# Generic Tasks of Cisco UCS Director Base Platform Connector Pack, Release 6.7.x.x

Revised: November 12, 2019

# Generic Task for Cisco UCS Director Base Platform Connector Pack, Release 6.7.3.1

This document describes the generic tasks that are introduced in the Cisco UCS Director Base Platform Connector Pack, Release 6.7.3.1.

## Generic API Task

Cisco UCS Director has a repository of out-of-the-box tasks that are built around specific operations for a particular device. These tasks have pre-defined inputs and pre-defined outputs. In certain situations, these out-of-the-box tasks doesn't meet your requirements. Say, for example, if new parameters are added to the device specific API to enhance the existing functionality, you have to wait till the next release for the task enhancement from the Cisco UCS Director development team.

To overcome this stituation, Cisco UCS Director provides generic API tasks that you can utilize in any device specific workflow and achieve the desired result. Generic API tasks such as SSH task, which allows you to automate operations on a device which can be accessed through SSH. As SSH is a standard way of interacting with devices, you can automate such operations without depending on pre-defined tasks in Cisco UCS Director.

Generic API task is introduced as an out-of-the-box task to help you to use any API in a standard HTTP/HTTPs manner.

The pre-defined structure of an HTTP/HTTPs-based API includes the following:

- IP address
- Port
- URL path
- Headers
- Request body
- Authentication parameters

For more information on generic API tasks, see Working with Generic API Task of Cisco UCS Director.

## Using the Process Text Task for Text Manipulation

The **Process Text** is a dynamic task that enables you to dynamically define inputs and outputs. You can add the **Process Text** task to any workflow to manipulate the defined inputs as per the operations and deliver the manipulated text as an output in other tasks.

You can add the **Process Text** task to a workflow, and edit the task to define the input list, operations, and the expected output list.

To add and define parameters in the **Process Text** task, perform the following:

### Procedure

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

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

**Step 3** Locate and select the workflow to which you want to add the **Process Text** task. Alternatively, you can also add the **Process Text** task while creating a new workflow.

**Step 4** Click **Workflow Designer**.

**Step 5** Add the **Process Text** task.

a) From **Available Tasks**, navigate to **Cloupia Tasks** > **General Tasks** folder, drag and drop **Process Text** to the work area.

b) On the **Dynamic Task Definition** screen, complete the following fields:

1. **Input List**—Click **Add** and enter an input name. You can add a list of input names as task inputs that can be mapped with values on which the operation has to be performed.

2. **Operations**—Click **Add**. Enter the operation name, choose an operation type, and choose the input parameters from the defined input list. Based on the operation type that you choose, additional fields are displayed. Operations are the list of text manipulation operations to be performed to get the desired output. An operation can use any input defined in the input list or in the previous operation output.

   The supported operation types are:

| **Operation Type** | **Script syntax** | **Comments** |
|---|---|---|
| COMPARE | COMPARE(inputText1, inputText2, caseSensitiveFlag) | Compares the given two input texts and returns values accordingly:<br>• Zero—When both the input texts are same<br>• Positive value—If the inputText1 is greater than the inputText2<br>• Negative value—If the inputText1 is lesser than the inputText2<br>To enable case sensitivity for the text comparison, set the caseSensitiveFlag flag to true. |
| CONCAT | CONCAT(inputText1, inputText2,..,inputTextn) | Combines the input texts without delimiter. |
| CONTAINS | CONTAINS(inputText,searchText, caseSensitiveFlag) | Returns true if the inputText contains the SearchText<br>To enable case sensitivity for the CONTAINS operation, set the caseSensitiveFlag flag to true. |
| DECODE | DECODE(inputText) | Converts base-64 encoded inputText into plain text. |
| ENCLOSE_QUOTE | ENCLOSE_QUOTE (*inputText*, Type of quote) | Encloses the inputText within single or double quotes as per the specified quote type. |

| **Operation Type** | **Script syntax** | **Comments** |
|---|---|---|
| ENCODE | ENCODE(inputText) | Converts the plain inputText into a base-64 encoded text. |
| ENDS_WITH | ENDS_WITH(inputText,suffix, caseSensitiveFlag) | Returns true if the inputText ends with the specified suffix.<br>To enable case sensitivity for the ENDS_WITH operation, set the caseSensitiveFlag flag to true. |
| EXTRACT | EXTRACT(inputText, startIndex, endIndex) | Extracts text beginning from startIndex to endIndex-1, from the inputText. |
| INDEX | INDEX(inputText,searchText) | Returns the index position of the searchText text in the inputText. |
| INSERT | INSERT(inputText,insertText, Index) | Inserts the insertText text into the inputText at the specified Index position. |
| JOIN | JOIN(inputText1,inputText2,.., inputTextn,delimiter) | Joins two or more input texts with delimiter. |
| LENGTH | LENGTH(inputText) | Identifies the length of the inputText. |
| LOWER | LOWER(inputText) | Converts the inputText into lower case. |
| MATCH | MATCH(inputText,searchText) | Returns true if the inputText matches the text or regular expression in the searchText. |
| PAD_LEFT | PAD_LEFT(inputText,padText, lengthofText) | Appends the padText to the beginning of the inputText so that the text length is equal to the given lengthoftext. |
| PAD_RIGHT | PAD_RIGHT(inputText,padText, lengthofText) | Appends the padText to the end of the inputText so that the text length is equal to the given lengthoftext. |
| REMOVE | REMOVE(inputText, startIndex, endIndex) | Removes text beginning from startIndex to endIndex-1 from the inputText. |
| REMOVE_QUOTE | REMOVE_QUOTE(inputText) | Removes the enclosing quote from the inputText. |
| REPLACE | REPLACE(inputText,oldText, newText) | Replaces all occurrences of the oldText in the inputText with the newText. |
| REVERSE | REVERSE(inputText) | Reverses the inputText. |

| Operation Type | Script syntax | Comments |
|---|---|---|
| SPLIT | SPLIT(inputText,delimiter, noofTokens) | Splits the inputText into fragments based on the delimiter. The delimeter can be text or a regular expression. noofTokens is the token count taken in the inputText to perform a split operation. |
| STARTS_WITH | STARTS_WITH(inputText,prefix, caseSensitiveFlag) | Returns true if the inputText starts with the prefix. To enable case sensitivity for the STARTS_WITH operation, set the caseSensitiveFlag flag to true. |
| TRIM | TRIM(inputText) | Trims the leading or trailing whitespaces in the inputText. |
| UPPER | UPPER(inputText) | Converts the inputText into upper case. |

3. **Output List**—Click **Add**. Enter an output name and associate the output with one of the operation output displayed based on the defined operation names. These outputs can be mapped as inputs for other task.
4. **Check this option for script mode**—Check this option to enter the script for processing the text in the following format.

```
inputList:[
input1
input2
input3
]
operations:[
op1 = UPPER(input1)
op2 = LOWER(input2)
op3 = SPLIT(op2.output,-,3)
]
outputList:[
OUT1 = op1.output
OUT2 = op3.output1
OUT3 = op3.output2
]
```

c) On the **Task Information** screen, enter basic workflow task details and click **Next**.
d) On the **User Input Mapping** screen, map the user input to task input attributes to use values accordingly. Click **Next**.
e) On the **Task Inputs** screen, enter values for the task inputs that are not mapped to workflow inputs. Click **Next**.
f) On the **User Output Mapping** screen, map the user output to task output attributes to use values from the workflow output fields. Click **Submit**.

**Step 6** Click **Close** to close the Workflow Designer.

## Converting the Task Output Type

The **Convert Type** task is a dynamic task that enables you to dynamically define the input list and output type for the corresponding input label. You can add the **Convert Type** task to any workflow to generate output type defined for the given input value. You can map the output type to other tasks.

To add and define parameters in the **Convert Type** task, perform the following:

#### Procedure

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

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

**Step 3** Locate and select the workflow to which you want to add the **Convert Type** task. Alternatively, you can also add the **Convert Type** task while creating a new workflow.

**Step 4** Click **Workflow Designer**.

**Step 5** Add the **Convert Type** task.

a) From **Available Tasks**, navigate to **Cloupia Tasks** > **General Tasks** folder, drag and drop **Convert Type** to the work area.

b) On the **Dynamic Task Definition** screen, complete the following fields:

   1. Expand **Define Task Output Parameters**.
   2. Click **Add**.
   3. Enter an input label, and choose an output type for the input label.
   4. Click **Submit**.

      You can add input labels and its corresponding output type by clicking **Add**.

c) On the **Task Information** screen, enter basic workflow task details and click **Next**.

d) On the **User Input Mapping** screen, map the user input to task input attributes to use values accordingly. Click **Next**.

e) On the **Task Inputs** screen, enter values for the task inputs that are not mapped to workflow inputs. Click **Next**.

f) On the **User Output Mapping** screen, map the user output to task output attributes to use values from the workflow output fields. Click **Submit**.

**Step 6** Click **Close** to close the Workflow Designer.

## Registering LOV as Workflow Input

The **Register LOV** task registers given key value pairs as an LOV in the workflow task input, which can be used by other tasks after registration. During LOV registration, you can define the LOV name, labels, and values in the JSON format:

```
{"LOVName1" : {"LOVLabel":"LOVValue","LOVLabel":"LOVValue","LOVLabel":"LOVValue"},
"LOVName2": {"LOVLabel":"LOVValue","LOVLabel":"LOVValue","LOVLabel":"LOVValue"}}
```

After the task is executed, the registered LOV is displayed in the workflow task input page. You can use the registered LOV in other tasks by mapping the registered LOV as task input or admin input.

To add and define parameters in the **Register LOV** task, perform the following:

**Procedure**

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

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

**Step 3** Locate and select the workflow to which you want to add the **Register LOV** task. Alternatively, you can also add the **Register LOV** task while creating a new workflow.

**Step 4** Click **Workflow Designer**.

**Step 5** Add the **Register LOV** task.

a) From **Available Tasks**, navigate to **Cloupia Tasks** > **General Tasks** folder, drag and drop **Register LOV** to the work area.

b) On the **Task Information** screen, enter basic workflow task details and click **Next**.

c) On the **User Input Mapping** screen, map the user input to task input attributes to use values accordingly. Click **Next**.

d) On the **Task Inputs** screen, complete the following fields:

1. Check the **Enter LOV List As Text** check box to enter the LOV details in the **LOV list** field. If you want to upload the LOV details from a file, uncheck the **Enter LOV List As Text** check box and upload the file with the LOV details in the **Upload File** field.

   The LOV details include the LOV name, labels, and values in the JSON format:

   ```
   {"LOVName1" : {"LOVLabel":"LOVValue","LOVLabel":"LOVValue","LOVLabel":"LOVValue"},
   "LOVName2": {"LOVLabel":"LOVValue","LOVLabel":"LOVValue","LOVLabel":"LOVValue"}}
   ```

2. In the **LOV Type** field, click **Select** and choose the type of variables that you want to create.

3. Check the **Replace if Already Present** check box to enable overriding of LOV pairs when there is an LOV with the same name.

4. Click **Next**.

e) On the **User Output Mapping** screen, map the user output to task output attributes to use values from the workflow output fields. Click **Submit**.

**Step 6** Click **Close** to close the Workflow Designer.

## Getting Data from Tabular Reports

You can use the **Get Data From Tabular Report** task to retrieve specific data from a tabular report. The task provides options to select columns for retrieving only the selected column data from the given report name, and to define filter condition to filter specific report rows. The output is provided in the `csv` format with each row separated by a new line character.

To add and define parameters in the **Get Data From Tabular Report** task, perform the following:

**Procedure**

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

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

**Step 3** Locate and select the workflow to which you want to add the **Get Data From Tabular Report** task. Alternatively, you can also add the **Get Data From Tabular Report** task while creating a new workflow.

**Step 4** Click **Workflow Designer**.

**Step 5** Add the **Get Data From Tabular Report** task.

a) From **Available Tasks**, navigate to **Cloupia Tasks** > **General Tasks** folder, drag and drop **Get Data From Tabular Report** onto the work area.

b) On the **Task Information** screen, enter basic workflow task details and click **Next**.

c) On the **User Input Mapping** screen, map the user input to task input attributes to use values accordingly. Click **Next**.

d) On the **Task Inputs** screen, complete the following fields:

1. Click **Select** and check a report name from which you want to retrieve the data.
2. Click **Select** and check one or more report columns that you want to retrieve from the report.
3. Expand **Filter Conditions** and click **Add** to set a filter condition by choosing column label and operator, and entering a column value. You can specify multiple filter conditions for each column. The report will include data that matches either of the conditions specified for the same column or data that matches all conditions specified for every column.

e) On the **User Output Mapping** screen, map the user output to task output attributes to use values from the workflow output fields. Click **Submit**.

**Step 6** Click **Close** to close the Workflow Designer.

## Processing Time-based Data

You can use the **Process Time** task to perform actions such as converting time format, get system time and so on. To process time, you can either define the input in a specific time format or define normal time value without following any format. Then, choose an operation such as Convert Time Format, to be performed on input time, and define the output format in which the processed time has to be output for each operation.

The supported operations are: Convert Time Format, Get System Time, Get Time Difference, Get Time Component, Get Prior or After Time, and Get Time from NTP server.

For the **Get Prior or After Time** operation type, you have to input two values: Date in any format and a generic number. Based on the set prior or after action and time component, the output will be processed. For example, if you have set input as 16/07/2019 and 2, choosen **Before** and **Date** in the **Select Prior or After** and **Select Time Component** drop-down lists, then the processed time output is **14**.

For **Get Time from NTP server** operation type, you have to provide the IP address of the NTP server or DNS name.

To add and define parameters in the **Process Time** task, perform the following:

### Procedure

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

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

**Step 3** Locate and select the workflow to which you want to add the **Process Time** task. Alternatively, you can also add the **Process Time** task while creating a new workflow.

**Step 4** Click **Workflow Designer**.

**Step 5** Add the **Process Time** task.

a) From **Available Tasks**, navigate to **Cloupia Tasks** > **General Tasks** folder, drag and drop **Process Time** to the work area.

b) On the **Dynamic Task Definition** screen, complete the following fields:

1. **Input List**—Click **Add**. Enter an input label and check the **Time Format** check box to define the input in a specific time format. From the **Input Time Format** drop-down list that appears on checking the **Time Format** check box, choose a required time format or choose **Other** to set a user-defined time format in the **Other Time Format** field.

2. **Operation List**—Click **Add**. Enter the operation name, choose an operation type, and choose the input parameters from the input list. Based on the operation type that you choose, additional fields are displayed to enter additional information for executing the operation.

3. **Output List**—Click **Add**. Enter an output name and associate the output with one of the operation output displayed based on defined operation name. These outputs can be mapped as input for other task.

c) On the **Task Information** screen, enter basic workflow task details and click **Next**.

d) On the **User Input Mapping** screen, map the user input to task input attributes to use values accordingly. Click **Next**.

e) On the **Task Inputs** screen, enter values for the task inputs that are not mapped to workflow inputs. Click **Next**.

f) On the **User Output Mapping** screen, map the user output to task output attributes to use values from the workflow output fields. Click **Submit**.

**Step 6** Click **Close** to close the Workflow Designer.

## Reading the Content of a File

You can use the **Read File** task to read the content of a specifc file and generate an output with either the entire content of the file or the content in a specific line.

You can also provide a regular expression pattern to read the lines matching the given pattern.

To add and define parameters in the **Read File** task, perform the following:

### Procedure

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

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

**Step 3** Locate and select the workflow to which you want to add the **Read File** task. Alternatively, you can also add the **Read File** task while creating a new workflow.

**Step 4** Click **Workflow Designer**.

**Step 5** Add the **Read File** task.

a) From **Available Tasks**, navigate to **Cloupia Tasks** > **General Tasks** folder, drag and drop **Read File** to the work area.

b) On the **Task Information** screen, enter basic workflow task details and click **Next**.
c) On the **User Input Mapping** screen, map the user input to task input attributes to use values accordingly. Click **Next**.
d) On the **Task Inputs** screen, complete the following fields:

1. Either drag and drop or click **Select a file** to upload a file from which the content has to be read.

   **Note** If you make any changes to the file after uploading, you must rename and upload the updated file to read the latest changes.

2. In the **Line Number** field, enter a line number to retrieve the content of a specific line number as output.

3. In the **Filter Expression** field, enter a regular expression pattern to read the lines matching the given pattern.

e) On the **User Output Mapping** screen, map the user output to task output attributes to use values from the workflow output fields. Click **Submit**.

**Step 6** Click **Close** to close the Workflow Designer.

## Related Documentation

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

- For a better understanding and usage of Cisco UCS Director Orchestrator, see the Cisco UCS Director Orchestration Guide.
- For automating operations on a device using generic API tasks, see the Working with Generic API Task of Cisco UCS Director.
- For a description of scripting technologies available in Cisco UCS Director and help choosing the right solution for your application, see the Cisco UCS Director API Customization and Integration Guide.
- For an introduction to developing custom tasks, see the Cisco UCS Director Custom Task Getting Started Guide.
- For examples of scripts that can be used to customize tasks, see the Cisco UCS Director CloupiaScript Cookbook.