



Creating Workflows

This chapter contains the following sections:

- [Viewing the Task Library, on page 1](#)
- [The Workflow Designer, on page 1](#)
- [Creating a Workflow, on page 3](#)
- [Example: Creating a Workflow, on page 25](#)
- [Resolving Workflow Validation Issues, on page 27](#)
- [Changing Flow of Control in Workflows, on page 28](#)
- [Locking Resources, on page 37](#)
- [File Uploads, on page 39](#)

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 3](#).
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 1](#).

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

Name	Description
Workflow Context	<p>The workflow context. Cisco UCS Director Orchestrator supports the following options:</p> <ul style="list-style-type: none"> • Any—Enables you to use the workflow in any context. • Selected VM—Enables you to execute the workflow only when you choose a VM.
Save as Compound Task	<p>Check Save as Compound Task to save this version of the workflow as a compound task. See Creating a Compound Task.</p>
Publish Task outputs as Compound Task outputs	<p>Publish Task outputs as Compound Task outputs appears if you check Save as Compound Task.</p> <p>Check Publish Task outputs as Compound Task outputs 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>
Always execute during System initialization	<p>Check Always execute during System initialization to execute the workflow as part of the startup sequence when Cisco UCS Director is initialized.</p> <p>Note Because the workflow is executed before the Cisco UCS Director GUI starts, workflows that require user input cannot be executed at system startup.</p>
Disable Auto-population of inputs	<p>Check this check box to disable auto-population of input from previous service requests.</p>
Place in New Folder	<p>Create a new folder in which to place the workflow. If you check Place in New Folder, the Select Folder drop-down list becomes a text field. Type the name of the new folder.</p>
Select Folder	<p>The folder in which to place the workflow.</p> <p>Note “Effective with Cisco UCS Director 6.8, the Default Folder option is not available.</p>
Rollback Workflow on Failure	<p>Check this check box to automatically initiate a rollback of the workflow in the event of a failure.</p> <p>Selecting the Rollback Workflow on Failure 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
Abort Rollback on Failure	<p>Check this check box to abort the rollback of the workflow if the rollback request results in errors.</p> <p>Selecting the Abort Rollback on Failure check box terminates the rollback of a service request if the execution of any task within the rollback service request fails.</p>
Lock	<p>Check this check box to lock a service request.</p> <p>Selecting the Lock check box prevents archiving of the service request.</p>
Notify status of execution to initiating user	<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 Additional User(s) to send Email Notification field.</p> <p>The result status for the workflow can be one of the following:</p> <ul style="list-style-type: none"> • Completed status • Failed execution status • Cancelled execution status <p>Leave Notify status of execution to initiating user unchecked to use an email policy.</p>
Email policy	<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"> • The user that starts execution of the workflow • The user and the user's group • The user's group and that group's administrator • External email IDs <p>You can define your own policies as well by navigating to Administration > System > Orchestration Policy and selecting Add. Policies you create are available in the Email policy 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](#).

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

Name	Description
Multiline/Multiple Value Input	<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>Note 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 User Input drop-down list on the User Input Mapping page.</p>
Input Field Size	This list appears if you choose Multiline Input . Choose a size for the text area that is displayed to enter the input value.
Value Restrictions	<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"> • Admin Input—Check to populate inputs with default values. User input is then optional at run time. • Admin Input Filter—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 Admin Input Filters, on page 17. • Admin Input List—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.
Admin Input	Available when you check Admin Input . Click Select to choose one or more admin input values.
Override Admin Input Value	Available when you check Admin Input . Check Override Admin Input Value to enable the user to replace the administrator value at runtime. Leaving unchecked hides the user's input control for this value at runtime.
Input Filter Criteria	Available when you click Admin Input Filter . Type a filter expression to limit the values available at run time. See Admin Input Filters, on page 17 .
Admin Input Value	Available when you click Admin Input Filter . Displayed for reference—Use the column headings as attribute names in your filter expression.
Select	Available when you click Admin Input List . Click Select , 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 10](#).

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
Output Label	The name or label of the output parameter.
Output Description	A description of the output parameter.
Optional	Make the output parameter optional (not mandatory) at the time of workflow execution.
Type	The data type of the output parameter. During the editing and saving of a workflow task, the Type 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 19](#).

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 10](#). 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](#).

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 10](#).



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 15](#).

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 8](#). 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 3](#). 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 8](#).

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](#).

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 3](#).

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:

- Choose **Orchestration**.
- On the **Orchestration** page, click **Workflows**.
- Open the **Examples** folder.
- Select the **Example 1 - Workflow Rules - Firewall Config** workflow.
- Click the **Edit** action to edit the workflow.
- 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:

- 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 12](#).

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 11](#).

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

- Choose **Orchestration**.
- On the **Orchestration** page, click **Workflows**.
- 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 13](#).

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 11](#) and [Example: Using Cascading Workflow Input Processing Rules, on page 12](#).

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 3 - Workflow Rules - Notification Timeouts** 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, **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:

- a) Again choose the **Input Rules 3 - Workflow Rules - Notification Timeouts** rule.
- b) Choose the **Execute Now** action.
- c) On the **Executing Workflow** page, click **Select** to open the workflow selection table.
- d) 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 11](#), [Example: Using Cascading Workflow Input Processing Rules, on page 12](#).

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](#).
- 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"> • EQUALS • NOT_EQUALS • CONTAINS • NOT_CONTAINS • IS_SUBSET • NOT_SUBSET <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 1](#).



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
Task Name	A name for this instance of the task. You can accept the automatically generated default or enter your own name for the task.
Task Category	The category (also the folder name) in which the task appears. Cannot be changed.
Task Type	The type (or generic name) of the task. Cannot be changed.
Comment	An optional comment about this use of the task in this workflow.
Retry Execution	Check Retry Execution to retry the task if it fails. The Retry Count and Retry Frequency controls appear. Retry Execution 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.
Disable Rollback	Check Disable Rollback to disable the task's rollback script. If you check this box, the task is unchecked in the Rollback Service Request dialog and cannot be rolled back; see Rolling Back a Service Request . If the task does not support rollback, checking Disable Rollback has no effect.
Retry Count	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
Retry Frequency	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 1, 10, 60 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
Manage Workflow User Inputs	Click Manage Workflow User Inputs to add to or edit the workflow user inputs.
Task Input	Each task input is displayed as a heading on the remainder of this page. The input is labeled (Mandatory) if it required.
Map to User Input	Check Map to User Input to populate the task input from a workflow user input at runtime.
User Input	This drop-down list appears if you check Map to User Input . Select the workflow user input to map to the task input. Note You cannot map a multiline workflow input to a single-line task input. The multiline workflow input does not appear in the User Input drop-down when creating or editing a task. See Defining Workflow Input and Output , on page 6.

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
Task Input	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
Task Output	Each task output is displayed as a heading on this page. The output is labeled (Mandatory) if it required.

Name	Description
Map to User Output	Check Map to User Output to populate a workflow user output from the task output at runtime.
User Output	This drop-down list appears if you check Map to User Output . 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 24](#).

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 19](#) and [Connecting a Task to a Workflow, on page 22](#).

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 19](#).

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 22](#).

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 27](#).

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

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

- 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 Map to User Input . 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 1](#).

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 Retry Execution 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 List based iteration 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 Start Loop task here.
Count based iteration check box	Check Count based iteration 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 25](#).

- 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 25](#).
 - 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
Task Name	Type StartVMLoop .
Comment	Type Loop through a list of VMs .
Task Details	Make a note of the message under Task Details regarding the task output START_LOOP_OUTPUT .

- 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
Input Label	Type VM Names .
Input Description	Type List of VMs to restart .
Optional	Leave Optional unchecked.

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

- 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 EndVMLoop .
Comment	Type End of VM loop .

- 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
Task Name	The name of the task.
Comment field	Comments for the task.
Retry Execution	Check Retry Execution to retry later if the task fails.
Retry count	Choose the number of retry attempts.
Retry Frequency	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
Specify the condition	<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: ==, !=, <, <=, >, >=, , &&, contains, startsWith, endsWith</p> <p>Following are some example conditions:</p> <ul style="list-style-type: none"> • <code>TaskName.OUTPUT_ATTRIBUTE_NAME == "2"</code> • <code>WORKFLOW_USERINPUT_LABEL_NAME contains "xyz"</code> • <code>TaskName.OUTPUT_ATTRIBUTE_NAME == WORKFLOW_USERINPUT_LABEL_NAME WORKFLOW_USERINPUT_LABEL_NAME != "123"</code>

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
Task Name	The name of the task.
Comment	Comments for the task.
Retry Execution	Check Retry Execution to retry later if the task fails.
Retry count	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>==, !=, <, <=, >, >=, , &&, contains, startsWith, endsWith</p> <p>Following are some examples that you can use while specifying conditions:</p> <ul style="list-style-type: none"> • TaskName.OUTPUT ATTRIBUTE NAME=="2" • WORKFLOW USERINPUT LABEL NAME contains "xyz" • TaskName.OUTPUT ATTRIBUTE NAME == WORKFLOW USERINPUT LABEL NAME WORKFLOW USERINPUT LABEL NAME != "123"

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 19](#). Complete the following fields:

Name	Description
Task Name	The task name.
Comment	A comment about the approval task (Optional.)
Retry Execution	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:.

Name	Description
User ID list	From the User ID list, choose one or more users whose approval is required to execute this workflow. Then click Submit .
Approval required from all the users	Check Approval required from all the users 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.
Number of approval request Reminders	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.
Reminder Interval (Hours)	Enter the number of hours between approval reminders.
Use Approval Email Template	Check Use Approval Email Template to select a template for the approval emails. Leave unchecked to use a default Approval template.

Step 7

Name	Description
<p>Select Email Template</p>	<p>Visible if Use Approval Email Template is checked. Select an email template to use.</p> <p>You can add, delete, and update email templates. See Working with Email Templates in the Cisco UCS Director Administration Guide.</p> <p>Note The Select Email Template 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>
<p>Subject</p>	<p>Visible if Use Approval Email Template is checked. The email subject line is displayed. You can edit this text.</p>
<p>Body</p>	<p>Visible if Use Approval Email Template is checked. The HTML text to output the email body text is displayed. You can edit this text.</p>
<p>Use Approval Email Template for Rollback</p>	<p>Check Use Approval email template during Rollback to select a template for emails requesting approval for rollback of the workflow. Leave unchecked to use a default Approval template.</p>
<p>Select Email Template</p>	<p>Visible if Use Approval Email Template during Rollback is checked. Select an email template to use.</p> <p>You can add, delete, and update email templates. See Working with Email Templates in the Cisco UCS Director Administration Guide.</p> <p>Note The Select Email Template 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>
<p>Subject</p>	<p>Visible if Use Approval Email Template during Rollback is checked. The email subject line is displayed. You can edit this text.</p>
<p>Body</p>	<p>Visible if Use Approval Email Template during Rollback is checked. The HTML text to output the email body text is displayed. You can edit this text.</p>

Name	Description
Reset to Default	<p>Visible beneath the Body text area in either of Use Approval Email Template or Use Approval Email Template for Rollback. Reset to Default has two effects:</p> <ul style="list-style-type: none"> • It immediately discards any changes you have made to the Subject and Body text and restores the selected template to its system default. • It ensures that the system supplied version of the selected template is used for the approval email in the workflow you are editing. <p>To save edits to a template, make changes in the Subject or Body text areas and be careful <i>not</i> to check Reset to Default.</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](#)) 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 Retry Execution 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 3](#).
- Step 4** Complete the fields on the **Add Workflow Details** page. See [Defining a Workflow, on page 3](#)
- Step 5** On the **Add User Input** page, add a new input. See [Defining Workflow Input and Output, on page 6](#).

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 6](#).
-

What to do next

Continue to construct your workflow. See [Connecting a Task to a Workflow, on page 22](#). 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 3](#).
- Step 4** Complete the fields on the **Add Workflow Details** page. See [Defining a Workflow, on page 3](#)
- Step 5** On the **Add User Input** page, add a new input. See [Defining Workflow Input and Output, on page 6](#).
- 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 6](#).
- 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 22](#). 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 39](#). 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.

