



Customizing Workflow Components

This chapter contains the following sections:

- [Creating a Compound Task, page 1](#)
- [Creating Custom Approvals, page 3](#)
- [Creating Custom Inputs, page 4](#)
- [Macros, page 5](#)

Creating a Compound Task

A compound task is a workflow that functions as a single task. A compound task, like any other task, is atomic; its component tasks are hidden.

You create a compound task by saving a workflow as a compound task when you create or edit the workflow. Do this, for example, if you find yourself building the same series of tasks into several different workflows.

You can define a simple workflow and save it as a compound task, then define another workflow that incorporates the compound task. You can use this pattern to define increasingly complex workflows.

To save an existing workflow as a compound task, do the following:



Note To create a new compound task from scratch, see [Creating a Workflow](#)

-
- Step 1** On the menu bar, choose **Policies > Orchestration**.
 - Step 2** Choose the **Workflow** tab.
 - Step 3** Select a workflow to save as a compound task.
 - Step 4** Click **Edit**.
 - Step 5** Check the **Save as Compound Task** check box.
 - Step 6** If you want all of the workflow's task outputs available as output of the compound task, click the **Publish Task outputs as Compound Task outputs** check box.
 - Step 7** Click **Next**.
 - Step 8** On the **Add User Inputs** screen, click **Next**.
 - Step 9** On the **Add User Outputs** screen, click **Submit**.
-

The new compound task is available in the **Compound Task** folder when you open the **Workflow Designer**.

Example: Creating a Compound Task

This example demonstrates repeating a workflow task for elements in a list.

Before You Begin

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

-
- Step 1** Navigate to **Policies > Orchestration**.
 - Step 2** Click the **Workflows** tab.
 - Step 3** Locate and select the PowerCycleVM workflow you created in [Example: Creating a Workflow](#).
 - Step 4** Click **Edit**.
 - Step 5** In the **Edit Workflow Details** window, check the **Save as Compound Task** check box.
 - Note** None of the tasks has an output, so ignore the **Publish Task Outputs as Compound Task outputs** check box. The workflow has nothing to do with system startup of Cisco UCS Director, so ignore also the **Always execute during System initialization** check box.
 - Step 6** Click through to the **Edit Workflow Output** page.
 - Step 7** Click **Submit**.
-

What to Do Next

Include the custom task in other workflows. For example, you can put this task before the Completed (Failed) task of workflows that modify to remotely hosted VMs. Then, the VM restarts if a modification fails.

Creating Custom Approvals

You can create custom approval tasks that allow approvers to enter input values at the time of workflow execution.

Step 1 On the menu bar, choose **Policies > Orchestration**.

Step 2 Choose the **Custom Approval Tasks** tab.

Step 3 Click **Add**.

Step 4 In the **Add Inputs** screen, complete the following fields:

Name	Description
Approval Task Name field	The name of the approval task as it appears in the Workflow Designer.
Approval Task Description field	The description of the approval task (optional).

Step 5 Click the **Add Input Field** button.

Step 6 Under the **User Input** heading for the new input field, complete the following fields:

Name	Description
Input Label field	The label for the input (supplied by the approver of the task).
Input Description field	A description of the input.
Input Type drop-down	The data type of the input.
Optional Input check box	If checked, the administrator must provide a default value for the input. The approver is not required to provide input.

Step 7 Repeat the previous two steps to add as many inputs as needed.

Step 8 Click **Submit**.

What to Do Next

You can now include the custom task in a workflow.

Creating Custom Inputs

You can create custom input types to use as workflow inputs. Custom input types are based on an existing input type. They are defined by filter criteria or by a selection set that further narrows the possible values of the input.

Step 1 On the menu bar, choose **Policies > Orchestration**.

Step 2 Click the **Custom Workflow Inputs** tab.

Step 3 Click **Add**.

Step 4 In the **Add Custom Workflow Input** screen, complete the following fields:

Name	Description
Custom Input Type Name field	The input name.
Input Type button	Pressing this button brings up a list of existing input types. From the list, choose an input type on which to base your input type.
Filter check boxes	Depending on your choice of input type, one or more of the following filter types is available: <ul style="list-style-type: none"> • Input Filter—A text field. Type a text filter string. • Input List—A list of values. Choose which existing values are valid instances for this input. • Input LOV—Define a list of allowable name-value pairs for the input. <p>Note The Label field and Value field descriptions should match.</p> • Input Range—A text field. Type a range of valid character values. • Validated Input—Choose a validator type from the table.

Step 5 Click the (+) **Add** icon.

Step 6 Click **Submit**.

The new input type is added to the **Custom Workflow Types** page. The new input type is available for selection when defining workflow and task inputs.

Macros

Macro variables, or macros, are variables that you can use within Cisco UCS Director Orchestrator.

Macros enable you to include several types of variable system information in two places:

- In task input variables inside a workflow, where you can access such information as:
 - Workflow inputs and task outputs
 - Service request IDs
 - VM information such as ID, name, IP address, and power state
- In VM names, where you can access such information as:
 - User information such as group and user IDs
 - Configuration information such as catalog and system profile
 - Deployment information such as cloud name and location

Orchestration Macros

Several *macros* can be accessed in the context of a workflow by workflow tasks. When you create a Cisco UCS Director workflow, you can use macros in any of the workflow's task inputs. An input field can contain any combination of text and macros. During execution of the workflow, Cisco UCS Director Orchestrator substitutes the macros' values into each task's inputs before executing the task. Macros available for use in task inputs are described in the following sections.

Input and Output Macros

Any workflow-level input or previous task output can be used as a macro in a subsequent task. For example, consider a workflow that has two inputs labeled *Enter Disk Size* and *Max Snapshots*. Suppose that the workflow has two tasks with IDs *task1* and *task2* (arranged so that *task1* executes first). Any input values to *task1* or *task2* that takes free-form input can use the following two macros:

- `${Enter Disk Size}`
- `${Max Snapshots}`

**Note**

Macros consist of a dollar sign (\$) followed by the macro name in curly braces ({}). A workflow level input can be used as a macro by including the label associated with the user input in the macro definition.

Also the second task, *task2*, can use the output of *task1*. If *task1* has two output variables, *OUTPUT_VOLUME_NAME* and *OUTPUT_VOLUME_SIZE*, then *task2* can use the following macros to capture their values in its inputs:

- `${task1.OUTPUT_VOLUME_NAME}`

- `${task1.OUTPUT_VOLUME_SIZE}`

**Note**

The macro name for a task output is the task name, followed by a period, followed by the task output name: `${taskName.outputName}`.

Service Request Macros

In addition to workflow inputs and task outputs, the following macros representing service requests are available:

- `${SR_ID}`—The ID of the current service request
- `${PARENT_SR_ID}`—The ID of the service request that is the parent of the current service request. (Available only if the current service request has a parent.)

Virtual Machine Macros

For workflows that are executed in the context of a VM, more VM macros are available. VM macros cannot be used in a non-VM context.

For the full list of VM macros, see [List of VM Macros and VM Annotations](#), on page 6.

VM Annotations

VM annotations represent information about a VM. You add these variables to a VMware system policy if you choose to define VM annotations in that policy. The output from these variables displays in the Annotations field of the VM in VMware vCenter.

For the full list of VM annotations, see [List of VM Macros and VM Annotations](#), on page 6.

List of VM Macros and VM Annotations

The syntax that you use for VM macros can be different from the syntax used for VM annotations. In addition, more variables are available for VM macros than for VM annotations. The following table shows the correct syntax for VM macros and for VM annotations. If a cell contains N/A, that variable is not available in the context.

For information about the variables that you can use in the **VM Name Template** and **VM Host Name Template** fields of the system policy, see the [Cisco UCS Director Administration Guide](#).

**Note**

This table does not include a complete list of Cloupiascript macros. For information on using Cloupiascript macros, see the [Cisco UCS Director Cloupiascript Cookbook](#).

Variables	VM Macros for Orchestration Workflows	VM Annotations for System Policies
VM name	<code>\${VM_NAME}</code>	<code>\${VMNAME}</code>
VM IP address	<code>\${VM_IPADDRESS}</code>	N/A
VM state (can be either <i>on</i> or <i>off</i>)	<code>\${VM_STATE}</code>	N/A
VM state details (can be either <i>power-on</i> or <i>power-off</i>)	<code>\${VM_STATE_DETAILS}</code>	N/A
ESX server or host node that hosts the VM	<code>\${VM_PARENT}</code>	N/A
Cloud used for VM provisioning	<code>\${VM_CLOUD}</code>	<code>\${CLOUD_NAME}</code>
Type of cloud	N/A	<code>\${CLOUD_TYPE}</code>
VM hostname	<code>\${VM_HOSTNAME}</code>	N/A
Short VM hostname	<code>\${VM_HOSTNAME_SHORT}</code>	N/A
VM hostname and domain	<code>\${VM_HOSTNAME_DOMAIN}</code>	N/A
Name of the group to which the VM belongs	<code>\${VM_GROUP_NAME}</code>	<code>\${GROUP_NAME}</code>
Full name of the group	N/A	<code>\${FULL_GROUP_NAME}</code>
ID of the group	<code>\${VM_GROUP_ID}</code>	N/A
Name of the parent group, if one exists	N/A	<code>\${GROUP_PARENT}</code>
ID of the catalog used to provision the VM	<code>\${VM_CATALOG_ID}</code>	N/A
Name of the catalog used to provision the VM	N/A	<code>\${CATALOG_NAME}</code>
VM ID	<code>\${VM_ID}</code>	N/A
Service request ID for the VM	<code>\${VM_SR_ID}</code>	<code>\${SR_ID}</code>
Comments from the user who requested the VM	<code>\${VM_COMMENTS}</code>	<code>\${COMMENTS}</code>
Virtual data center name	<code>\${VM_VDC_NAME}</code>	N/A
Virtual data center ID	<code>\${VM_VDC_ID}</code>	N/A

Variables	VM Macros for Orchestration Workflows	VM Annotations for System Policies
Type of VM	<code>\${VM_TYPE}</code>	N/A
Scheduled termination of the VM	<code>\${VM_SCHED_TERM}</code>	N/A
Location specified in the account	N/A	<code>\${LOCATION}</code>
Cost center for the VM	N/A	<code>\${COST_CENTER}</code>
Current time in milliseconds converted to a unique ID for the VM	N/A	<code>\${UNIQUE_ID}</code>
User who requested the VM	N/A	<code>\${USER}</code>
Full name of the user who requested the VM	N/A	<code>\${FULL_USER_NAME}</code>
Appcode from the catalog	N/A	<code>\${APPCODE}</code>
Name of the system policy associated with the application category	N/A	<code>\${PROFILE_NAME}</code>
Name of the user who initiated the request	N/A	<code>\${INITIATING_USER}</code>
Simple name of the user who initiated the request	N/A	<code>\${INITIATING_USER_SIMPLE_NAME}</code>
Email address of the user who submitted the request	N/A	<code>\${SUBMITTER_EMAIL}</code>
The ID of the user who submitted the request	N/A	<code>\${SUBMITTER_USERID}</code>
First name of the user who submitted the request	N/A	<code>\${SUBMITTER_FIRSTNAME}</code>
Last name of the user who submitted the request	N/A	<code>\${SUBMITTER_LASTNAME}</code>
The role of the user who submitted the request	N/A	<code>\${SUBMITTER_ROLE}</code>
Name of the group to which the user who submitted the request belongs	N/A	<code>\${SUBMITTER_GROUPNAME}</code>

Variables	VM Macros for Orchestration Workflows	VM Annotations for System Policies
The ID of the group to which the user who submitted the request belongs	N/A	<code>\${SUBMITTER_GROUPID}</code>

