trigger to execute a workflow when a set of conditions are met.

**Step 1**

Create a workflow with a single task as follows:

- a) Create the workflow as described in the following table.

Name	Description
<b>Workflow Name</b>	Type <i>VM Power Event Notify</i> .
<b>Description</b>	Type <i>Sends an email when a VM changes power state</i> .
<b>Workflow Context</b>	Choose <i>Any</i> .
<b>Select Folder</b>	Choose <i>Orchestration Examples</i> .

- b) Create an input as described in the following table:

Name	Description
<b>Input Label</b>	Type <i>Admin Emails</i> .
<b>Input Description</b>	Type <i>Email addresses to send power event notifications</i> .
<b>Optional</b>	Leave unchecked.
<b>Input Type</b>	Choose <i>email_address_list</i> .
<b>Value Restrictions</b>	Do not check <b>Admin Input</b> .

- c) Do not define any user outputs.  
 d) In the **Workflow Designer**, add a **Send Email** task to the workflow.  
 e) Complete the task information as described in the following table:

Name	Description
<b>Task Information</b> screen	
<b>Task Name</b>	Type <i>VMPowerEventEmail</i> .
<b>User Input Mapping</b> screen	
<b>Email Addresses (Mandatory)</b>	Check <b>Map to User Input</b> .
<b>Task Inputs</b> screen	
<b>Subject</b>	This is the subject line of the notification email. Type <i>VM power event notification</i> .
<b>Body</b>	This is the body of the notification email. Type <i>One of the following VMs has had a power -on or power-off event: (list the VMs in your VMPowerCycle workflow)</i> .
<b>Value Restrictions</b>	Do not check <b>Admin Input</b> .

- f) On the **User Output Mapping** page, do not map any outputs.  
 g) Validate the workflow, then exit the **Workflow Designer**.

**Step 2** Click **Triggers**.

**Step 3** Choose **Add**.

**Step 4** On the **Trigger Information** screen, complete the following fields:

Name	Description
<b>Trigger Name</b>	Type <i>VMPowerEvent</i> .
<b>Is Enabled</b>	Leave this checked.
<b>Description</b>	Type <i>Executes VMPowerEventEmail when a power-on or power-off occurs on a monitored VM.</i>
<b>Frequency</b>	To test the trigger, you want a short time between polls. Choose <i>3 minutes</i> .
<b>Trigger Type</b>	Choose <i>Stateful</i> . This causes the trigger to activate when a condition changes between polls.

**Step 5** On the **Specify Conditions** screen, add some conditions.

a) Expand the **Conditions** list and add a condition using the following values:

Name	Description
<b>Type of Object to Monitor</b>	Choose <i>VM</i> .
<b>Object</b>	Choose a VM to be monitored.
<b>Parameter</b>	Choose <i>Power Status</i> .
<b>Operation</b>	Choose <i>Equals</i> .
<b>Value</b>	Choose <i>ON</i> .

b) Repeat the last step to add more VMs to be monitored. Each VM has its own condition.

c) In the **Trigger When** drop-down list, select **Any Condition(s) Satisfied**.

**Step 6** On the **Specify Workflow** screen, complete the fields using the following values:

Name	Description
<b>Maximum Invocations</b>	For demo purposes, choose <i>20</i> . You might choose <i>Unlimited</i> in an actual production application.
<b>When Trigger State Becomes Active</b>	In the <b>Select Workflow</b> drop-down list, choose <i>VMPowerEventNotify</i> . The trigger executes this workflow when any of your monitored VMs changes from <i>OFF</i> to <i>ON</i> as defined on the <b>Specify Conditions</b> page.  <b>Note</b> Items appear by serially assigned ID number in the drop-down list, so your workflow is at the bottom of list.

Name	Description
When Trigger State Becomes Clear	In the <b>Select Workflow</b> drop-down list, choose <i>VMPowerEventNotify</i> . The trigger executes this workflow when any of your monitored VMs changes from <i>ON</i> to <i>OFF</i> as defined on the <b>Specify Conditions</b> screen.

### What to do next

Test the trigger by turning on or off one of the monitored VMs. Make sure that the VM's state remains changed for at least three minutes. The trigger polls the state, so any change that is not visible at the time of polling (on and back off during the three minute wait interval, for example) does not activate the trigger.

## Creating a Trigger Template

To create a trigger template, follow these steps:

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Triggers**.
- Step 3** Click **Add**.
- Step 4** On the **Add Trigger: Trigger Information** screen, complete the following fields:

Name	Description
<b>Enable Template</b>	Check this check box to create a trigger template.
<b>Trigger Template Name</b>	A unique name for the trigger template.
<b>Description</b>	A description of the trigger template.
<b>Frequency</b>	<p>The period between checks for the trigger rule. The choices for this period span three minutes to one month.</p> <p><b>Note</b> Internally, Cisco UCS Director polls for triggers every two minutes. Therefore, the actual period might be up to two minutes longer than specified. This can be significant for small trigger periods.</p> <p>For example, with <b>Frequency</b> set to three minutes, the trigger might activate up to three + two = five minutes after the trigger condition is satisfied.</p>

Name	Description
<b>Trigger Type</b>	Choose the trigger type: <ul style="list-style-type: none"> <li>• <b>Stateful</b>—The last trigger state is recorded and actions are executed when there is a change in the trigger state. (The trigger state is Active when the trigger conditions are met, and Clear otherwise.) You choose workflows for both trigger state transitions: one for when the trigger state changes from Active to Clear, and one for when the trigger state changes from Clear to Active. The trigger state is checked at the frequency specified in the Frequency drop-down list.</li> <li>• <b>Stateless</b>—The trigger is executed anytime the trigger conditions are met, at a frequency specified in the Frequency drop-down list.</li> </ul>

**Step 5** Click **Next**.

**Step 6** Specify the trigger template conditions.

- a) On the **Add Trigger: Specify Conditions** screen, do the following:
  - Expand the **Conditions** list.
- b) In the **Add Entry to Monitor** dialog box, complete the following fields:

Name	Description
<b>Name</b>	Enter a unique condition name.
<b>Type of Object to Monitor</b>	Choose which type of object to monitor with this condition.
<b>Parameter</b>	Choose an operational parameter to monitor. Entries in this list vary depending on the object.
<b>Operation</b>	Choose the relational operation for the trigger comparison. Entries in this list vary depending on the possible states of the parameter.
<b>Value</b>	Choose the value with which to compare the parameter. When the relation defined by Parameter, Operation, and Value is true, the trigger state is Active; otherwise, the trigger state is Clear.

- c) Click **Submit**.
- d) Repeat the previous two steps to add more trigger template conditions.
- e) Choose one of the two options in the **Trigger When** drop-down list:
  - Choose **All Condition(s) Satisfied** to specify that the trigger state is true only when all of the trigger conditions are met.

- Choose **Any Condition(s) Satisfied** to specify that the trigger state is true if one or more of the trigger conditions are met.

**Step 7** Click **Next**.

**Step 8** On the **Add Trigger: Specify Workflow** screen, choose the workflow to execute on the trigger:

- Choose the maximum number of invocations from the drop-down list.
- In the **Select Workflow** drop-down list under **When Trigger State Becomes Active**, choose a workflow. This workflow is executed:
  - Whenever the trigger state is Active (polled at a frequency specified in the **Frequency** drop-down list), if the trigger is stateless.
  - If and only if the trigger state has changed from Clear to Active since last checked one period ago (as specified in the **Frequency** drop-down list), if the trigger is stateful.
- Under **When Trigger State Becomes Active**, check the **Pass Monitored Object** checkbox and choose a workflow input from the **Select Workflow Input** drop-down list.
- If you chose the stateful trigger type, in the **Select Workflow** drop-down list under **When Trigger State Becomes Clear**, choose a workflow. This workflow is executed if and only if the trigger state has changed from Active to Clear since last checked one period ago (as specified in the **Frequency** drop-down list).
- Under **When Trigger State Becomes Clear**, check the **Pass Monitored Object** checkbox and choose a workflow input from the **Select Workflow Input** drop-down list.

**Step 9** Click **Next**.

**Step 10** On the **Add Trigger: Specify Workflow Inputs** screen, enter all the required inputs for the workflow, and any needed optional inputs.

**Note** You must enter all inputs that are needed by the workflow. There is no opportunity for user input in a triggered workflow.

**Note** You must check the **Force** check box, if you want to edit and save the changes in the existing trigger template. By checking **Force**, the existing trigger template mappings used in the workflow will be lost.

**Step 11** Click **Submit**.

**Note** Trigger template does not support export, import and dynamic input rules.

## Scheduling a Service Request

You can schedule execution of a workflow for a specific time.

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Workflows**.

**Step 3** On the **Workflows** screen, choose the workflow that you want to schedule.

**Step 4** Click **Schedule**.

**Step 5** On the **Schedule Workflow** screen, complete the following fields:

Name	Description
<b>Recurrence Type</b>	Whether and at what frequency you want the workflow execution to recur. Choose one of the following: <ul style="list-style-type: none"> <li>• <b>No End</b>—Execute the workflow repeatedly without end at intervals defined by <b>Frequency Type</b> and <b>Frequency Interval</b>, starting at <b>Start Time</b>.</li> <li>• <b>Only once</b>—Execute the workflow only once, at <b>Start Time</b>.</li> <li>• <b>Fixed Number of Times</b>—Execute the workflow a number of times specified by <b>Repeat Count</b> at intervals defined by <b>Frequency Type</b> and <b>Frequency Interval</b>, starting at <b>Start Time</b>.</li> <li>• <b>End by Date</b>—Execute the workflow until the defined end time at intervals defined by <b>Frequency Type</b> and <b>Frequency Interval</b>, starting at <b>Start Time</b>.</li> </ul>
<b>Start Time</b>	The time at which to start the workflow execution (or the first of a series of recurring executions).
<b>Frequency Type</b>	The unit of the recurrence frequency interval. Choose hourly, daily, weekly, or monthly.
<b>Frequency Interval</b>	Choose the interval at which the workflow needs to be executed for the frequency that you chose. For example, if the <b>Frequency Interval</b> is four and the <b>Frequency Type</b> is daily, the workflow is executed every four days.
<b>User ID</b> (optional)	The user ID to run the workflow. The service request runs with the permissions of this user ID.

**Note** You must provide inputs with the workflow. Scheduled workflows do not accept user inputs.

**Note** The same workflow can be run from any number of workflow schedules.

**What to do next**

Click **Workflow Schedules** to view scheduled workflows.

On the **Workflow Schedules** screen, you can modify a scheduled workflow by choosing a workflow and clicking the **Edit** button. The **Modify Workflow Schedule** screen appears. Modify the fields described in the previous steps. The **Modify Workflow Schedule** screen is identical to the **Schedule Workflow** screen.

## Example: Scheduling Workflow Execution

This example demonstrates how to schedule a workflow for later execution.

**Before you begin**

Create the example workflow as described in [Example: Creating a Workflow, on page 39](#).

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Choose the *VMPowerCycle* workflow.
- Step 4** Click **Schedule**.
- Step 5** On the **Schedule Workflow** dialog, complete the following:

Name	Description
<b>Recurrence Type</b>	Choose <i>Only once</i> . You might choose other values in an actual production application.
<b>Start Time</b>	Choose today's date and set a time a few minutes later than your current time.
<b>Use ID</b>	Type the administrator ID you are signed in as.

- Step 6** Click **Submit**.
- 

**What to do next**

Click the **Workflow Schedules** tab. The scheduled workflow is listed.

Check the service request queue at the time you specified in the Start Time control. See [Viewing Service Requests, on page 75](#). A service request is generated for the workflow you specified.

## Creating a Context Workflow Mapping

A context workflow mapping consists of an action label mapped to a workflow on a page, such that clicking on the action label triggers the workflow.

The **Context Workflow Mapping** module allows you to dynamically add a context workflow mapping. In this way, you can customize the Cisco UCS Director UI by creating an action label for a page and assigning a workflow to that action label.

The following lists the mappings and the relevant navigation path for the action labels.

- **Initiator**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > Initiator**.
- **Initiator Group**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > Initiator Group**.
- **LUN**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > LUN**.
- **vFilers**—Navigate to **Organization > Physical Resources > vFilers**.

- **vFiler LUN**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > vFiler > LUN**.
- **vFiler Volume**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > vFiler > Volume**.
- **vFiler Initiator**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > vFiler > Initiator**.
- **vFiler Initiator Group**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > vFiler > Initiator Group**.
- **NetApp Volume**—Navigate to **Physical > Storage**, choose a Netapp 7 mode device, and then navigate to **Filers > Volume**.
- **UCS Server**—Navigate to **Organization > Physical Resources > Servers**.
- **UCS Service Profile**—Navigate to **Organization > Physical Resources > Service Profiles**.
- **Cloud**—Navigate to **HyperConverged > Virtual SAN > IaaS**.

Creating a context workflow mapping requires knowing the name of the page to which you are adding the action label. The name, and other metadata, of a page are displayed in an **Information** dialog that is available when you enable the **Developer Menu** in Cisco UCS Director.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Context Workflow Mapping**.
- Step 3** On the **Context Workflow Mapping** page, double-click the mapping you want to use.
- Step 4** On the **Edit Workflow Mappings** screen, click **Add Workflow**.
- Step 5** On the **Workflow 1** screen, complete the following:

Name	Description
<b>Selection Required</b>	If you check <b>Selection Required</b> , the <b>Workflow</b> drop-down list no longer has any values available. To make the workflow visible in the workflow drop-down list, execute <b>Context Workflow Mapping</b> on the <b>Workflow Designer</b> screen.  If you do not check this box, provide information for the remaining fields.
<b>Action Label</b>	Enter the name of the action label to connect to the workflow. It should be a unique name.
<b>Workflow</b> drop-down list	Select the workflow to be executed when the user clicks the action label in the report.  If you checked <b>Selection Required</b> , a new workflow is created. If you did not check <b>Selection Required</b> , choose a workflow listed in this drop-down list.

Name	Description
Authorized User Types	<p>On the <b>Select Items</b> screen, click the each of the user types you authorize. Your choices are:</p> <ul style="list-style-type: none"> <li>• Service End-User</li> <li>• Group Admin</li> <li>• System Admin</li> <li>• Operator/Other Administrator</li> </ul>

To add an additional **Workflow** step to this **Context Workflow Mapping**, click the **Add Workflow** button and complete the fields.

To delete a **Workflow** step from this **Context Workflow Mapping**, click **Delete Field** below the workflow you want to delete.

**Step 6** Click **Submit**.

## Using VM Action Policies

You can create a user VM action policy to apply to virtual data centers (VDCs). The policy contains workflows that can be run on VMs within that VDC.



**Note** Only one VM action policy can be assigned per VDC.

To create a VM action policy, do the following:

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **User VM Action Policy**.
- Step 3** Click **Add**.
- Step 4** On the **Add Policy: Create New Policy** screen, complete the following:

Name	Description
Policy Name	The policy name.
Policy Description	A description of the policy.
Select No of Actions	The number of actions defined by the policy. One action specifies a single workflow.

**Step 5** Click **Next**.

**Step 6** For each **VM Action** on the **Add Policy: Add VM Actions** screen, complete the following:

Name	Description
<b>Action Label</b>	The label for the VM action. The action is available when you select a VM that is part of a virtual data center (VDC) to which the action policy is applied.
<b>Workflow</b>	The workflow represented by this action.
<b>Authorized User Types</b>	The user types authorized to run the policy workflows.

**Step 7** Click **Submit**.

---

#### **What to do next**

After you create a User VM Action policy, as an administrator, you can choose this policy when you create a VDC. All VMs that belong to the VDC can then use the workflows specified in the action policy.





## CHAPTER 6

# Using Service Requests

---

This chapter contains the following sections:

- [Viewing Service Requests, on page 75](#)
- [Rolling Back a Service Request, on page 76](#)
- [Resubmitting a Service Request, on page 77](#)
- [Approving and Denying Service Requests, on page 78](#)
- [Deleting Service Requests, on page 79](#)
- [Locking a Service Request, on page 80](#)

## Viewing Service Requests

You can view details of a service request, including its progress through the workflow, the service request log, objects created by the service request, and input and output of the service request workflow. To view service request details:

---

**Step 1** Choose **Organizations > Service Requests**.

**Step 2** On the **Service Requests** page, click **Service Requests**.

**Step 3** Choose a service request.

**Step 4** Click **View Details**. The **Service Request Status** screen appears.

**Step 5** Choose among different aspects of the service request to examine:

- **Workflow Status**—Which tasks have succeeded or failed, which task (if any) is currently running, name of the user who initiated the request, user input values for the workflow, and user output values for the workflow.

**Note**

- User input and output details for Pre- and post-provisioning workflows are not displayed for a compound workflow, when you publish a standard service request in a catalog.
- User input and output details are not displayed for a compound workflow that is associated with a parent workflow. Only user input and output details for the parent service request is displayed.

- **Log**—Debug, information, warning, and error messages generated by the workflow tasks.

**Note**

You can export the log file by clicking **Export**. The file will be downloaded in your local system.

- **Objects Created and Modified**—Details about objects created by the workflow, including the owning tenant, the account, and other information.
- **Input/Output**—Input and output of each workflow task. See [Viewing Service Request Input and Output, on page 76](#).

**Step 6** Click **Close**.

## Viewing Service Request Input and Output

The **Input/Output** screen in the **Service Request** dialog displays a list of the values for each task in the workflow at the time of approval. There are four types of input or output that may be visible:

- Administrator input.
- Task input.
- Task output.
- Mapped input or output. In the case of a mapped input (or output), the name of the corresponding source (or target) output (or input) is shown in the **Input/Output** column. For example, a task input mapped to the EMAIL\_ADDRESSES output from task SendEmail\_94 would display SendEmail\_94.EMAIL\_ADDRESSES in the **Input/Output** column.



**Note** The input and output values for a service request are viewable even if the underlying workflow has been deleted. However, in this case the input values cannot be edited.

- Step 1** Choose **Organizations > Service Requests**.
- Step 2** On the **Service Requests** page, click **Service Requests**.
- Step 3** Click a service request. The **Input/Output** table appears.
- Step 4** Click **Input/Output**. The input and output values of all the tasks in a workflow are accessible from this screen. The types of outputs appear in the **Input/Output** column.
- Step 5** Click an arrow next to a folder icon in the left column to expand a task. The type, name, and value of each of the task's inputs and outputs are displayed in the table.

## Rolling Back a Service Request

You can undo all or part of an executed workflow by using the Service Request Rollback feature. See [Rollback, on page 10](#).

You can roll back most workflow tasks that executed successfully. Exceptions:

- Some tasks are designed such that they cannot be rolled back.
- Rollback can be disabled for a task within a particular workflow. See the following discussion.

Any changes made by tasks that are not selected for rollback (for example, creating virtual resources) are not undone (for example, the virtual resources are not removed). If you choose to roll back an entire successful service request, all its workflow tasks are rolled back (with the previously listed exceptions).



---

**Note** You can roll back a service request even if the underlying workflow has been deleted.

---

- 
- Step 1** Choose **Organizations > Service Requests**.
- Step 2** On the **Service Requests** page, click **Service Requests**.
- Step 3** Choose the service request that corresponds to the workflow that needs to be rolled back.
- Step 4** (Optional) View the assets affected by the workflow. To view the workflow assets:
- Click **View Details**.
  - Click **Objects Created and Modified**.  
The list shows all the objects created or modified by the workflow.
  - Click **Close**.
- Step 5** Click **Rollback Request**.
- Step 6** Initially, all tasks that can be rolled back are checked. Uncheck any tasks that you do not want to roll back.
- Tasks for which rollback has been disabled (see [Configuring a Task in a Workflow, on page 33](#)) are unchecked. Checking these tasks to request their rollback results in an error when you submit the workflow rollback. If this happens, uncheck any tasks with disabled rollbacks and submit the workflow rollback again.
- You can roll back subtasks at any level in the workflow's hierarchy of compound tasks. The **Task Name** column in the **Rollback Service Request** dialog displays the ancestry of each listed task.
- Step 7** Click **Submit**.
- After the roll back is complete, and if an email policy is associated with the workflow, then an email notification on the roll back is sent to the configured recipients.

---

#### What to do next

Rollback workflows start with **Rollback** in the **Catalog/Workflow Name** column of the **Service Requests** page. Right-click a rollback service request and choose **View Details** to examine its status.

## Resubmitting a Service Request

You can resubmit all or part of a failed or cancelled service request.

- 
- Step 1** Choose **Organizations > Service Requests**.
- Step 2** On the **Service Requests** page, click **Service Requests**.
- Step 3** Click a failed or cancelled service request.
- Step 4** (Optional) Change user input values for the workflow before resubmitting the service request. You must have administrator privileges to change user inputs before resubmitting a workflow.

**Note** Only user workflow inputs can be changed. Admin inputs and task inputs cannot be changed before resubmitting.

To change the input values:

- a) Click **View Details**.
- b) Click **Input/Output**.
- c) Enter new values for any of the inputs.
- d) Click **Close**.

**Step 5** Click **Resubmit Request**. The **Resubmit Service Request** dialog appears.

**Step 6** Select a subtask from the **Resubmit Step** drop-down menu. The workflow will be resubmitted beginning with this task.

You can select any task up to and including the subtask that caused the workflow to fail.

**Note** You can select nested compound tasks at any depth for the resubmitted task.

**Important** If a task failed inside any loop task or outside of a completed loop task, in order to avoid mismatch in the iteration value and incorrect workflow execution, it is recommended to resubmit the service request from the failed task or rollback the service request and execute the workflow again.

**Step 7** (Optional) If all approvals succeeded before the workflow failed, bypass the approval tasks by checking **Skip approvals if already approved**.

**Step 8** (Optional) To reset a loop, check **Reset Loop** and enter an Index number.

**Note** This option is available only when Service Request is executed with the loop task.

**Step 9** Click **Submit**.

**Note** After you submit a failed service request, it is executed after a gap of about 20 seconds.

## Approving and Denying Service Requests

A workflow can include an approval task that requires a user, usually an administrator or someone with budget authority, to approve the workflow before it completes. To deal with workflows requiring your approval, do the following:

### Before you begin

A service request has been submitted with an Approval task that requires approval of a user ID that you own.

**Step 1** Choose **Organizations > My Approvals**.

**Step 2** On the **My Approvals** screen, choose an approval with status Pending in the **My Approval Status** column.

**Step 3** (Optional). To view the service request details, click **View Details**.

**Step 4** Choose what to do with the request:

- To approve the service request and allow the workflow to resume execution, click **Approve**.
- To deny the request but leave it in your approvals list in a rejected state, click **Reject**.

**Note** If you reject an approval, the workflow continues with the On failure path from the approval task. The On failure path can be used, for example, to execute an email task to notify a user that the approval was rejected. Or, it can execute another approval task, giving a different user an opportunity to approve the task.

- To deny the request and remove it from your approvals list, click **Cancel Request**. The workflow stops in a failed state at the approval task.

If you selected **Approve** or **Reject**, the **Service Request** screen appears. Continue to the next step.

If you selected **Cancel Request**, the **Cancel Service Request** dialog appears. Skip the next step.

**Step 5** (Optional). Enter a comment in the **Comment** field of the **Service Request** screen.

**Step 6** Click the button corresponding to your chosen action (**Accept**, **Reject**, or **Cancel Request**) to complete the action.

---

### What to do next

To see the status of any workflow in your approval list, click **View Details**.

To see the status of a workflow that you cancelled or a workflow not in your approval list, choose **Organizations > Service Requests** from the main menu.

## Deleting Service Requests

You can delete archived service requests from Cisco UCS Director. The deleted service requests are removed permanently from Cisco UCS Director.

You can enter archived service requests to delete in one of two ways:

- By selecting the service requests on the **Archived Service Requests** screen and clicking **Delete Requests**.
- By clicking **Purge Requests** and entering the IDs of the archived service requests.

Both methods result in the permanent removal of the specified service requests. The only difference is the method of data entry.

You can delete only archived service requests. For information about archiving service requests, see the current release of the [Cisco UCS Director Administration Guide](#). Because active service requests cannot be archived, you cannot delete service requests that are in progress, or that contain child service requests that are in progress.

You also cannot delete a service request that has a rollback that is in progress or that has failed. For example, say that you submit a rollback for service request (SR) 100 that generates a rollback service request SR 101. You cannot delete SR 100 while SR 101 is in progress. Furthermore, you cannot delete SR 100 if SR 101 fails.

To delete service requests, do the following:

---

**Step 1** Choose **Organizations > Service Requests**.

**Step 2** On the **Service Requests** page, click **Archived Service Requests**.

**Step 3** You can either enter service request IDs using the keyboard or choose service requests from the **Archived Service Requests** report.

To enter service request IDs, skip to the next step. To choose service requests instead, do the following:

- a) Choose all the service requests that you want to delete.

**Note** Select multiple items as you would in any other application on your system. For example, in Windows, hold down the **Ctrl** key to choose more items or **Shift** to choose a range of items.

When you choose one or more service requests, the **Delete Request** icon appears.

- b) Click **Delete Request**.
- c) On the **Delete Request** screen, click **Delete**.

**Step 4** To enter service requests, do the following:

- a) Click **Purge Requests**.
- b) In **SR IDs** on the **Delete Request** screen, enter the IDs of the service requests that you want to delete . Use hyphens to indicate ranges of IDs and commas to separate ranges or individual IDs; for example: **101-111, 113, 116-118**.
- c) Click **Delete**.

## Locking a Service Request

You can lock a service request to prevent archiving of the service request.

**Step 1** Choose **Organizations > Service Requests**.

**Step 2** On the **Service Requests** page, click **Service Requests**.

**Step 3** Choose one or more service requests that you want to lock.

- a) To lock a service request, choose the service request and click **Lock/Unlock**.
- b) Click **Submit**.

**Note** To unlock the service request, click **Lock/Unlock** and click **Submit**.

**Step 4** To lock more than one service request, do the following:

- a) Click **Lock/Unlock**.  
The **Service Request(s) Lock/Unlock** screen comes up.
- b) Click the **Lock** radio button and click **Lock** to prevent archiving of the service requests.

**Note** To unlock more than one service request, choose the service requests, click the **Unlock** radio button and click **Unlock**.



## CHAPTER 7

# Managing Workflows

---

This chapter contains the following sections:

- [Workflow Editing, on page 81](#)
- [Exporting and Importing Cisco UCS Director Artifacts, on page 89](#)
- [Workflow Version History, on page 95](#)
- [Cloning a Workflow, on page 98](#)
- [Searching a Workflow, on page 99](#)
- [Saving a Picture of a Workflow, on page 100](#)

## Workflow Editing

You can edit many features of an existing workflow. You can:

- Rename a workflow
- Reorder inputs
- Delete inputs
- Rename inputs
- Change an input from optional to mandatory (but not the reverse)
- Edit tasks in the workflow
- Delete tasks from a workflow

You can also:

- Delete any or all versions of a single workflow
- Delete the default version or all versions of multiple workflows at one time

The following sections describe these procedures.



---

**Note** To modify a predefined workflow, we recommend that you create a new version of that workflow before making the changes. If you modify a default predefined workflow, you can affect what happens in Cisco UCS Director when you click an action button. Actions that you take in Cisco UCS Director, such as creating or modifying a VM, use the default predefined workflows.

For example, when you click the **Create** button on the **HyperConverged > Datastore** page, Cisco UCS Director runs the default Create HyperFlex Datastore workflow and creates a service request for the action.

---

## Renaming Workflows

You can rename any workflow in UCS Director. Exception: you cannot rename a compound task.



---

**Note** All versions of the workflow are renamed.

---

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Choose the workflow that you want to rename.
  - Step 4** Click **Edit**.
  - Step 5** In **Workflow Name** on the **Edit Workflow** screen, enter the new name for the workflow.
  - Step 6** Click **Next**.
  - Step 7** On the **Add User Inputs** screen, click **Next**.
  - Step 8** On the **Add User Outputs** screen, click **Submit**.
- 

## Reordering Inputs

After creating a workflow, you can change the order in which workflow user inputs are processed when the workflow is executed.

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Click **Edit**.
  - Step 4** On the **Edit Workflow Details** screen, click **Next**.
  - Step 5** On the **Edit User Inputs** screen, choose an input.
  - Step 6** Click the **Up Arrow** icon or **Down Arrow** icon to move the input up or down in the input order.  
Alternatively, you can drag and drop an input to a different place in the list.
  - Step 7** Click **Next**.

**Step 8** On the **Edit Workflow Outputs** screen, click **Submit**.

---

## Deleting Workflow Inputs

You can delete one or more inputs while editing an existing workflow.



**Note** You can also delete inputs in a cloned workflow.

Alternatively, while editing a task, you can navigate to the **Edit Workflow Inputs** screen by clicking **Manage Workflow User Inputs** on the **User Input Mapping** page.

---

To delete inputs from a workflow, do the following:

---

- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Choose the workflow you want to edit.
  - Step 4** Click **Edit**.
  - Step 5** Click **Next**.
  - Step 6** In the **Workflow User Inputs** list on the **Edit User Inputs** screen, choose the inputs that you want to delete.  
To select multiple inputs from the list, hold down the <Shift> key (for a range of items) or <Ctrl> key (to select individual items).
  - Step 7** Click **Delete (X)**.
  - Step 8** On the **Delete Entry** screen, click **Submit**.
- 

### What to do next

Click **Next** to move to the **Edit User Outputs** screen, then click **Submit**.

## Renaming Workflow Inputs

You can rename inputs while editing an existing workflow.

Alternatively, while editing a task, you can navigate to the **Edit Workflow Inputs** screen by clicking **Manage Workflow User Inputs** on the **User Input Mapping** page.

To rename inputs from an opened workflow, do the following:

---

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Choose the workflow you want to edit.
- Step 4** Click **Edit**.

- Step 5** Click **Next**.
- Step 6** In the **Workflow User Inputs** list on the **Edit User Inputs** screen, choose the input that you want to edit.
- Step 7** Click **Edit**.
- Step 8** On the **Edit Entry** screen, replace the **Input Label**.
- Step 9** Click **Submit**.
- Step 10** To edit another input, repeat the previous four steps.

---

#### What to do next

Click **Next** to move to the **Edit User Outputs** page, then click **Submit**.

## Changing Input Optional or Required Status

You can change the optionality (optional or required status) of workflow inputs.




---

**Note** You cannot map an optional workflow input to a mandatory task input. Similarly, you cannot change an input from mandatory to optional if it is mapped to a mandatory task input.

---

#### Before you begin

Select the workflow you want to edit and choose the **Edit** action.

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Choose the workflow you want to edit.
  - Step 4** Click **Edit**.
  - Step 5** Click **Next**.
  - Step 6** In the **Workflow User Inputs** list on the **Edit User Inputs** screen, choose the input that you want to edit.
  - Step 7** Click **Edit**.
  - Step 8** On the **Edit Entry** screen, check or uncheck **Optional** to make the input optional or required, respectively, at the time of execution.
- Note** You cannot make a workflow user input optional if it is mapped to a mandatory task input.
- Step 9** Click **Submit**.

---

#### What to do next

Click **Next** to move to the **Edit User Outputs** screen, then click **Submit**.

## Editing a Task in an Existing Workflow

You can edit a task in an existing workflow. The change is local to the workflow and version containing the edited task. For example, if you edit task A in version 0 of workflow X, the task remains unchanged in workflow Y and in version 1 of workflow X.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** In the Workflow report, click the workflow you want to edit.
- Step 4** Click **Workflow Designer**.
- Step 5** In the workflow display, double-click the task that you want to edit. The **Edit Task** screen appears.
- Step 6** On the **Task Information** screen, the following fields are available:

Name	Description
<b>Task Name</b>	You cannot change the name of the task.
<b>Task Category</b>	You cannot change the task category.
<b>Task Type</b>	You cannot change the task type.
<b>Comment</b>	You can change or add to the comment.
<b>Retry Execution</b>	You can check or uncheck <b>Retry Execution</b> . If <b>Retry Execution</b> is checked, <b>Retry Count</b> and <b>Retry Frequency</b> controls appear.
<b>Disable Rollback</b>	You can check or uncheck <b>Disable Rollback</b> . If <b>Disable Rollback</b> is checked, the task is unchecked in the <b>Rollback Service Request</b> screen and cannot be rolled back; see <a href="#">Rolling Back a Service Request, on page 76</a> .  If the task does not support rollback, checking <b>Disable Rollback</b> has no effect.
<b>Retry Count</b>	The number of times (up to 5) to retry the task on failure.
<b>Retry Frequency</b>	A comma-separated list of integers specifying how many seconds to wait between retries. For example, if you selected a retry count of three, enter <b>1, 10, 60</b> to retry after one, 10, and 60 seconds, respectively.

- Step 7** Click **Next**.
- Step 8** On the **User Input Mappings** screen, you can edit the following fields:

Name	Description
<b>Manage Workflow User Inputs</b>	Click <b>Manage Workflow User Inputs</b> to add to or edit the workflow user inputs.

Name	Description
<b>Task Input</b>	Each task input is displayed as a heading on the remainder of this page. The input is labeled ( <b>Mandatory</b> ) if it is required.
<b>Map to User Input</b>	Check <b>Map to User Input</b> to populate the task input from a workflow user input at runtime.
<b>User Input</b>	This drop-down list appears if you check <b>Map to User Input</b> . Select the workflow user input to map to the task input.

**Step 9** Click **Next**.

**Step 10** On the **Task Input** screen, you can edit the following fields:

Name	Description
(input label)	Each task input that was not mapped to a user input on the previous page is displayed with an appropriate input control (text field, drop-down list, and so on). Use the input controls to specify the value. Optional inputs can be left blank.

**Step 11** On the **User Output Mapping** screen, you can edit the following fields:

Name	Description
<b>Task Output</b>	Each task output is displayed as a heading on this page. The output is labeled ( <b>Mandatory</b> ) if it is required. The output heading labels are assigned automatically and cannot be modified.
<b>Map to User Output</b>	Check <b>Map to User Output</b> to populate a workflow user output from the task output at runtime.
<b>User Output</b>	This drop-down list appears if you check <b>Map to User Output</b> . Select the workflow user output to map to the task output.

**Step 12** Click **Submit**.

## Opening Workflow Designer from a Header Search

You can use the search field in the Cisco UCS Director header to find and select a workflow from anywhere in Cisco UCS Director. You can then open the workflow in the **Workflow Designer**.

To find and edit a workflow from the Cisco UCS Director header:

### Before you begin

You have previously created a workflow that you want to edit immediately from any page in Cisco UCS Director.

- 
- Step 1** In the header bar of Cisco UCS Director, click the **Search** icon. The **Search Category** drop-down list and **Search** field are displayed.
  - Step 2** Select **Workflow** from the **Search Category** drop-down list.
  - Step 3** Begin entering the name of the workflow you want to open. Once there are three or more letters in the **Search** field, a drop-down list of workflow name matches appears.
  - Step 4** Finish entering the workflow name or select the workflow name from the drop-down list.

---

The workflow opens in the **Workflow Designer**.

### What to do next

Modify the workflow just as you would if you had selected the workflow from the **Orchestration > Workflow** page. When you are done, Cisco UCS Director returns to the page you were on when you opened the **Workflow Designer**.

## Deleting a Workflow

You can delete any or all versions of a workflow.

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Choose the workflow you want to delete.
  - Step 4** Click **Delete**.
  - Step 5** On the **Delete Workflow(s)** screen, choose the versions that you want to delete. To choose all versions for deletion, check the box at the top of the list.
  - Step 6** Click the **Delete** button.

## Deleting Multiple Workflows

You can delete more than one workflow at a time.

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Choose all the workflow you want to delete.
- Note** Select multiple items as you would in any other application on your system. For example, in Windows, hold down the **Ctrl** key to choose more items or **Shift** to choose a range of items.

- Step 4** Click **Delete All**.
- Step 5** On the **Delete Workflow** screen, check **Delete All Versions** if you want to delete all versions of the selected workflows. Otherwise only the default version is deleted.
- Step 6** Click **Delete**.

## Running a Quick Export on Workflows

Use the **Quick Export** option to select multiple workflows and export them to your machine.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Choose the workflows that you want to export.
- Step 4** Click **Quick Export**.

The subsequent screen lists the following information for the selected workflows:

- Script modules
- Workflow
- Custom tasks
- Activities
- Open APIs

- Step 5** In the **Export Workflow** screen, complete the following information:

Name	Description
<b>Exported By</b>	Provide your name.
<b>Comments</b>	Enter any comments on the workflows that you have selected.
<b>Encrypt the exported file</b>	Check the <b>Encrypt the exported file</b> check box to encrypt the file to be exported. By default, the check box is checked.
<b>Key</b>	Enter the key for encrypting the file.  This field is displayed only when the <b>Encrypt the exported file</b> check box is checked.  Preserve the key as it is needed while importing the workflow for decryption.
<b>Confirm Key</b>	Enter the key again for confirmation.  This field is displayed only when the <b>Encrypt the exported file</b> check box is checked.
<b>Exported File Name</b>	Provide a file name.

**Step 6** Click **Export Workflow**.

---

## Exporting and Importing Cisco UCS Director Artifacts

You can export and import workflows, custom tasks, script modules, and activities in Cisco UCS Director. This is useful if, for example, you want to:

- Move or copy workflows or other entities to different Cisco UCS Director instances.
- Back up or store entities.
- Use templates to standardize workflows.

### Exporting and Importing Artifact Files

Cisco UCS Director artifacts are exported from and imported to Cisco UCS Director in a single package. The you exported or imported file has a `.wfdx` extension and is an XML file containing a serialized representation of the objects. The file contains at least one of the following:

- One or more workflows
- One or more custom tasks
- One or more script modules
- One or more activities
- One or more api's

For example, you might import a file that contains only a single custom task; or a file that contains several workflows, a script module, and a few activities; and so on.

Custom inputs are exported when contained in a custom task. The custom inputs become available in Cisco UCS Director when the custom task is imported.

### Exporting and Importing Workflows as Templates

You can export and import workflows as templates.

When you export a template, all of the tasks within the workflow are retained. When you create a new workflow using a template, task inputs that have been mapped to workflow inputs cannot be modified. However, you can modify inputs which have not been mapped.

## Exporting Workflows, Custom Tasks, Script Modules, and Activities

To export artifacts from Cisco UCS Director, do the following:



---

**Note** Global variables associated with a workflow will be automatically exported while exporting a workflow.

---

**Step 1** Click **Export**.

**Step 2** On the **Select Workflows** screen, choose the workflows that you want to export.

**Note** Custom workflows, tasks, and scripts created in Cisco UCS Director before version 6.6 can fail to import if they contain XML data.

**Step 3** Click **Next**.

**Step 4** On the **Select Custom Tasks** screen, choose the custom tasks that you want to export.

**Note** The exported custom task contains all custom inputs that are used by that custom task.

**Step 5** Click **Next**.

**Step 6** On the **Export: Select Script Modules** screen, choose the script modules that you want to export.

**Step 7** Click **Next**.

**Step 8** On the **Export: Select Activities** screen, choose the activities that you want to export.

**Step 9** Click **Next**.

**Step 10** On the **Export: Select Open APIs** screen, choose the APIs that you want to export.

**Step 11** On the **Export: Confirmation** screen, complete the following fields:

Name	Description
<b>Exported By</b>	Your name or a note on who is responsible for the export.
<b>Comments</b>	Comments about this export.
<b>Encrypt the exported file</b>	Check the <b>Encrypt the exported file</b> check box to encrypt the file to be exported. By default, the check box is checked.
<b>Key</b>	Enter the key for encrypting the file.  This field is displayed only when the <b>Encrypt the exported file</b> check box is checked.  Preserve the key as it is needed while importing the workflow for decryption.
<b>Confirm Key</b>	Enter the key again for confirmation.  This field is displayed only when the <b>Encrypt the exported file</b> check box is checked.
<b>Exported File Name</b>	The name of the file on your local system. Type only the base filename; the file type extension (.wfdx) is appended automatically.

**Step 12** Click **Export**.

You are prompted to save the file.

## Importing Workflows, Custom Tasks, Script Modules, and Activities

To import artifacts into Cisco UCS Director, do the following:



**Note** Global variables associated with a workflow will be imported while importing a workflow if the global variable is not available in the appliance.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Click **Import**.
- Step 4** On the **Import** screen, click **Select a File**.
- Step 5** On the **Choose File to Upload** screen, choose the file to be imported. Cisco UCS Director import and export files have a `.wfdx` file extension.
- Step 6** Click **Open**.  
When the file is uploaded, the **File Upload/Validation** screen displays `File ready for use` and **Key**.
- Step 7** Enter the key that was entered when exporting the file.
- Step 8** Click **Next**.  
The **Import Policies** screen displays a list of Cisco UCS Director objects contained in the uploaded file.
- Step 9** (Optional) On the **Import Policies** screen, specify how objects are handled if they duplicate names already in the workflow folder. On the **Import** screen, complete the following fields:

Name	Description
<b>Workflows</b>	Choose from the following options to specify how identically named workflows are handled: <ul style="list-style-type: none"> <li>• <b>Replace</b>—Replace the existing workflow with the imported workflow.</li> <li>• <b>Keep Both</b>—Import the workflow as a new version.</li> <li>• <b>Skip</b>—Do not import the workflow.</li> </ul>
<b>Custom Tasks</b>	Choose from the following options to specify how identically named custom tasks are handled: <ul style="list-style-type: none"> <li>• <b>Replace</b></li> <li>• <b>Keep Both</b></li> <li>• <b>Skip</b></li> </ul>
<b>Script Modules</b>	Choose from the following options to specify how identically named script modules are handled: <ul style="list-style-type: none"> <li>• <b>Replace</b></li> <li>• <b>Keep Both</b></li> <li>• <b>Skip</b></li> </ul>

Name	Description
<b>Activities</b>	Choose from the following options to specify how identically named activities are handled: <ul style="list-style-type: none"> <li>• <b>Replace</b></li> <li>• <b>Keep Both</b></li> <li>• <b>Skip</b></li> </ul>
<b>Import Workflows to Folder</b>	Check <b>Import Workflows to Folder</b> to import the workflows. If you do not check <b>Import Workflows to Folder</b> and if no existing version of a workflow exists, that workflow is not imported.
<b>Select Folder</b>	Choose a folder into which to import the workflows. If you chose <b>[New Folder..]</b> in the drop-down list, the <b>New Folder</b> field appears.
<b>New Folder</b>	Enter the name of the new folder to create as your import folder.

**Step 10** Click **Import**.

## Workflow Templates

A workflow template is like an outline of a workflow. It contains placeholders for all the tasks in the workflow, but is not executable.

A workflow template is useful when you want to import a workflow to a different Cisco UCS Director instance, especially if the workflow is large or contains custom tasks.

A workflow template contains the following elements:

- Task names. Only the names are exported; the tasks themselves must exist in the Cisco UCS Director environment in order to execute.
- Workflow structure (connections between tasks).
- Input names.

A workflow template does not contain these elements:

- Custom tasks. Since only task names are exported in a template, you must import custom tasks in the new Cisco UCS Director environment in order for them to execute.
- Admin input values. Since the elements referred to by input values are unlikely to exist in the new Cisco UCS Director environment, input values must be redefined.
- Custom workflow input values. As with admin input values, workflow input values must be redefined.

## Exporting Workflows as Templates

You can export a workflow as a template. To export a workflow as a template:

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Choose the workflow you want to export.
- Step 4** Click **Export As Template**.
- Step 5** On the **Export as Template** screen, complete the following fields:

Name	Description
Template Name	The name for the template.
Description	A description of the template.

- Step 6** Click **Submit**.  
You are prompted to save the template to your system.
- Note** The template is saved under a system-generated filename. The template name you assign is for display in Cisco UCS Director only.
- 

## Importing a Workflow Template

To import a workflow template:

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Click **Import**.
- Step 4** On the **Import** screen, click **Upload**.
- Step 5** On the **File Upload** screen, click **Click and select a file from your computer**.
- Step 6** Select the template file. Cisco UCS Director template files have a `.wft` file extension. When the file is uploaded, the **File Upload** dialog displays `File ready for use`.
- Step 7** Dismiss the **File Upload** dialog.
- Step 8** On the **Import Template** screen, click **Submit**.
- 

The template appears on the **Workflow Templates** page.

## Creating a Workflow from a Template

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflow Templates**.
- Step 3** Choose a template.
- Step 4** Click **Create Workflow**.
- Step 5** On the **Overview** screen, complete the following fields:

Name	Description
Name	The name of the workflow to be created.
Description	An optional description for the workflow.

**Step 6** Click **Next**.

**Step 7** Click through the remaining screens of the **Create Workflow from Template** wizard. You can change any editable task-specific inputs that appear.

**Step 8** Click **Submit**.

The workflow is created in the **Template Workflows** folder on the **Workflows** page.

### What to do next

Open the workflow in the **Workflow Designer** and configure it to work in its new environment.

## Predefined Templates

Cisco UCS Director installs with the following predefined templates. The templates are available, along with any user-imported templates, on the **Policies > Orchestration > Workflow Templates** screen.

Name	Purpose	Description
<b>Customer Onboarding</b>	Customer onboarding in a secure multitenancy setting	<p>The workflow does the following:</p> <ul style="list-style-type: none"> <li>• Creates a virtual data center (VDC)</li> <li>• Generates a VLAN from the selected VLAN policy</li> <li>• Creates a VLAN on the selected Cisco Nexus 5000 Series switches</li> <li>• Creates a port profile on selected Nexus 1000v Series switches</li> <li>• Creates a VLAN interface</li> <li>• Creates flexible volume</li> <li>• Creates and establishes a vFiler on the controller</li> <li>• Mounts the storage as a data store on a selected host node</li> </ul>
<b>Deploy ESXi Host</b>	Deployment of a new blade as an ESXi Host on a VMware (vCenter) account	<p>The workflow does the following:</p> <ul style="list-style-type: none"> <li>• Creates a service profile</li> <li>• Associates the service profile to a selected server</li> <li>• Configures SAN zones on a Cisco Nexus 5000 Series switch</li> <li>• Establishes a PXE boot</li> <li>• Monitors PXE booting and registers the host with a selected VMware (vCenter) account</li> </ul>

Name	Purpose	Description
<b>Deploy ESXi Host with ONTAP</b>	Deployment of a new blade as an ESXi Host on a VMware (vCenter) account.	Workflow does the following: <ul style="list-style-type: none"> <li>• Creates a service profile</li> <li>• Associates the service profile to a selected server</li> <li>• Configures SAN zones on a Cisco Nexus 5000 Series switch</li> <li>• Establishes a PXE boot</li> <li>• Monitors PXE booting and registers the host with a selected VMware (vCenter) account</li> </ul>
<b>Deploy ESXi Host of Local Storage (without a Cisco Nexus 1000v Series switch)</b>	Deployment of a new blade as an ESXi local storage host (without using a Cisco Nexus 1000v Series switch) on a VMware (vCenter) account	The workflow does the following: <ul style="list-style-type: none"> <li>• Creates a service profile</li> <li>• Associates the service profile to a selected server</li> <li>• Configures SAN Zones on a Cisco Nexus 5000 Series switch</li> <li>• Establishes a PXE boot</li> <li>• Monitors PXE booting and registers the host with a selected VMware (vCenter) account</li> </ul>
<b>Attach VLAN to Storage Controller</b>	Attaching a NetApp storage controller's Ethernet ports to a new VLAN	The workflow does the following: <ul style="list-style-type: none"> <li>• Generates a VLAN from the selected VLAN policy</li> <li>• Creates a VLAN on the selected devices</li> <li>• Creates a VLAN interface</li> <li>• Creates a vFiler on the controller</li> </ul>

## Workflow Version History

Cisco UCS Director provides a set of features for managing versions of a workflow. See [Workflow Versioning, on page 10](#).

The following sections discuss how to manage workflow versions.

## Creating a New Version of a Workflow

You can create a new version of a workflow. You can then modify the new version without changing the old version.

### Before you begin

You have an existing workflow that you want to modify.

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Choose the workflow to duplicate.

**Step 4** Click **Create New Version**.

**Step 5** On the **Create New Version** screen, complete the following fields:

Name	Description
<b>Version Label</b>	The label for the version. The default label for a new workflow is zero, so consider labeling new versions numerically starting from one.
<b>Description</b>	A description for the version.

**Step 6** Click **Submit**.

The new version opens in the **Workflow Designer**.

**Step 7** In **Workflow Designer**, complete your changes for the workflow.

**Step 8** Click **Close**.

If the **Manage Versions** settings specify **Set latest version as default**, the new version becomes the default version. The **Workflows** report displays the new version of the workflow under the **Version** column.

If the **Manage Version** settings specify **Set selected version as default**, the new version is created but the default version does not change. See [Choosing the Default Version of a Workflow, on page 96](#).

## Choosing the Default Version of a Workflow

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Workflows**.

**Step 3** From the **More Actions** drop-down list, choose **Manage Versions**.

**Step 4** On the **Manage Versions** screen, complete the following fields:

Name	Description
<b>Show latest version</b>	<p>Check <b>Show latest version</b> to set the default version of a workflow to be the last version that was created. The default version of a workflow is displayed in the <b>Workflows</b> page. If you create a new version of a workflow, the new version becomes the default.</p> <p>For example, if the latest version of a workflow is X.0, the <b>Version Label</b> column in the Workflows page displays X.0. If you then create a version X.1, the <b>Version Label</b> column displays X.1.</p>
<b>Set default version</b>	<p>Check <b>Set default version</b> to set the default version of a workflow to be the version that you select. The default version of a workflow is displayed in the <b>Workflows</b> page. If you create a new version of a workflow, your selected version remains the default.</p> <p>For example, if the latest version of a workflow is X.0, the <b>Version Label</b> column in the Workflows page displays X.0. If you then create a version X.1, the <b>Version Label</b> column remains X.0.</p>
<b>Version</b>	If you chose <b>Set default version</b> , check one version from the list to be the default version.

**Step 5** Click **Submit**.

---

## Managing Versions of a Compound Task

When you update a compound task, the new version of the compound task replaces the old one in all workflows that use the compound task.

However, it is possible for some versions of a workflow to be compound tasks while other versions of the same workflow are not. Normally, the behavior of a workflow is defined by the workflow's default version, but this is sometimes not the case with compound tasks.

If the default version of a workflow is not a compound task, then when the workflow is called as a compound task the system uses the most recent version of the workflow that is defined as a compound task. This can be confusing. The following example illustrates this behavior.

### Before you begin

1. Create a compound task called *CT*. Note that in the Workflow tab, the Version Label for *CT* is shown as 0. We will refer to this version of *CT* as *CTv0*.
2. Include the compound task in a workflow called *WF*.
3. Validate the workflow *WF*. Validation should pass.

---

**Step 1** Save *CTv0* as a normal (non-compound-task) workflow:

- a) Choose **Orchestration**.
- b) On the **Orchestration** page, click **Workflows**.
- c) Choose the compound task workflow *CTv0*.
- d) Click **Edit Workflow**.
- e) Uncheck **Save as Compound Task**.
- f) Click through to the **Edit User Outputs** screen and click **Submit**.

**Step 2** In the **Workflow Designer**, validate *WF*.

- a) On the **Orchestration** page, click **Workflows**.
- b) Choose the workflow *WF*.
- c) Click **Workflow Designer**.
- d) On the **Workflow Designer** page, click **Validate Workflow**.

Validation fails with the message The task "Compound add user" does not exist anymore.

**Step 3** Create a new version of *CT* (*CTv1*).

- a) On the **Orchestration** page, click **Workflows**.
- b) Choose the workflow *CTv0*.
- c) Click **Create New Version**.
- d) On the **Create New Version** screen, enter the required fields:
  - Version Label - Something recognizable, for example **v1**.
  - Description - Any description.

- e) Click **Submit**.
- f) When the **Workflow Designer** comes up, click **Close**.
- g) Click **Edit**.
- h) Verify that for *CTv1*, **Save as Compound Task** is unchecked.
- i) Close the **Edit Workflow** page.

**Step 4** Validate *WF* again, as in Step 2.  
Validation fails with the message The task "Compound add user" does not exist anymore.

- Step 5** Make *CTv1* a compound task.
- a) On the **Orchestration** page, click **Workflows**.
  - b) Choose the workflow *CT*.
  - c) Click **Edit Workflow**.
  - d) Check **Save as Compound Task**.
  - e) Click through to the **Edit User Outputs** page and click **Submit**.

**Step 6** Validate *WF* again.  
Validation is successful.

- Step 7** Change the default version of *CT* back to *CTv0*.
- a) On the **Orchestration** page, click **Workflows**.
  - b) Choose the workflow *CT*.
  - c) Click **Manage Versions**.
  - d) On the **Manage Versions** screen, uncheck **Set Latest Version as Default**.
  - e) Check **Set Selected Version as Default**.
  - f) In the table, check *CTv0*.
  - g) Click **Submit**.

In the **Workflows** tab, **Version Label** is now *0*, indicating that *CTv0* is the default version of *CT*.

- Step 8** Validate *WF* again.  
Validation is successful. Note that *WF* is using *CTv1* as the compound task, even though *CTv0* is the default. Be aware that this can cause confusion if you have more than one version of a workflow that is used as a compound task.

## Cloning a Workflow

You can clone a workflow. The cloned workflow is identical to the original workflow. You can edit the new workflow immediately. You might do this, for example, to create a workflow that is similar to the original workflow but which has different inputs. The new workflow has a new, separate version history.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Select the workflow you want to clone.
- Step 4** Click **Clone**.  
The **Clone Workflow** screen comes up.
- Step 5** Edit the workflow details, inputs, and outputs. See [Creating a Workflow, on page 17](#). You must enter a new name for the workflow.
- Step 6** Click **Next** to proceed to the **Workflow User Inputs** screen.

**Step 7** Edit the workflow user inputs.

---

## Searching a Workflow

---

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Workflows**.

**Step 3** Click **Search Workflow**.

**Step 4** On the **Search Details** screen, the following fields are available:

Name	Description
<b>Search Workflow by drop-down list</b>	Choose the option that you want to search. The options are: <ul style="list-style-type: none"> <li>• Tasks</li> <li>• Activities</li> <li>• Workflow Inputs</li> <li>• Workflow Outputs</li> <li>• Global Variables</li> </ul>
<b>Tasks</b>	Click <b>Select</b> and choose a task. Click <b>Select</b> . This field is displayed based on the option chosen in the <b>Search Workflow by drop-down list</b> .
<b>Activities drop-down list</b>	Choose an activity. This field is displayed based on the option chosen in the <b>Search Workflow by drop-down list</b> .
<b>Workflow Inputs</b>	Click <b>Select</b> and choose a workflow input. Click <b>Select</b> . This field is displayed based on the option chosen in the <b>Search Workflow by drop-down list</b> .
<b>Workflow Outputs</b>	Click <b>Select</b> and choose a workflow output. Click <b>Select</b> . This field is displayed based on the option chosen in the <b>Search Workflow by drop-down list</b> .
<b>Global Variables</b>	Click <b>Select</b> and choose a global variable. Click <b>Select</b> . This field is displayed based on the option chosen in the <b>Search Workflow by drop-down list</b> .

**Step 5** Click **Export Report**. Based on the search criteria, .csv file will be downloaded in your local system.

---

# Saving a Picture of a Workflow

You can create a graphics image of a workflow. To capture an image, do the following:

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Workflows**.
  - Step 3** Choose a workflow.
  - Step 4** Click **Workflow Designer**.
  - Step 5** From the **Workflow Designer Action Menu** click the **Export as Image** icon.  
The entire workflow is converted to a .png file and displayed in a new browser window.
  - Step 6** You can save the image using your browser.
-



## CHAPTER 8

# Customizing Workflow Components

---

This chapter contains the following sections:

- [Creating a Compound Task, on page 101](#)
- [Creating Custom Approvals, on page 102](#)
- [Creating Custom Inputs, on page 103](#)
- [Generating a Task Using a OpenAPI Specification File, on page 104](#)
- [Orchestration Global Variables, on page 106](#)

## Creating a Compound Task

A compound task is a workflow that functions as a single task. A compound task, like any other task, is atomic: its component tasks are hidden.

You create a compound task by saving a workflow as a compound task when you create or edit the workflow. Do this, for example, if you find yourself building the same series of tasks into several different workflows.

You can define a simple workflow and save it as a compound task, then define another workflow that incorporates the compound task. You can use this pattern to define increasingly complex workflows.

To save an existing workflow as a compound task, do the following:



---

**Note** To create a new compound task from scratch, see [Creating a Workflow, on page 17](#).

---

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Select a workflow to save as a compound task.
- Step 4** Click **Edit**.
- Step 5** Check **Save as Compound Task**.
- Step 6** If you want all of the workflow's task outputs available as output of the compound task, click **Publish Task outputs as Compound Task outputs**.
- Step 7** Click **Next**.
- Step 8** On the **Add User Inputs** screen, click **Next**.

**Step 9** On the **Add User Outputs** screen, click **Submit**.

---

The new compound task is available in the **Compound Task** folder when you open the **Workflow Designer**.

## Example: Creating a Compound Task

This example demonstrates repeating a workflow task for elements in a list.

### Before you begin

Create the example workflow as described in [Example: Creating a Workflow, on page 39](#).

**Step 1** Navigate to **Policies > Orchestration**.

**Step 2** Click the **Workflows** tab.

**Step 3** Locate and select the PowerCycleVM workflow you created in [Example: Creating a Workflow, on page 39](#).

**Step 4** Click **Edit**.

**Step 5** In the **Edit Workflow Details** window, check the **Save as Compound Task** check box.

**Note** None of the tasks has an output, so ignore the **Publish Task Outputs as Compound Task outputs** check box. The workflow has nothing to do with system startup of Cisco UCS Director, so ignore also the **Always execute during System initialization** check box.

**Step 6** Click through to the **Edit Workflow Output** page.

**Step 7** Click **Submit**.

---

### What to do next

Include the compound task in workflows. For example, you can insert this task before the **Completed (Failed)** task of a workflow that modifies a remotely hosted VM. Then, the VM restarts if a modification fails.

## Creating Custom Approvals

You can create custom approval tasks that allow approvers to enter input values at the time of workflow execution.

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Custom Approval Tasks**.

**Step 3** Click **Add**.

**Step 4** On the **Add Inputs** screen, complete the following fields:

Name	Description
Approval Task Name	The name of the approval task as it appears in the Workflow Designer.

Name	Description
Approval Task Description	A description of the approval task (optional).

**Step 5** Click **Add Input Field**.

**Step 6** Under the **User Input** heading for the new input field, complete the following fields:

Name	Description
Input Label	The label for the input (supplied by the approver of the task).
Input Description	A description of the input.
Input Type	A drop-down list containing the data type of the input.
Optional Input	If <b>Optional Input</b> is checked, the administrator must provide a default value for the input. The approver is not required to provide input.

**Step 7** Repeat the previous two steps to add as many inputs as needed.

**Step 8** Click **Submit**.

### What to do next

You can now include the custom task in a workflow.

## Creating Custom Inputs

You can create custom input types to use as workflow inputs. Custom input types are based on an existing input type. They are defined by filter criteria or by a selection set that further narrows the possible values of the input.

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Custom Workflow Inputs**.

**Step 3** Click **Add**.

**Step 4** On the **Add Custom Workflow Input** page, complete the following fields:

Name	Description
Custom Input Type Name	The input name.
Input Type	Clicking this button brings up a list of existing input types. From the list, choose an input type on which to base your input type.

Name	Description
<b>Filter</b>	<p>Depending on your choice of input type, one or more of the following filter types is available:</p> <ul style="list-style-type: none"> <li>• <b>Input Filter</b>—A text field. Type a text filter string.</li> <li>• <b>Input List</b>—A list of values (LOV). Choose which existing values are valid instances for this input.</li> <li>• <b>Input LOV</b>—Define a list of allowable name-value pairs for the input.</li> </ul> <p><b>Note</b>      The <b>Label</b> field and <b>Value</b> field descriptions should match.</p> <ul style="list-style-type: none"> <li>• <b>Input Range</b>—A text field. Type a range of valid character values.</li> <li>• <b>Validated Input</b>—Choose a validator type from the table.</li> </ul>

**Step 5** Click (+) **Add**.

**Step 6** Click **Submit**.

---

The new input type is added to the **Custom Workflow Types** page. The new input type is available for selection when defining workflow and task inputs.

## Generating a Task Using a OpenAPI Specification File

The OpenAPI specification file from connectors such as NetApp, VMware, and vCenter defines all APIs of connectors in the JSON or YAML format. You can upload the OpenAPI specification file to Cisco UCS Director to make use of the connector specific API for creating a task. You can use the task created based on a OpenAPIs specification file, in Cisco UCS Director workflow to perform a specific operation on the connector account.




---

**Note** Cisco UCS Director supports uploading of OpenAPI specification file of vCenter Release 6.5 or later.

---

### Before you begin

Ensure that you have the OpenAPI specification file of a connector, such as NetApp, VMware, Intersight, and vCenter, in the JSON format.

---

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **OpenAPI Integration**.

A list of connectors for which the OpenAPI specification files are uploaded to Cisco UCS Director, is displayed. Expand a connector to view the API specific details of the connector.

- Step 3** Upload a OpenAPI specification file of a connector to make use of the connector specific API details of the connector:
- Click **Upload Spec**.
  - Enter a name under which you want to import the APIs in the OpenAPI specification file.
  - Either drag and drop or click **Select a file** to upload a OpenAPI specification file.

**Note** While using OpenAPI version 2.0, the OpenAPI specification file must be in JSON format. If you have the OpenAPI specification file in any other different format such as YAML, convert the file in to JSON format by choosing **File > Convert and save as JSON** in the OpenAPI editor. If you are using OpenAPI version 3.0, the OpenAPI specification file can be in JSON or YAML format.

- Click **Submit**.  
The connector specific API details are downloaded to Cisco UCS Director.

- Step 4** Create a task using the connector specific API:

- Expand a connector folder and choose an API.
- Click **Generate Task**.

The **Generate Task** screen appears.

- Check the **Use Existing Account** check box to use one of the connector account available in Cisco UCS Director. If the **Use Existing Account** check box is left unchecked, Cisco UCS Director will ask for connector login credentials during task execution.

**Note** For importing APIs of Intersight, you must add respective Intersight DNS to the Cisco UCS Director server. The **Existing Account** and **Credential Policy** checkboxes are not applicable for Intersight APIs.

- Check the **Use Credential Policy** check box to choose a credential policy from Cisco UCS Director for account access.
- Check the **Use Proxy** check box to use the system proxy configuration.
- Enter the API operation name and description. These fields may be auto-populated with details based on information available in the OpenAPI specification file.
- Choose **http** or **https** as the protocol.

**Note** The VMware APIs does not support HTTP protocol.

- Enter the HTTP request type.
- Displays the base path of the API.
- Choose the HTTP content type from the drop-down list.
- Check the **Use the HTTP Basic Authentication** check box to pass the HTTP Basic Authentication parameters during API call.
- Check the **Use selected Account/policy credentials** check box to choose authentication parameters from selected account or credential policy.

The API path is displayed.

- Expand **API Parameters**. The default parameters that are applicable for the API are displayed. Click **Add** and complete all the fields in the **Add Entry to API Parameters** screen to add an additional API parameter.

You must add **Accept** and **Content Type** parameters for all the accounts. In addition, for VMware account, you must add a session ID parameter.

- Check the **Output Full Response** check box to output the full response of API.

- o) Expand **Output**. Click **Add** and define the API output that can be mapped as input for other tasks.
- p) Check the **Define Rollback** checkbox to define the rollback specifications for the task.
- q) Click **Select** and choose the rollback workflow that needs to be associated with the task. You have to create a rollback workflow for the task in the Workflow designer.
- r) Expand **Rollback Workflow Inputs**. Click **Add** and define the parameters for the rollback workflow input.

The **Password** field must be configured as user input with the help of Macros. In the **Password** field, avoid hardcoding the password.

The parameters defined as rollback workflow input, will get displayed under the **API Parameters** field after creating the task by clicking **Submit**.

- s) Click **Submit**.

### What to do next

In the **Workflow Designer**, you can drag and drop the newly created task from the **Available Tasks > Custom Tasks > OpenAPI Custom tasks** folder to the workflow area and use the task in a workflow.

To reset an API to their original form, choose an API within the connector folder and click **Reset**. Before initiating the reset process, ensure that the workflows that use the API task are deleted from Workflow designer and tasks created based on the API are deleted.

To delete a connector folder, select the folder and click **Delete Folder**. Before initiating the folder deletion process, ensure that APIs within the connector folder are not associated to any task or workflow.

## Orchestration Global Variables

Global variables are variables that you can use within Cisco UCS Director Orchestrator to expose several types of variable system information in two places:

- In task input variables inside a workflow, where you can access such information as:
  - Workflow inputs and task outputs
  - Service request IDs
  - VM information such as ID, name, IP address, and power state
- In VM names, where you can access such information as:
  - User information such as group and user IDs
  - Configuration information such as catalog and system profile
  - Deployment information such as cloud name and location

You can also define your own global variables and make them available in workflows.



**Note** Global variables are overridden by local workflow inputs or outputs. See [Input and Global Variable Scoping, on page 108](#).

Export and Import of the Global variables are supported in workflows. If the global variable is available in the importing appliance the global variable will not be imported.

## Orchestration Variables

*Global variables* are variables that can be accessed by any workflow task during workflow execution. When you create a Cisco UCS Director workflow, you can use variables in any of the workflow's task, user, or admin input values.

An input field can contain any combination of text and variables. During execution of the workflow, Cisco UCS Director Orchestrator substitutes the variables' values into each task's inputs before executing the task. Furthermore, variables can be nested; nested variables are evaluated recursively from the innermost to the outermost nested variable as long as the expression still contains a valid variable name with an assigned value. See [Input and Output System Variables, on page 107](#).

Variables from the following four categories are available:

### **Input and Output System Variables**

Input and output variables are simply the inputs and outputs of a workflow at runtime. These inputs and outputs can be captured and used to build inputs to tasks in the workflow.

### **Service Request Global Variables**

Service request global variables provide information about the service request containing the current running workflow.

### **Virtual Machine Global Variables**

For workflows executed in the context of a Virtual Machine (VM), VM global variables contain information about the VM.

### **User-Defined Global Variables**

You can create user-defined global variables that are available to any task in any service request. Unlike system-supplied variables, user-defined global variables can be modified and deleted.

The following sections describe the various types of variables.

## Input and Output System Variables

Any workflow-level input or previous task output can be used as a variable in a subsequent task. For example, consider a workflow that has two inputs labeled *Enter Disk Size* and *Max Snapshots*. Suppose that the workflow has two tasks with IDs *task1* and *task2* (arranged so that *task1* executes first). Any input values to *task1* or *task2* that takes free-form input can use the following two variables:

- `${Enter Disk Size}`
- `${Max Snapshots}`




---

**Note** System variables consist of a dollar sign (\$) followed by the variable name in curly braces ({}). A workflow level input can be used as a variable by including the label associated with the user input in the variable definition.

---

Also the second task, *task2*, can use the output of *task1*. If *task1* has two output variables, *OUTPUT\_VOLUME\_NAME* and *OUTPUT\_VOLUME\_SIZE*, then *task2* can use the following notation to capture their values in its inputs:

- `${task1.OUTPUT_VOLUME_NAME}`
- `${task1.OUTPUT_VOLUME_SIZE}`




---

**Note** The variable name for a task output is the task name, followed by a period, followed by the task output variable name: `${taskName.outputName}`.

---

### Input and Global Variable Scoping

In the case where a task or workflow input variable has the same name as a global variable, the value of the input variable is substituted into the expression. The global variable is overridden by the local input variable.

For example, imagine there is a user-defined global variable BMA with the value **titanium**. Now suppose there is a workflow with a user input name BMA, and that during a service request a user enters **aluminum** for the value of the variable. If a task in the workflow includes a reference to the variable in an input expression: `${BMA}`, then the reference evaluates to **aluminum**. The global value of **titanium** is "hidden" for the duration of the workflow.




---

**Note** Scoping is calculated every time a variable substitution is made in a context with more than one identically named variables. This includes substitutions in nested variables at every level.

---

### Using Input Variables to Filter Admin Inputs

You can use input variables to populate an admin input, admin input list, or admin input filter. For example, suppose you have a workflow with the following two custom task inputs:

- Host Pool, of type `gen_text_input` (generic text), with an input list of values (LOV) containing several pool labels **Pool 1**, **Pool 2**, etc.
- Host, of type `gen_text_input`, with an input LOV containing several host/pool label combinations **Host 1 Pool 1**, **Host 2 Pool 1**, etc., and with an admin filter that refers to the Host Pool input as follows: `CONTAINS ${Host Pool}`.

When you run the workflow and select a pool from the Host Pool LOV, the Host input LOV shows only those entries containing the pool that you selected. For example, if you select **Pool 2** from the Host Pool LOV, the Host LOV is limited to **Host 1 Pool 2**, **Host 2 Pool 2**, and so on. If you select a different pool, the input screen refreshes again and only hosts from the new pool appear in the Host LOV.



**Note** In this example, when you select an entry from the Host Pool LOV, the workflow input screen refreshes, updating all controls on the page, including the Host LOV. This refresh behavior cannot be triggered by simple text input controls. Therefore, filtering on workflow inputs does not work for free-form, non-list inputs on the same screen.

For example, if Host Pool were of type `gen_text_input` (rather than a custom input type with an LOV), the Host drop-down would never update no matter what you typed in the Host Pool field.

### Nested Variables

You can nest variables in input values. Cisco UCS Director resolves nested variables to any arbitrary level. Put another way, the result of every variable substitution is evaluated to resolve variables until there are no more variables to evaluate.

For example, assume that a workflow has two tasks, **GetStore** and **GetData**. The **GetData** task has an admin input **GetData.DataStore** defined as `#{GetStore.#{GetStore.StoreRef}}`.

The **GetStore** task has the following two outputs:

- StoreRef, with value **LUN**
- LUN, with value **7**

When **GetStore** runs, then evaluation of **GetData.DataStore** proceeds as follows:

`#{GetStore.StoreRef}`

evaluates to:

**LUN**

This value is substituted into the value of **GetData.DataStore**:

`#{GetStore.#{GetStore.StoreRef}}`

which then evaluates to:

`#{GetStore.LUN}`

Which in turn is evaluated to:

**7**

The following rules apply to nested variables:

- Nested input values can contain any combination of nested variables and text. For example, this is a valid input expression: `VM name is #{VM_#{VMIndex}}`. If `#{VMIndex} = 23` and `#{VM_23} = DNA`, then `#{VM_#{VMIndex}}` evaluates to `VM name is DNA`.
- Nested input values are evaluated any number of times until the resulting expression no longer contains any variable references. The innermost variable expression is evaluated first, then the resulting expression is evaluated, and so on. For example, this is a valid input expression:

`#{#{#{Earth}}}`.

If the following variables are set:

`#{Earth} = Sol`

```
#{Sol} = MilkyWay
```

```
#{MilkyWay} = Universe
```

then evaluation of the expression cascades to its final value as follows:

```
#{#{#{Earth}}}
```

```
#{#{Sol}}
```

```
#{MilkyWay}
```

```
Universe
```

- A user-defined global variable can contain references only to other global variables. See [User-Defined Global Variables](#), on page 110.

## Service Request Global Variables

In addition to workflow inputs and task outputs, the following variables representing service requests are available:

- `#{SR_ID}`—The ID of the current service request
- `#{PARENT_SR_ID}`—The ID of the service request that is the parent of the current service request. (Available only if the current service request has a parent.)
- `#{ROOT_SR_ID}`—The ID of the top most parent service request.

For the full list of VM system variables, see [List of VM Macros and VM Annotations](#), on page 113.

## Virtual Machine System Variables

For workflows that are executed in the context of a VM, more VM system variables are available. VM system variables cannot be used in a non-VM context workflow.

For the full list of VM system variables, see [List of VM Macros and VM Annotations](#), on page 113.

## User-Defined Global Variables

You can create, clone, modify, and delete global variables. Global variables are available to any task in any workflow in Cisco UCS Director.

User-defined global variables can be used only in generic text input types. They can be called in nested input variables. They can contain references to other global variables, but cannot contain references to input variables.




---

**Note** It is possible to delete a user-defined global variable that is used by a task in an existing workflow. When such a workflow is run, any task with a missing global variable runs without resolving the variable and might cause unanticipated results. It is up to administrators to ensure that service requests do not contain deleted global variables.

---

## Creating Global Variables

To create a global variable, do the following:

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Global Variables**.
- Step 3** Click (+) **Add**.
- Step 4** On the **Add Global Variable** screen, complete the following fields:
- In the **Name** field, enter a unique name for the global variable.
  - In the **Description** field, enter a description of the global variable (optional).
  - In the **Value** field, enter a value for the global variable. This is the text that is inserted when the global variable is used in a task.
- User-defined global variables have the following restrictions:
- A global variable **Name** may not contain any of the following 19 characters: "%&'\*\*+, ./:;<=>?^|}{
  - A global variable **Description** or **Value** may contain any character.
  - A global variable **Name**, **Description**, or **Value** may contain spaces, but may not start or end with a space. Space characters are stripped from the beginning and end of a name, value, or description.
  - Global variables are usable only in generic text type inputs.
- Step 5** Click **Submit**.
- 

## Cloning Global Variables

To clone a global variable, do the following:

### Before you begin

You have created a global variable. See [Creating Global Variables, on page 110](#).

---

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Global Variables**.
- Step 3** Choose the global variable you want to clone. You can choose a system-defined or a user-defined global variable.
- Step 4** Click **Clone**.
- Step 5** On the **Clone Global Variable** screen, complete the following fields:
- In the **Name** field, enter a unique name for the global variable. You must change the global variable's name.
  - In the **Description** field, enter a description of the global variable (optional).
  - In the **Value** field, enter a value for the global variable. This is the text that is inserted when the global variable is used in a task.
- User-defined global variables have the following restrictions:
- A global variable **Name** may not contain any of the following 19 characters: "%&'\*\*+, ./:;<=>?^|}{
  - A global variable **Description** or **Value** may contain any character.
  - A global variable **Name**, **Description**, or **Value** may contain spaces, but may not start or end with a space. Space characters are stripped from the beginning and end of a name, value, or description.
  - Global variables are usable only in generic text type inputs.

**Step 6** Click **Submit**.

---

## Editing Global Variables

To modify a global variable, do the following:

### Before you begin

You have created a global variable. See [Creating Global Variables, on page 110](#).

---

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Global Variables**.
- Step 3** From the `User Defined` folder, choose the global variable you want to edit.
- Step 4** Click **Edit**.
- Step 5** On the **Edit Global Variable** screen, complete the following fields:
- You cannot change the **Name** field of an existing global variable.
  - In the **Description** field, change or add a description of the global variable (optional).
  - In the **Value** field, change the value for the global variable. This is the text that is inserted when the global variable is used in a task.

User-defined global variables have the following restrictions:

- A global variable **Name** may not contain any of the following 19 characters: "%&'\*+,./:;<=>?^|}{
- A global variable **Description** or **Value** may contain any character.
- A global variable **Name**, **Description**, or **Value** may contain spaces, but may not start or end with a space. Space characters are stripped from the beginning and end of a name, value, or description.
- Global variables are usable only in generic text type inputs.

**Step 6** Click **Submit**.

---

## Deleting Global Variables

To delete a global variable, do the following:

---

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Global Variables**.
- Step 3** From the `User Defined` folder, choose the global variable you want to delete.
- Step 4** Click **Delete**.
- Step 5** On the **Delete Global Variable** screen, click **Delete**.
-

## VM Annotations

VM annotations represent information about a VM. You add these variables to a VMware system policy if you choose to define VM annotations in that policy. The output from these variables displays in the Annotations field of the VM in VMware vCenter.

For the full list of VM annotations, see [List of VM Macros and VM Annotations, on page 113](#).

## List of VM Macros and VM Annotations

The syntax that you use for VM macros can be different from the syntax used for VM annotations. In addition, more variables are available for VM macros than for VM annotations. The following table shows the correct syntax for VM macros and for VM annotations. If a cell contains N/A, the variable is not available in that context.

For information about the variables that you can use in the **VM Name Template** and **VM Host Name Template** fields of the system policy, see the [Cisco UCS Director Administration Guide](#).



**Note** This table does not include a complete list of Cloupiascript macros. For information on using Cloupiascript macros, see the [Cisco UCS Director Cloupiascript Cookbook](#).

Variable	VM Macros for Orchestration Workflows	VM Annotations for System Policies
VM name	<code>\${VM_NAME}</code>	<code>\${VMNAME}</code>
VM IP address	<code>\${VM_IPADDRESS}</code>	N/A
VM state (can be either <i>on</i> or <i>off</i> )	<code>\${VM_STATE}</code>	N/A
VM state details (can be either <i>power-on</i> or <i>power-off</i> )	<code>\${VM_STATE_DETAILS}</code>	N/A
ESX server or host node that hosts the VM	<code>\${VM_PARENT}</code>	N/A
Cloud used for VM provisioning	<code>\${VM_CLOUD}</code>	<code>\${CLOUD_NAME}</code>
Type of cloud	N/A	<code>\${CLOUD_TYPE}</code>
VM hostname	<code>\${VM_HOSTNAME}</code>	N/A
Short VM hostname	<code>\${VM_HOSTNAME_SHORT}</code>	N/A
VM hostname and domain	<code>\${VM_HOSTNAME_DOMAIN}</code>	N/A
Name of the group to which the VM belongs	<code>\${VM_GROUP_NAME}</code>	<code>\${GROUP_NAME}</code>
Full name of the group	N/A	<code>\${FULL_GROUP_NAME}</code>
ID of the group	<code>\${VM_GROUP_ID}</code>	N/A

Variable	VM Macros for Orchestration Workflows	VM Annotations for System Policies
Name of the parent group, if one exists	N/A	\${GROUP_PARENT}
ID of the catalog used to provision the VM	\${VM_CATALOG_ID}	N/A
Name of the catalog used to provision the VM	N/A	\${CATALOG_NAME}
VM ID	\${VM_ID}	N/A
Service request ID for the VM	\${M_SR_ID}	\${SR_ID}
Comments from the user who requested the VM	\${VM_COMMENTS}	\${COMMENTS}
Virtual data center name	\${VM_VDC_NAME}	N/A
Virtual data center ID	\${VM_VDC_ID}	N/A
Type of VM	\${VM_TYPE}	N/A
Scheduled termination of the VM	\${VM_SCHED_TERM}	N/A
Location specified in the account	N/A	\${LOCATION}
Cost center for the VM	N/A	\${COST_CENTER}
Current time in milliseconds converted to a unique ID for the VM	N/A	\${UNIQUE_ID}
User who requested the VM	N/A	\${USER}
Full name of the user who requested the VM	N/A	\${FULL_USER_NAME}
Appcode from the catalog	N/A	\${APPCODE}
Name of the system policy associated with the application category	N/A	\${PROFILE_NAME}
Name of the user who initiated the request	N/A	\${INITIATING_USER}
Simple name of the user who initiated the request	N/A	\${INITIATING_USER_SIMPLE_NAME}
Email address of the user who submitted the request	N/A	\${SUBMITTER_EMAIL}

<b>Variable</b>	<b>VM Macros for Orchestration Workflows</b>	<b>VM Annotations for System Policies</b>
The ID of the user who submitted the request	N/A	<code>#{SUBMITTER_USERID}</code>
First name of the user who submitted the request	N/A	<code>#{SUBMITTER_FIRSTNAME}</code>
Last name of the user who submitted the request	N/A	<code>#{SUBMITTER_LASTNAME}</code>
The role of the user who submitted the request	N/A	<code>#{SUBMITTER_ROLE}</code>
Name of the group to which the user who submitted the request belongs	N/A	<code>#{SUBMITTER_GROUPNAME}</code>
The ID of the group to which the user who submitted the request belongs	N/A	<code>#{SUBMITTER_GROUPID}</code>





## CHAPTER 9

# Using Activities

This chapter contains the following sections:

- [Activities, on page 117](#)
- [Creating an Activity, on page 117](#)
- [Associating an Activity with a Workflow, on page 118](#)
- [Adding an Activity to a Workflow, on page 120](#)
- [Importing and Exporting Activities, on page 121](#)
- [Deleting an Activity, on page 121](#)

## Activities

Activities provide a layer of abstraction to workflow input variables. The abstraction helps you to efficiently execute any administration scenario (such as tenant onboarding or application deployment) by separating the specification of a workflow from its actual implementation. Using activities, you can do the following:

- Define the input properties for a workflow and associate those properties with one or more workflows. Then, depending upon a matching condition, the activity triggers the correct workflow.
- Use an activity as a workflow task, making the task usable in multiple similar but differently implemented use case scenarios.
- An activity can be either triggered programmatically or by providing inputs dynamically.

## Creating an Activity

You create an activity by defining inputs and outputs. To create an activity:

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Activities**.
- Step 3** Click **Add**.
- Step 4** On the **Add Activity** screen, complete the following fields:

Name	Description
Name	The activity name.

Name	Description
Label	A display label for the activity.
Description field	A description of the activity.

**Step 5** Click **Next**.

**Step 6** On the **Add User Inputs** screen, click (+) **Add** to add workflow input properties.

a) On the **Add Entry to Input Fields** screen, complete the following fields:

Name	Description
Input Name	The name of the activity workflow input.
Input Label	The label for the activity workflow input.
Optional	Check <b>Optional</b> to make the input optional during workflow execution.
Input Type	Click <b>Select</b> . Choose the input type.

b) Click **Submit**.

**Step 7** Click **Next**.

**Step 8** On the **Add User Outputs** screen, click the (+) **Add** to add workflow output properties.

a) On the **Add Entry to Output Fields** screen, complete the following fields:

Name	Description
Output Name	The name of the activity workflow output.
Output Label	A display label for the activity workflow output.
Output Type	Click <b>Select</b> . Choose the output type.

b) Click **Submit**.

**Step 9** Click **Submit**.

## Associating an Activity with a Workflow

You can associate an activity with one or more workflows to run the activity as part of the workflow. The associated activity is displayed as a custom workflow task in the `System Activity Tasks` folder of the Workflow Designer.

### Before you begin

You have an activity and at least one workflow that is to include the activity. For more information about how to create an activity, see [Creating an Activity, on page 117](#).

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Workflows**.
- Step 3** Click **Add Workflow**.
- Step 4** On the **Add Workflow** screen, complete the following fields:

Name	Description
<b>Workflow Name</b>	<p>The workflow name.</p> <p><b>Note</b> Workflow names cannot contain the following characters: \, ", %, &amp;, ', *, +, ,, .., /, :, ;, &lt;, =, &gt;, ?, ^, /.</p>
<b>Description</b>	A description of the workflow.
<b>Workflow Context</b>	<p>Choose the context in which the workflow can be used. Orchestrator supports the following options:</p> <ul style="list-style-type: none"> <li>• <b>Any</b>—Enables you to use the workflow in any context.</li> <li>• <b>Selected VM</b>—Enables you to execute the workflow only when you choose a VM.</li> </ul>
<b>Save as Compound Task</b>	Check <b>Save as Compound Task</b> to define the workflow as a compound task.
<b>Place in New Folder</b>	Check <b>Place in New Folder</b> to assign the workflow to a new folder. The <b>Folder Name</b> field replaces the <b>Select Folder</b> drop-down list. Type a name for the new folder.
<b>Select Folder</b>	Choose the folder into which you want to place the workflow.
<b>Notify status of execution to initiating user</b>	<p>Check <b>Notify status of execution to initiating user</b> to notify the user through email of the execution status of the workflow. If checked, enter the additional email addresses in the <b>Additional User(s) to send Email Notification</b> field. The execution status for the workflow can be one of the following:</p> <ul style="list-style-type: none"> <li>• <b>Completed status</b></li> <li>• <b>Failed execution status</b></li> <li>• <b>Cancelled execution status</b></li> </ul>

- Step 5** Click **Next**.
- Step 6** On the **Add User Inputs** screen, check **Associate to Activity**.
- Step 7** From the **Activity** drop-down list, choose an activity. The user input table is updated based on the selected activity.
- Step 8** Click **Next**.

On the **Add User Outputs** screen, the user outputs are displayed based on the selected activity.

- Step 9** Click **Submit**.  
A workflow is created and is available in the **Workflows** report.

## Adding an Activity to a Workflow

An activity that is associated with one or more workflows is displayed as a custom workflow task in the **System Activity Tasks** folder of the Workflow Designer. Drag-and-drop the selected activity onto the **Workflow Designer** work area to add it to a workflow. Define the condition for controlling execution of workflows associated with an activity in the **Context Input** table of the workflow.

### Before you begin

Associate an activity with a workflow so that the activity is displayed as a custom workflow task in the **System Activity Tasks** folder of the **Workflow Designer**. For more information about how to associate an activity with a workflow, see [Creating an Activity, on page 117](#).

Open the workflow in the **Workflow Designer**.

- Step 1** Choose the activity from the **System Activity Tasks** folder.
- Step 2** Drag-and-drop the selected activity onto the **Workflow Designer** work area.  
The **Add Task** page appears.
- Step 3** In the **Task Information** screen, complete the following fields:

Name	Description
<b>Task Name</b>	The name of the task.
<b>Task Category</b>	The name of the task category (preselected information).
<b>Task Type</b>	The name of the type of task (preselected information).
<b>Comments</b>	Enter more task information.
<b>Retry Execution</b>	Check <b>Retry Execution</b> to retry a task (later) if the task fails. This feature is useful when the state of a resource is not available and a retrial of the task depends on the state.
<b>Retry Count</b>	Choose the number of retry attempts.
<b>Retry Frequency</b>	Choose the duration between retry attempts. The workflow remains on the failed task until the task succeeds or until it fails Retry Frequency number of times. It then proceeds to the next task or completion of the workflow.

- Step 4** Click **Next**.  
The **User Input Mapping** screen displays the inputs that are defined for the activity.
- Step 5** Check **Map to User Input** to use the corresponding input in the workflow.

**Step 6** In the **Context Input** table, click (+) **Add** to add a contextual input:

a) In the **Add Entry to Input Fields** dialog box, complete the following fields:

Name	Description
<b>User Inputs</b>	Choose an input to which to apply the condition.
<b>Minimum Condition list</b>	Choose the condition that you want to apply to the input.
<b>Value</b>	The value set for the condition.
<b>Associated Workflow list</b>	Choose a workflow to execute when the input satisfies the selected condition.

b) Click **Next**.

**Step 7** Click **Revalidate** to validate the current values.

**Step 8** Click **Next**.

The **User Output Mapping** screen displays the outputs that are defined for the activity.

**Step 9** Check **Map to User Output** to use the corresponding output in the workflow.

**Step 10** Click **Submit**.

## Importing and Exporting Activities

You can import and export activities using the **Import** and **Export** actions you use to import and export workflows, custom tasks, and script modules.

To import an activity, see [Importing Workflows, Custom Tasks, Script Modules, and Activities, on page 91](#).

To export an activity, see [Exporting Workflows, Custom Tasks, Script Modules, and Activities, on page 89](#).

## Deleting an Activity

You can delete an activity.

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Activities**.

**Step 3** Select the activity to delete.

**Step 4** Click **Delete**.

The **Activity** screen appears to confirm deletion of the activity.

**Step 5** Click **Delete**.





# CHAPTER 10

## Using Script Modules

This chapter contains the following sections:

- [Using Script Modules, on page 123](#)
- [Adding Script Modules, on page 124](#)
- [Adding Libraries, on page 124](#)
- [Jar Files, on page 125](#)
- [Lists of Values, on page 126](#)
- [Tabular Reports, on page 128](#)
- [Context Mapping, on page 131](#)
- [Importing and Exporting Script Modules, on page 135](#)

## Using Script Modules

A script module is essentially a container for custom scripts, jar files, and input controls. The contents of script modules enable you to perform customized actions such as adding library scripts that can be integrated with custom workflow tasks. You can export a script module and import it to a different appliance. Registered scripts in the imported module are available in the new appliance.

The following table describes the actions you can perform using script modules:

Task	See
Adding script modules	<a href="#">Adding Script Modules, on page 124</a>
Adding libraries	<a href="#">Adding Libraries, on page 124</a>
Adding jars	<a href="#">Adding Jar Files, on page 125</a>
Adding Lists of Values (LOVs)	<a href="#">Lists of Values, on page 126</a>
Adding Tabular Reports	<a href="#">Tabular Reports, on page 128</a>
Adding Context Mappings	<a href="#">Context Mapping, on page 131</a>
Exporting script modules	<a href="#">Exporting Workflows, Custom Tasks, Script Modules, and Activities, on page 89</a>

Task	See
Importing script modules	<a href="#">Importing Workflows, Custom Tasks, Script Modules, and Activities, on page 91</a>

## Adding Script Modules

To create a new script module, do the following:

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Script Module**.
  - Step 3** Click **Add**.
  - Step 4** On the **Modules Information** screen, complete the following:

Name	Description
<b>Module Name</b>	The name for the script module.
<b>Module Description</b> field (optional but recommended)	A description of the script module.

- Step 5** Click **Submit**.
- 

## Adding Libraries

To add a task library, do the following:

### Before you begin

A script module is required before you can add a library. See [Adding Script Modules, on page 124](#)

- 
- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Script Module**.
  - Step 3** In the **Script Module** pane, double-click the script module to which you want to add the library.
  - Step 4** On the **Script Module** screen, click **Library**.
  - Step 5** Click **Add**.
  - Step 6** On the **Library Information** screen, complete the following:

Name	Description
<b>Name</b>	The name for the script module.
<b>Description</b>	A description for the script module.
<b>Script</b>	Add library scripts in this area.

**Step 7** Click **Submit**.

---

## Accessing Libraries

You can access libraries in the following ways:

### Invoke a Library from Another Library

Use the following syntax to invoke a library from another library:

```
loadLibrary("Module Name"/<Library Name>")
```

For example:

```
ImportPackage(java.lang);

function test1(){
  logger.addInfo("test1");
  loadLibrary("Test_Module/testlib1");
}
test1();
```

### Invoke a Library from a Custom Task

When you create a custom workflow task, you can use the syntax shown in this example to invoke a library. Refer to the [Cisco UCS Director Custom Task Getting Started Guide](#) for more information on using custom tasks, including how to invoke a library from a custom task.

## Jar Files

You can register `.jar` files with a script module. Using the script module, you can then invoke the contents of the `.jar` file (its classes, methods, and resources) from a library or a custom task.

## Adding Jar Files

To add a `.jar` file to a script module, do the following:

---

- Step 1** Choose **Orchestration**.
  - Step 2** On the **Orchestration** page, click **Script Module**.
  - Step 3** On the **Script Modules** screen, double-click the script module to which you want to add the jar files.
  - Step 4** Click **Jars**.
  - Step 5** Click **Add**.
  - Step 6** On the **Add Jar** screen, click **Browse**.
  - Step 7** Select the `.jar` file to upload from your local folder.
  - Step 8** Click **Submit**.
-

## Lists of Values

A list of values (LOV) is a searchable list that is callable from the Cisco UCS Director GUI or as a task or workflow input. For example, when you choose a data type for a task or workflow input, the choices are presented in an LOV.

You can create an LOV to provide your own set of values for a task or workflow input, and store the LOV in a script module.

## Adding a List of Values

To create a list of values (LOV), do the following.

### Before you begin

Create a **Script Module**.

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Script Module**.
- Step 3** On the **Script Module** screen, double-click the script module you want to use.
- Step 4** Click **LOVs**.
- Step 5** Click **Add**.
- Step 6** On the **LOV Information** screen, complete the following:

Name	Description
<b>Name</b>	The name of the LOV.
<b>Description</b>	A description of the LOV (optional).
<b>Script</b>	<p>Contains the template code for the LOV provider registration. Provide your own implementation by modifying the <code>getDataProvider()</code> method as described in the next step.</p> <p><b>Note</b> Edit only the script lines described in the following steps. Do <i>not</i> edit the <code>createLOV()</code>, <code>registerLOV()</code>, or <code>registerGlobalInputs()</code> methods. Doing so could cause the LOV to fail to work.</p>

- Step 7** In the Script text box, add name-value pairs to your list as follows:

- a) Locate the `getDataProvider()` function. The function appears as follows:

```
function getDataProvider(){
var lovRegistry = LOVProviderRegistry.getInstance();
var lovProvider = new com.cloupia.service.cim.inframgr.forms.wizard.LOVProviderIf({
getLOVs : function(session) {
//provide your own implementation for Lovprovider
var formlovs = [];
```

```
// modify the following lines to add your name-value pairs:
var formlov = new FormLOVPair("Flex","1");
formlovs[0] = formlov;
var formlov = new FormLOVPair("Generic","2");
formlovs[1] = formlov;
//
return formlovs;
//End of implementation for Lovprovider
}
});
return lovProvider;
}
```

b) Modify the highlighted text to define your own name-value pairs.

**Step 8** Click **Submit**.

## Editing a List of Values

To edit an existing list of values (LOV), do the following:

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Script Module**.
- Step 3** On the **Script Module** screen, double-click the script module that contains the LOV you want to edit.
- Step 4** Click **LOVs**.
- Step 5** Choose the LOV you want to edit.
- Step 6** Click **Edit**.
- Step 7** Edit the LOV fields as described in the following table:

Name	Description
<b>Name</b>	Displays the name of the LOV.  <b>Note</b> You cannot change the name of an existing LOV.
<b>Description</b>	A description of the LOV.
<b>Script</b>	Edit the template code for the LOV provider registration. Provide your own implementation by modifying the <code>getDataProvider()</code> method as described in <a href="#">Adding a List of Values, on page 126</a> .  <b>Note</b> Do <i>not</i> edit the <code>createLOV()</code> , <code>registerLOV()</code> , or <code>registerGlobalInputs()</code> methods. Doing so could cause the LOV to fail to work.

**Step 8** Click **Submit**.

## Deleting a List of Values

### Before you begin

Remove references to this list of values (LOV) from all custom tasks and workflows.

- 
- Step 1** On the menu bar, choose **Policies > Orchestration**.
  - Step 2** Click the **Script Module** tab.
  - Step 3** In the **Script Module** pane, double-click the script module that contains the LOV you want to delete.
  - Step 4** Click the **LOVs** tab.
  - Step 5** Choose the LOV you want to delete.
  - Step 6** Click **Delete**.
  - Step 7** Click **Submit**.
- 

## Tabular Reports

A tabular report is a columnar list that is callable from the Cisco UCS Director GUI or as a task or workflow input. For example, when you choose a workflow in the **Workflow** tab, the choices are presented in tabular report.

You can create a tabular report to provide your own predefined set of values for a task or workflow input. You can store the tabular report in a script module.

## Adding a Tabular Report

### Before you begin

Create a Script Module.

- 
- Step 1** On the menu bar, choose **Policies > Orchestration**.
  - Step 2** Click the **Script Module** tab.
  - Step 3** In the **Script Module** pane, double-click the script module you want to use.
  - Step 4** Click the **Tabular Reports** tab.
  - Step 5** Click the Add (+) button.
  - Step 6** In the **Tabular Report Information** dialog box, complete the following:

Name	Description
<b>Tabular Report Name</b> field	The name of the tabular report.
<b>Description</b> field	A description of the LOV (optional).
<b>Column Entries</b> list	Click + ( <b>Add</b> ) and complete the following fields:

Name	Description
Column Name text box	The name of the column.
Column Type drop-down list	The data type of the column entries. The options are: <ul style="list-style-type: none"> <li>• Text</li> <li>• Integer</li> <li>• Long</li> <li>• Double</li> </ul>
Column Entries check box	Choose a check box to add an option to a column. The options are: <ul style="list-style-type: none"> <li>• <b>Management column</b> check box—Must be checked for exactly one column entry.</li> <li>• <b>Display column</b> check box—Must be checked for exactly one column entry.</li> <li>• <b>Hide the Field</b> check box—Check this option for any column entry.</li> </ul>

**Step 7** Click **Submit**.

**Step 8** Repeat the previous two steps for every column you want to create.

**Step 9** **Show Script** check box—Check this check box to see and edit the **Script** text area.

The column-creation script in the **Script** text area is automatically generated when you create columns.

A record consists of a row with one value for each column. Edit the **Script** text area to create records.

**Note** Do not edit or delete anything before `//START OF YOUR IMPLEMENTATION` and after `//END OF YOUR IMPLEMENTATION` in the **Script** text area.

a) Create **Record Entries**:

Create a value for each **Column Entry** defined in the **Tabular Report**. Create these values in the section of the script between the lines `//START OF YOUR IMPLEMENTATION`, and `//END OF YOUR IMPLEMENTATION`. Use the correct function depending on the **Column Type** of the **Column Entry**, as shown in the following list.

- **Text**—`model.addTextValue("value");`
- **Integer**—`model.addNumberValue(42);`
- **Long**—`model.addLongNumberValue(1000);`
- **Double**—`model.addDoubleValue(8.6);`

Separate the **Record Entry** function calls using `model.completedRow();`.

**Step 10** Click **Submit**.

### Example: Creating Record Entries for Column Entries

Suppose that you created two **Column Entries** for your tabular report. The first column entry has a column name of `Name` and column type of `Text`. The second column entry has a column name of `Department` and a column type of `Long`. Once you create the column entries in your tabular report, the system generates function calls in the script to create these columns, as follows:

```
function implementationForTabularReport(report)
{
var model = new TabularReportInternalModel();
model.addTextColumn("Name", "Name");
model.addNumberColumn("Department", "Department");
model.completedHeader();
//START OF YOUR IMPLEMENTATION.

//END OF YOUR IMPLEMENTATION.
model.updateReport(report);
}
function getSelectionColumnId(){
return "1";
}
function getDisplayColumnId(){
return "0";
}
```

A section in the middle of the script begins with `//START OF YOUR IMPLEMENTATION.` and ends with `//END OF YOUR IMPLEMENTATION.` Create record entries between these two lines. Record entries assign values to the column entries.

The following example assigns a text value of `"Smith"` to the column with column name `Name` and a long number value of `40` to the column with column name `Department`. The function call `model.completedRow();` indicates the end of this record entry.

```
function implementationForTabularReport(report)
{
var model = new TabularReportInternalModel();
model.addTextColumn("Name", "Name");
model.addLongNumberColumn("Department", "Department");
model.completedHeader();
//START OF YOUR IMPLEMENTATION.

model.addTextValue("Smith");
model.addLongNumberValue(40);
model.completedRow();

//END OF YOUR IMPLEMENTATION.
model.updateReport(report);
}
function getSelectionColumnId(){
return "1";
}
function getDisplayColumnId(){
return "0";
}
```

## Editing a Tabular Report

---

- Step 1** On the menu bar, choose **Policies > Orchestration**.
- Step 2** Click the **Script Module** tab.
- Step 3** In the **Script Module** pane, double-click the script module for which you want to edit a tabular report.
- Step 4** Choose the **Tabular Report** tab.
- Step 5** Choose the name of the tabular report you want to edit and click **Edit**.
- Step 6** In the **Edit** dialog box, edit the fields you want to change.
- Note** You can change the **Column Type** drop-down list values of a Management column or of a Display column. To uncheck the **Column Type** check box for a Management column or for a Display column, delete the existing **Column Entry** and create a new one.
- Step 7** Click **Submit**.
- 

## Deleting a Tabular Report

### Before you begin

Remove references to the tabular report from all custom workflow tasks.

---

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Script Module**.
- Step 3** On the **Script Module** screen, double-click the script module that contains the tabular report you want to delete.
- Step 4** Choose **Tabular Report**.
- Step 5** Select the name of the tabular report you want to delete.
- Step 6** Click **Delete**.
- Step 7** Click **Submit**.
- 

## Context Mapping

A context workflow mapping consists of an action label mapped to a workflow on a page, such that clicking on the action label triggers the workflow.

The **Context Mapping** module allows you to dynamically add a context workflow mapping. In this way, you can customize the Cisco UCS Director UI by creating an action label for a page and assigning a workflow to that action label.

Creating a context mapping requires knowing the name of the page to which you are adding the action label. The name, and other metadata, of a page are displayed in an **Information** dialog that is available when you enable the **Developer Menu** in Cisco UCS Director.

Enable metadata as described in the next section before you begin to create a context mapping.

## Enabling Metadata

To access report metadata, first enable the **Developer Menu**.



---

**Note** The term *report* in *Report Metadata* refers to a page in the Cisco UCS Director user interface.

---

To enable the **Developer Menu**, do the following:

- 
- Step 1** In Cisco UCS Director, hover the mouse over the user icon at the top right corner and choose **Edit My Profile** from the drop-down list.
- Step 2** On the **Edit My Profile** page, click **Show Advanced Settings**.
- Step 3** Check **Enable Developer Menu (requires re-login)**.
- The **REST API Browser** is activated in the **Orchestration** page, and the **Report Metadata** option becomes available in the report views.
- Tip** The **Advanced** area displays the REST API Access Key code for the account.
- Step 4** Click **Close**.
- 

### What to do next

View report metadata, including the report name, by clicking **Report Metadata** in any report.

## Adding a Context Mapping

### Before you begin

- Create a **Script Module**. See [Using Script Modules, on page 123](#).
- Identify the page which is to use the action label. Use the report metadata to find the name of this page; see [Enabling Metadata, on page 132](#). You use this page name when you create a **Context Mapping**.

- 
- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Script Module**.
- Step 3** On the **Script Module** page, double-click the script module you want to use.
- Step 4** Click **Context Mapping**.
- Step 5** Click **Add**.
- Step 6** On the **Add Context Mapper** screen, complete the following:

Name	Description
Name	Enter a unique name for the context mapping. The system creates a context workflow mapping that has the same name.
Report Name	Enter the name of the report (page) to which to add the action label. This name is the <i>reportName</i> field from the <b>Report Metadata</b> of the report.
Select Tabular Field	Click <b>Select</b> to display the <b>Select</b> screen. Check the tabular field you want to use.
Description	A description of the context mapping.
Script	The script that creates the context workflow mapping and associates it with an existing report.  <b>Note</b> The system updates the script with the information you enter in the fields. You do not need to edit the script.

**Step 7** Click **Submit**.

The **Status** column displays Success or Fail.

**Step 8** If the result was **Fail**, edit the **Context Mapping** to ensure that the correct **Report Name** and **Tabular Field** have been entered, then click **Submit** again.

**Step 9** If the result was **Success** then, from the menu bar, choose **Policies > Orchestration**.

**Step 10** Click **Context Workflow Mapping**.

The **Context Mapping** you created is listed in the **Mapping Name** column.

**Step 11** Click the context workflow mapping you created.

**Step 12** Click **Edit**.

**Step 13** On the **Edit Workflow Mappings** screen, click **Add Workflow**.

**Step 14** On the **Workflow 1** screen, complete the following:

Name	Description
Selection Required	If you check <b>Selection Required</b> , the <b>Workflow</b> drop-down list no longer has any values available. To make the workflow visible in the workflow drop-down list, execute <b>Context Workflow Mapping</b> on the <b>Workflow Designer</b> screen.  If you do not check this box, provide information for the remaining fields.
Report Name	Enter the name of the report (page) to add the action label to. This name is the <i>reportName</i> field from the <b>Report Metadata</b> of the report.
Action Label	Enter the name of the action label to connect to the workflow. It be a unique name.

Name	Description
<b>Workflow</b>	<p>Select the workflow to be executed when the user clicks the action label in the report.</p> <p>If you checked <b>Selection Required</b>, a new workflow is created. If you did not check <b>Selection Required</b>, choose a workflow listed in this drop-down list.</p>
<b>Selection Required</b>	<p>The script that creates the context workflow mapping and associates it with an existing report.</p> <p><b>Note</b> The system updates the script with the information you enter in the fields. You do not need to edit the script.</p>
<b>Authorized User Types</b>	<p>On the <b>Select Items</b> screen, click the each of the user types you authorize. Your choices are:</p> <ul style="list-style-type: none"> <li>• Service End-User</li> <li>• Group Admin</li> <li>• System Admin</li> <li>• Operator/Other Administrator</li> </ul>

To add an additional **Workflow** step to this **Context Workflow Mapping**, click the **Add Workflow** button and complete the fields.

To delete a **Workflow** step from this **Context Workflow Mapping**, click **Delete Field** below the workflow you want to delete.

**Step 15** If you are satisfied with your workflow mappings, click **Submit**.  
In the mapped report, the **Action Label** that you defined appears.

## Editing a Context Mapping

To edit a context mapping, do the following:

- Step 1** Choose **Orchestration**.
- Step 2** On the **Orchestration** page, click **Script Module**.
- Step 3** On the **Script Module** page, double-click the script module containing the context mapping you want to edit.
- Step 4** Choose **Context Mapping**.
- Step 5** Choose the name of the context mapping you want to edit.
- Step 6** Click **Edit**.
- Step 7** On the **Edit Context Mapper** screen, edit the fields you want to change.

You cannot edit the **Name** field.

Select a value for **Tabular Fields**. This selection is required; the previous value does not persist.

**Step 8** Click **Submit**.

---

## Deleting a Context Mapping

To delete a context mapping, use the following procedure.

### Before you begin

Remove references to the **Context Mapping** from all **Workflows** and reports.

---

**Step 1** Choose **Orchestration**.

**Step 2** On the **Orchestration** page, click **Script Module**.

**Step 3** On the **Script Module** page, double-click the script module that contains the context mapping you want to delete.

**Step 4** Choose **Context Mapping**.

**Step 5** Choose the name of the context mapping you want to delete.

**Step 6** Click **Delete**.  
A popup appears to confirm the deletion.

**Step 7** Click **Submit**.

---

## Importing and Exporting Script Modules

You can import and export script modules using the **Import** and **Export** actions you use to import and export workflows, custom tasks, and activities.

To import a script module, see [Importing Workflows, Custom Tasks, Script Modules, and Activities, on page 91](#).

To export a script module, see [Exporting Workflows, Custom Tasks, Script Modules, and Activities, on page 89](#).

