Using Script Modules

This chapter contains the following sections:

- Using Script Modules, on page 1
- Adding Script Modules, on page 2
- Adding Libraries, on page 2
- Jar Files, on page 3
- Lists of Values, on page 4
- Tabular Reports, on page 6
- Context Mapping, on page 9
- Importing and Exporting Script Modules, on page 13

Using Script Modules

A script module is essentially a container for custom scripts, jar files, and input controls. The contents of script modules enable you to perform customized actions such as adding library scripts that can be integrated with custom workflow tasks. You can export a script module and import it to a different appliance. Registered scripts in the imported module are available in the new appliance.

The following table describes the actions you can perform using script modules:

Task	See
Adding script modules	Adding Script Modules, on page 2
Adding libraries	Adding Libraries, on page 2
Adding jars	Adding Jar Files, on page 3
Adding Lists of Values (LOVs)	Lists of Values, on page 4
Adding Tabular Reports	Tabular Reports, on page 6
Adding Context Mappings	Context Mapping, on page 9
Exporting script modules	Exporting Workflows, Custom Tasks, Script Modules, and Activities

Task	See
Importing script modules	Importing Workflows, Custom Tasks, Script Modules, and Activities

Adding Script Modules

To create a new script module, do the following:

Step 1 Choose **Orchestration**.

Step 2 On the **Orchestration** page, click **Script Module**.

Step 3 Click **Add**.

Step 4 On the **Modules Information** screen, complete the following:

Name	Description
Module Name	The name for the script module.
Module Description field (optional but recommended)	A description of the script module.

Step 5 Click **Submit**.

Adding Libraries

To add a task library, do the following:

Before you begin

A script module is required before you can add a library. See Adding Script Modules, on page 2

Step 1 Choose **Orchestration**.

Step 2 On the **Orchestration** page, click **Script Module**.

Step 3 In the **Script Module** pane, double-click the script module to which you want to add the library.

Step 4 On the **Script Module** screen, click **Library**.

Step 5 Click **Add**.

Step 6 On the **Library Information** screen, complete the following:

Name	Description
Name	The name for the script module.
Description	A description for the script module.
Script	Add library scripts in this area.

Step 7 Click **Submit**.

Accessing Libraries

You can access libraries in the followings ways:

Invoke a Library from Another Library

Use the following syntax to invoke a library from another library:

```
loadLibrary("Module Name>/<Library Name>")
```

For example:

```
ImportPackage(java.lang);

function test1(){
 logger.addInfo("test1");
 loadLibrary("Test_Module/testlib1");
}
test1();
```

Invoke a Library from a Custom Task

When you create a custom workflow task, you can use the syntax shown in this example to invoke a library. Refer to the Cisco UCS Director Custom Task Getting Started Guide for more information on using custom tasks, including how to invoke a library from a custom task.

Jar Files

You can register `.jar` files with a script module. Using the script module, you can then invoke the contents of the `.jar` file (its classes, methods, and resources) from a library or a custom task.

Adding Jar Files

To add a `.jar` file to a script module, do the following:

Step 1 Choose **Orchestration**.
Step 2 On the **Orchestration** page, click **Script Module**.
Step 3 On the **Script Modules** screen, double-click the script module to which you want to add the jar files.
Step 4 Click **Jars**.
Step 5 Click **Add**.
Step 6 On the **Add Jar** screen, click **Browse**.
Step 7 Select the `.jar` file to upload from your local folder.
Step 8 Click **Submit**.

Lists of Values

A list of values (LOV) is a searchable list that is callable from the Cisco UCS Director GUI or as a task or workflow input. For example, when you choose a data type for a task or workflow input, the choices are presented in an LOV.

You can create an LOV to provide your own set of values for a task or workflow input, and store the LOV in a script module.

Adding a List of Values

To create a list of values (LOV), do the following.

Before you begin

Create a **Script Module**.

Step 1 Choose **Orchestration**.

Step 2 On the **Orchestration** page, click **Script Module**.

Step 3 On the **Script Module** screen, double-click the script module you want to use.

Step 4 Click **LOVs**.

Step 5 Click **Add**.

Step 6 On the **LOV Information** screen, complete the following:

Name	Description
Name	The name of the LOV.
Description	A description of the LOV (optional).
Script	Contains the template code for the LOV provider registration. Provide your own implementation by modifying the `getDataProvider()` method as described in the next step. **Note** Edit only the script lines described in the following steps. Do *not* edit the `createLOV()`, `registerLOV()`, or `registerGlobalInputs()` methods. Doing so could cause the LOV to fail to work.

Step 7 In the Script text box, add name-value pairs to your list as follows:

a) Locate the `getDataProvider()` function. The function appears as follows:

```
function getDataProvider(){
var lovRegistry = LOVProviderRegistry.getInstance();
var lovProvider = new com.cloupia.service.cIM.inframgr.forms.wizard.LOVProviderIf({
getLOVs : function(session) {
//provide your own implementation for Lovprovider
var formlovs = [];
```

```
// modify the following lines to add your name-value pairs:
var formlov = new FormLOVPair("Flex","1");
formlovs[0] = formlov;
var formlov = new FormLOVPair("Generic","2");
formlovs[1] = formlov;
//
return formlovs;
//End of implementation for Lovprovider
}
});
return lovProvider;
}
```

b) Modify the highlighted text to define your own name-value pairs.

Step 8 Click **Submit**.

Editing a List of Values

To edit an existing list of values (LOV), do the following:

Step 1 Choose **Orchestration**.

Step 2 On the **Orchestration** page, click **Script Module**.

Step 3 On the **Script Module** screen, double-click the script module that contains the LOV you want to edit.

Step 4 Click **LOVs**.

Step 5 Choose the LOV you want to edit.

Step 6 Click **Edit**.

Step 7 Edit the LOV fields as described in the following table:

Name	Description
Name	Displays the name of the LOV. **Note** You cannot change the name of an existing LOV.
Description	A description of the LOV.
Script	Edit the template code for the LOV provider registration. Provide your own implementation by modifying the `getDataProvider()` method as described in Adding a List of Values, on page 4. **Note** Do *not* edit the `createLOV()`, `registerLOV()`, or `registerGlobalInputs()` methods. Doing so could cause the LOV to fail to work.

Step 8 Click **Submit**.

Deleting a List of Values

Before you begin

Remove references to this list of values (LOV) from all custom tasks and workflows.

Step 1 On the menu bar, choose **Policies** > **Orchestration**.
Step 2 Click the **Script Module** tab.
Step 3 In the **Script Module** pane, double-click the script module that contains the LOV you want to delete.
Step 4 Click the **LOVs** tab.
Step 5 Choose the LOV you want to delete.
Step 6 Click **Delete**.
Step 7 Click **Submit**.

Tabular Reports

A tabular report is a columnar list that is callable from the Cisco UCS Director GUI or as a task or workflow input. For example, when you choose a workflow in the **Workflow** tab, the choices are presented in tabular report.

You can create a tabular report to provide your own predefined set of values for a task or workflow input. You can store the tabular report in a script module.

Adding a Tabular Report

Before you begin

Create a Script Module.

Step 1 On the menu bar, choose **Policies** > **Orchestration**.
Step 2 Click the **Script Module** tab.
Step 3 In the **Script Module** pane, double-click the script module you want to use.
Step 4 Click the **Tabular Reports** tab.
Step 5 Click the Add **(+)** button.
Step 6 In the **Tabular Report Information** dialog box, complete the following:

Name	Description
Tabular Report Name field	The name of the tabular report.
Description field	A description of the LOV (optional).
Column Entries list	Click **+ (Add)** and complete the following fields:

Name	Description
Column Name text box	The name of the column.
Column Type drop-down list	The data type of the column entries. The options are: • Text • Integer • Long • Double
Column Entries check box	Choose a check box to add an option to a column. The options are: • **Management column** check box—Must be checked for exactly one column entry. • **Display column** check box—Must be checked for exactly one column entry. • **Hide the Field** check box—Check this option for any column entry.

Step 7 Click **Submit**.

Step 8 Repeat the previous two steps for every column you want to create.

Step 9 **Show Script** check box—Check this check box to see and edit the **Script** text area.

The column-creation script in the **Script** text area is automatically generated when you create columns.

A record consists of a row with one value for each column. Edit the **Script** text area to create records.

Note Do not edit or delete anything before `//START OF YOUR IMPLEMENTATION` and after `//END OF YOUR IMPLEMENTATION` in the **Script** text area.

a) Create **Record Entries**:

Create a value for each **Column Entry** defined in the **Tabular Report**. Create these values in the section of the script between the lines `//START OF YOUR IMPLEMENTATION.` and `//END OF YOUR IMPLEMENTATION.`. Use the correct function depending on the **Column Type** of the **Column Entry**, as shown in the following list.

- Text—`model.addTextValue("value");`
- Integer—`model.addNumberValue(42);`
- Long—`model.addLongNumberValue(1000);`
- Double—`model.addDoubleValue(8.6);`

Separate the **Record Entry** function calls using `model.completedRow();`.

Step 10 Click **Submit**.

Example: Creating Record Entries for Column Entries

Suppose that you created two **Column Entries** for your tabular report. The first column entry has a column name of `Name` and column type of `Text`. The second column entry has a column name of `Department` and a column type of `Long`. Once you create the column entries in your tabular report, the system generates function calls in the script to create these columns, as follows:

```
function implementationForTabularReport(report)
{
var model = new TabularReportInternalModel();
model.addTextColumn("Name","Name");
model.addNumberColumn("Department","Department");
model.completedHeader();
//START OF YOUR IMPLEMENTATION.



//END OF YOUR IMPLEMENTATION.
model.updateReport(report);
}
function getSelectionColumnId(){
return "1";
}
function getDisplayColumnId(){
return "0";
}
```

A section in the middle of the script begins with `//START OF YOUR IMPLEMENTATION.` and ends with `//END OF YOUR IMPLEMENTATION.` Create record entries between these two lines. Record entries assign values to the column entries.

The following example assigns a text value of `"Smith"` to the column with column name `Name` and a long number value of `40` to the column with column name `Department`. The function call `model.completedRow();` indicates the end of this record entry.

```
function implementationForTabularReport(report)
{
var model = new TabularReportInternalModel();
model.addTextColumn("Name","Name");
model.addLongNumberColumn("Department","Department");
model.completedHeader();
//START OF YOUR IMPLEMENTATION.

model.addTextValue("Smith");
model.addLongNumberValue(40);
model.completedRow();

//END OF YOUR IMPLEMENTATION.
model.updateReport(report);
}
function getSelectionColumnId(){
return "1";
}
function getDisplayColumnId(){
return "0";
}
```

Editing a Tabular Report

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

Step 2 Click the **Script Module** tab.

Step 3 In the **Script Module** pane, double-click the script module for which you want to edit a tabular report.

Step 4 Choose the **Tabular Report** tab.

Step 5 Choose the name of the tabular report you want to edit and click **Edit**.

Step 6 In the **Edit** dialog box, edit the fields you want to change.

Note You can change the **Column Type** drop-down list values of a Management column or of a Display column. To uncheck the **Column Type** check box for a Management column or for a Display column, delete the existing **Column Entry** and create a new one.

Step 7 Click **Submit**.

Deleting a Tabular Report

Before you begin

Remove references to the tabular report from all custom workflow tasks.

Step 1 Choose **Orchestration**.

Step 2 On the **Orchestration** page, click **Script Module**.

Step 3 On the **Script Module** screen, double-click the script module that contains the tabular report you want to delete.

Step 4 Choose **Tabular Report**.

Step 5 Select the name of the tabular report you want to delete.

Step 6 Click **Delete**.

Step 7 Click **Submit**.

Context Mapping

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

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

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

Enable metadata as described in the next section before you begin to create a context mapping.

Enabling Metadata

To access report metadata, first enable the **Developer Menu**.

Note The term *report* in *Report Metadata* refers to a page in the Cisco UCS Director user interface.

To enable the **Developer Menu**, do the following:

Step 1 In Cisco UCS Director, hover the mouse over the user icon at the top right corner and choose **Edit My Profile** from the drop-down list.

Step 2 On the **Edit My Profile** page, click **Show Advanced Settings**.

Step 3 Check **Enable Developer Menu (requires re-login)**.

The **REST API Browser** is activated in the **Orchestration** page, and the **Report Metadata** option becomes available in the report views.

Tip The **Advanced** area displays the REST API Access Key code for the account.

Step 4 Click **Close**.

What to do next

View report metadata, including the report name, by clicking **Report Metadata** in any report.

Adding a Context Mapping

Before you begin

- Create a **Script Module**. See Using Script Modules, on page 1.
- Identify the page which is to use the action label. Use the report metadata to find the name of this page; see Enabling Metadata, on page 10. You use this page name when you create a **Context Mapping**.

Step 1 Choose **Orchestration**.

Step 2 On the **Orchestration** page, click **Script Module**.

Step 3 On the **Script Module** page, double-click the script module you want to use.

Step 4 Click **Context Mapping**.

Step 5 Click **Add**.

Step 6 On the **Add Context Mapper** screen, complete the following:

Name	Description
Name	Enter a unique name for the context mapping. The system creates a context workflow mapping that has the same name.
Report Name	Enter the name of the report (page) to which to add the action label. This name is the *reportName* field from the **Report Metadata** of the report.
Select Tabular Field	Click **Select** to display the **Select** screen. Check the tabular field you want to use.
Description	A description of the context mapping.
Script	The script that creates the context workflow mapping and associates it with an existing report. **Note** The system updates the script with the information you enter in the fields. You do not need to edit the script.

Step 7 Click **Submit**.
The **Status** column displays Success or Fail.

Step 8 If the result was **Fail**, edit the **Context Mapping** to ensure that the correct **Report Name** and **Tabular Field** have been entered, then click **Submit** again.

Step 9 If the result was **Success** then, from the menu bar, choose **Policies** > **Orchestration**.

Step 10 Click **Context Workflow Mapping**.
The **Context Mapping** you created is listed in the **Mapping Name** column.

Step 11 Click the context workflow mapping you created.

Step 12 Click **Edit**.

Step 13 On the **Edit Workflow Mappings** screen, click **Add Workflow**.

Step 14 On the **Workflow 1** screen, complete the following:

Name	Description
Selection Required	If you check **Selection Required**, the **Workflow** drop-down list no longer has any values available. To make the workflow visible in the workflow drop-down list, execute **Context Workflow Mapping** on the **Workflow Designer** screen. If you do not check this box, provide information for the remaining fields.
Report Name	Enter the name of the report (page) to add the action label to. This name is the *reportName* field from the **Report Metadata** of the report.
Action Label	Enter the name of the action label to connect to the workflow. It be a unique name.

Name	Description
Workflow	Select the workflow to be executed when the user clicks the action label in the report. If you checked **Selection Required**, a new workflow is created. If you did not check **Selection Required**, choose a workflow listed in this drop-down list.
Selection Required	The script that creates the context workflow mapping and associates it with an existing report. **Note** The system updates the script with the information you enter in the fields. You do not need to edit the script.
Authorized User Types	On the **Select Items** screen, click the each of the user types you authorize. Your choices are: • Service End-User • Group Admin • System Admin • Operator/Other Administrator

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

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

Step 15 If you are satisfied with your workflow mappings, click **Submit**.
In the mapped report, the **Action Label** that you defined appears.

Editing a Context Mapping

To edit a context mapping, do the following:

Step 1 Choose **Orchestration**.
Step 2 On the **Orchestration** page, click **Script Module**.
Step 3 On the **Script Module** page, double-click the script module containing the context mapping you want to edit.
Step 4 Choose **Context Mapping**.
Step 5 Choose the name of the context mapping you want to edit.
Step 6 Click **Edit**.
Step 7 On the **Edit Context Mapper** screen, edit the fields you want to change.

You cannot edit the **Name** field.

Select a value for **Tabular Fields**. This selection is required; the previous value does not persist.

Step 8 Click **Submit**.

Deleting a Context Mapping

To delete a context mapping, use the following procedure.

Before you begin

Remove references to the **Context Mapping** from all **Workflows** and reports.

Step 1 Choose **Orchestration**.
Step 2 On the **Orchestration** page, click **Script Module**.
Step 3 On the **Script Module** page, double-click the script module that contains the context mapping you want to delete.
Step 4 Choose **Context Mapping**.
Step 5 Choose the name of the context mapping you want to delete.
Step 6 Click **Delete**.
A popup appears to confirm the deletion.
Step 7 Click **Submit**.

Importing and Exporting Script Modules

You can import and export script modules using the **Import** and **Export** actions you use to import and export workflows, custom tasks, and activities.

To import a script module, see Importing Workflows, Custom Tasks, Script Modules, and Activities.

To export a script module, see Exporting Workflows, Custom Tasks, Script Modules, and Activities.