



# Orchestration Concepts

---

This chapter contains the following sections:

- [Introduction to UCS Director Orchestration, on page 1](#)
- [Tasks, on page 1](#)
- [Workflows, on page 2](#)
- [Service Requests, on page 2](#)
- [Input and Output, on page 2](#)
- [Workflow Validation, on page 3](#)
- [Workflow Versioning, on page 4](#)
- [Approvals, on page 4](#)
- [Rollback, on page 4](#)
- [Libraries and Catalogs, on page 5](#)
- [Activities, on page 5](#)
- [Managing the Workflow Lifecycle, on page 6](#)

## 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](#). 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 4](#)), completed, or failed (a service request can fail when one of its component tasks fails to execute properly; see [Rollback, on page 4](#) and [Resubmitting a Service Request](#)).

## 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](#).

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](#).

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](#) 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](#).

# 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</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</a> and <a href="#">Connecting a Task to a Workflow</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</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</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</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</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</a> .
<b>Execute a Workflow</b>	Create a service request immediately from the selected workflow. See <a href="#">Executing a Workflow</a> .
<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> .
<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> .

Operation	Description
<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> .
<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> .
<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> .
<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> .

