Creating Workflows and Wizards Using the Workflow Editor and Activation Features

Using the Workflow GUI Client to Create Workflows

The Workflow and Activation features are only available in upgrades to Prime Network 4.0. Because Workflow and Activation will be deprecated in the future, you should use Transaction Manager instead.

The following topics describe how to use the Prime Network Workflow GUI client to create workflow templates that allow you to deploy and execute a series of commands in runtime.

- What is Prime Network Workflow?, page D-2
- Prime Network Workflow Window, page D-2
- Steps for Creating a Workflow Template, page D-5
- Creating and Testing a New Workflow Template, page D-5
- Using Tasks in the Task Library, page D-8
- Prime Network Activation Workflow Template Attributes, page D-13
- Running, Viewing, Aborting, and Deleting Workflows, page D-14
- Viewing Workflow History Using Prime Network Events, page D-18
- Creating Custom Tasks for Workflows, page D-19

Additional information about Prime Network Workflow is available on the Cisco Developer Network (CDN), including sample workflow that will help you become familiar with the framework.
What is Prime Network Workflow?

Prime Network Workflow is used for creating activation workflow templates. A workflow template consists of several tasks which are grouped together and specified into a flow, with certain sequences, branches, and failure policies (including rollback procedures). You can test and preview workflow templates before deploying (copying) them to the gateway to make them available to Prime Network users.

After a template is deployed to the gateway, users with OperatorPlus privileges can execute the workflow using a BQL command. Each time a user executes a workflow, Prime Network creates a new workflow instance with its own unique ID, and stores it in the database. Thus there can be many instantiations of a single workflow template. Each instance has its own output, which you can export and use to analyze the commands.

Workflows can be nested, which means one workflow can invoke subworkflows synchronously, and pass arguments to each subworkflow invocation. This allows you to group and isolate tasks that are in a child workflow (sub-workflow) from tasks that are in the parent workflow.

Workflows can include commands that you created using Command Builder as long as the commands have been published (made accessible to) the NEs that will be affected by the workflow. Those commands are called by BQL tasks in the workflow. Like Command Builder, Workflow can access all information models (IMOs).

You can use Command Builder, but not Command Manager, with the Workflow application.

The Workflow GUI is based on LiquidBPM by Autonomy, Inc., with extensions that are specific to Prime Network. Because in many cases the Prime Network procedure is the same as the LiquidBPM procedure, the topics in this section will direct you to the LiquidBPM online help, which can be launched from the Workflow GUI client.

The Workflow Engine resides on the gateway using AVM 66. A workflow that is stored on the gateway (rather than locally on a client machine) is called a deployed workflow. Deployed workflows can be managed from the Prime Network Administration GUI client, where users with Administrator privileges can view the workflow’s status and properties. To deploy, retrieve, or delete a workflow template from the gateway, AVM 66 must be running.

Prime Network Workflow Window

If Prime Network is installed as a standalone product, launch Workflow as follows:

Start > All Programs > Cisco Prime Network > Cisco Prime Network Workflow

If Prime Network is installed in suite mode with Cisco Prime Central, launch Workflow by selecting Tools > Workflow in the Prime Network Vision GUI client.
Figure D-1 shows an example of the Prime Network Workflow window when a template is opened.

**Figure D-1  Workflow Window**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drawing area</td>
</tr>
<tr>
<td>2</td>
<td>Task toolbar (see Table D-2)</td>
</tr>
<tr>
<td>3</td>
<td>Task attribute table</td>
</tr>
<tr>
<td>4</td>
<td>Tree</td>
</tr>
<tr>
<td>5</td>
<td>Action toolbar (see Table D-1)</td>
</tr>
</tbody>
</table>

Table D-1 and Table D-2 list the action and tasks with which you should become familiar.

**Table D-1  Important Workflow Template Actions**

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>New Workflow</td>
<td>Creates a new workflow template and automatically saves it on the local client machine.</td>
</tr>
<tr>
<td><strong>Tip</strong></td>
<td>If you want to create a wizard using the Activation Wizard Builder (AWB), choose <strong>File &gt; New Activation Workflow</strong> to create the activation. That template contains information that is necessary for creating a wizard using the AWB.</td>
</tr>
<tr>
<td>The Open Workflow,</td>
<td>Save Workflow As, Rename Workflow, and Delete choices all perform operations on this same directory.</td>
</tr>
<tr>
<td>Import Workflow</td>
<td>Loads a saved template XML file into the Workflow Editor.</td>
</tr>
</tbody>
</table>
Table D-1  Important Workflow Template Actions (continued)

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![icon]</td>
<td>Export saves a workflow template as an XML file on the local machine. This is useful if you want to perform file actions, such copying or e-mailing it to another location. (The XML template is not saved in the Prime Network database.)</td>
</tr>
<tr>
<td>![icon]</td>
<td>Execute runs the workflow template on the local machine. If the workflow template contains valid BQL tasks, they will be executed on the gateway.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Retrieve Workflow from Server retrieves a workflow template from the Prime Network database and loads it into the Workflow Editor.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Delete Workflow from Server deletes a workflow template from the Prime Network database.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Deploy Workflow saves the workflow template on the gateway and in the Prime Network database.</td>
</tr>
</tbody>
</table>

Table D-2  Important Workflow Template Tasks

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![icon]</td>
<td>Execute BQL calls commands that reside on the client machine. For more information, see Execute BQL Task, page D-8.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Subflow another workflow in an embedded fashion. See Workflow Call Task, page D-10.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Lock protects a resource so that it cannot be used concurrently by multiple workflows. See Lock and Unlock Task, page D-12.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Unlock unlocks a locked resource so that it can be locked and unlocked using the Lock task and the Unlock task. See Lock and Unlock Task, page D-12.</td>
</tr>
<tr>
<td>![icon]</td>
<td>Activation Tokenizer provides repetitive looping so that multiple attributes can be passed to the workflow. See Activation Tokenizer Task, page D-13.</td>
</tr>
</tbody>
</table>
Steps for Creating a Workflow Template

**Note**
The Workflow and Activation features are only available in upgrades to Prime Network 4.0. Because Workflow and Activation will be deprecated in the future, you should use Transaction Manager instead.

These steps describe how to create a workflow template.

**Figure D-2 **Steps for Creating a Workflow

1. **Step 1:** Create a command using Command Builder and preview it
2. **Step 2:** Define tasks and workflows using Prime Network Workflow
3. **Step 3:** Test the workflow locally
4. **Step 4:** Deploy the workflow to the gateway
5. **Step 5:** Run the workflow
6. **Step 6:** View the workflow results in Prime Network Administration

**Note**
Commands you create with Command Builder must be published to all NEs that are part of the workflow. If a workflow calls a command that has not been published to the NE, it will fail on the NE. (You cannot use commands created using Command Manager.)

For more information about:

- Creating a command using the Command Manager GUI client, see Creating Commands and Command Scripts to Perform Device Configuration Operations, page 4-1).
- Creating a workflow template and testing it locally, see Creating and Testing a New Workflow Template, page D-5.
- Running the workflow, see Running, Viewing, Aborting, and Deleting Workflows, page D-14.

Creating and Testing a New Workflow Template

You can create workflow templates from scratch, or retrieve an existing workflow template, rename it, and edit it. When you create a new workflow template, the template information is saved on your local client machine until you deploy (save) it to the gateway. When you deploy the workflow template to the
gateway, it can be viewed and managed from the Prime Network Administration GUI client. To retrieve, deploy, or delete an existing workflow templates from the gateway, AVM 66 must be running so that Workflow and the gateway can properly communicate.

Tip
Use Export to save templates. This saves the template as an XML file.

Use this procedure to create a new workflow template.

Step 1
Launch Prime Network Workflow.
If Prime Network is installed as a standalone product, launch Workflow as follows:

Start > All Programs > Cisco Prime Network > Cisco Prime Network Workflow

If Prime Network is installed in suite mode with Cisco Prime Central, launch Workflow by selecting Tools > Workflow in the Prime Network Vision GUI client.

Step 2
Create the new workflow template by choosing one of the following:

- If you want to create a non-activation workflow, choose File > New Workflow.
- If you want to create an activation workflow, choose File > New Activation Workflow.
- To open an existing template, choose File > Open Workflow, which will list all of the templates that are stored in C:/Cisco Systems/Prime Network/studio_db.

Step 3
Name the template in accordance with the following rules, and click OK.

- Do not include underscore (_ ) or percent (%) wildcards.
- If you are creating an activation workflow, give the template an NSA_ prefix (otherwise it will not appear in the Activation History window).

Tip
Use a suffix to differentiate workflow templates created in different Prime Network versions, such as _v_3_8 and _v_3_9. This will help when you upgrade the Prime Network software.

The template is saved on the local client machine under C:/Cisco Systems/Prime Network/studio_db.

Step 4
Add Tasks to the workflow template. The following steps describe the basic process:

a. Click a Task in the task toolbar, and click your mouse in the drawing area. The task is added to the drawing area.

Several tasks are especially useful for Prime Network and are described in Using Tasks in the Task Library, page D-8. For information about the other tasks that are available, see the LiquidBPM online help (select File >Help and select Task Library).

b. Assign properties to the task by right-clicking the task icon and selecting Edit Properties. This opens a properties sheet.

c. Define the properties. Properties are divided into three types:

- General Properties, which are property types that are common to all tasks (such as Name). Some properties are assigned by Prime Network (for example, a task ID number); these are called fixed properties. The other properties are defined by the user and are called editable properties.
- Attributes, which are task-specific properties.
Appendix D  Creating Workflows and Wizards Using the Workflow Editor and Activation Features

Using the Workflow GUI Client to Create Workflows

---

**Tip** Incorrect attribute names are a common cause of workflow template errors, so be careful when entering this information.

---

- Callback Scripts, which specify actions that should be performed (or criteria that should be met) before or after performing the task, or if an exception is encountered.

---

**Tip** If you want Prime Network to generate an event when a workflow is aborted, configure a preActivateScript in the Callback Scripts tab:

```java
thisWorkflow.getRootTask().abort();
```

---

**Note** By default, all attributes in Prime Network Workflow are of the type String. To use another attribute type, such as Integer, set the flag use-workflow-string-param-casting to true in `NETWORKHOME/Main/registry/workflowavm.xml`. In `workflowavm.xml`, the flag `use-workflow-string-param-casting` appears under the `workflow` key. After you change this flag to true, you can use attribute types other than String.

---

**Step 5** Test the template locally.

a. Choose **Tools > Execute Workflow**. Different tabbed views are opened. The graphical, tabular, and XML views provide different ways to examine the workflow progress, and the Workflow State Analysis tab that reflects the activation status of the workflow.

b. Set the test attribute values:

   - Right-click anywhere in the grey area and select **Workflow Properties**.
   - In the Attributes tab, click the Default Values area and enter some values.
   - Click **Apply** and **Close**.

c. Click **Activate**. A Console window opens and prints the status of each task in the workflow template.

d. Close the Console window. Note that the shape and colors of the tasks in the Graphical View tab changes to show the activation status of each task: Ready (blue), Active (green), Abort (red), Done (grey), and Passive (light grey).

---

**Step 6** Once you have tested the workflow template successfully on the local client, you can do any of the following:

a. Save the workflow template on the gateway and in the Prime Network database by choosing **Tools > Deploy Workflow**.

b. Use a BQL command to run the workflow template on the gateway. See **Previewing and Running Workflow Templates**, page D-15.

c. (Activation workflow templates only) Create a wizard that will call the workflow template. See **Steps for Creating an Activation Wizard**, page D-24.
Using Tasks in the Task Library

The following tasks are of special importance in Prime Network. See the accompanying topics for more information:

<table>
<thead>
<tr>
<th>Task Type</th>
<th>For more information, see:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Execute BQL</td>
<td>Execute BQL Task, page D-8</td>
</tr>
<tr>
<td>Workflow Call</td>
<td>Workflow Call Task, page D-10</td>
</tr>
<tr>
<td>Subflow</td>
<td>Subflow Task, page D-12</td>
</tr>
<tr>
<td>Lock and Unlock</td>
<td>Lock and Unlock Task, page D-12</td>
</tr>
<tr>
<td>Activation Tokenizer</td>
<td>Activation Tokenizer Task, page D-13</td>
</tr>
</tbody>
</table>

BQL commands are specified in the Command Template tab of the Task Properties dialog box, and the result is stored in a task attribute called Result so that it can be used by scripts and other tasks. The Command Template tab can reference workflow template attributes and task attributes. At runtime, the attribute’s values are substituted into the template before it is run.

**Note**

The BQL task in a workflow template can only execute script BQLs and any of the generic commands: Get, Delete, Find, Update, or Set.

**Note**

When you execute a workflow to run commands on a device, using the RunWorkflow BQL command, it will be executed only if the preview flag in the RunWorkflow BQL is false. If the value of the preview flag is true, it will only be previewed. See Previewing and Running Workflow Templates, page D-15 for details.

**Execute BQL Task**

The BQL task in a workflow template can only execute script BQLs and any of the generic commands: Get, Delete, Find, Update, or Set. The Execute BQL task has an additional tab in its properties sheet, Command Template, where you enter the command definition.

In the following example, a user used an activation template, and created a BQL task and named it Create VRF. The BQL task calls AddVrfSite, which is a command the user built with Command Builder. (You cannot use commands created with Command Manager.) All of the attributes in the AddVrfSite command are entered in the Attributes tab.
Note that the attributes listed in Figure D-3 are used in the AddVrfSite command illustrated in Figure D-4.

Each BQL task includes a task attribute called RollbackEnabled which controls whether or not the BQL task command is rolled back if the workflow stops. The default value is True.

Tip
Incorrect attribute names (attribute evaluation failures) are a common cause of workflow template errors, so a best practice is to have the text of the Command Builder command on hand so you can cut and paste the attribute names into the Attributes tab. (You cannot use commands created with Command Manager.)
Figure D-4 Command Template for a Create BQL Task

![Command Template](image)

Use the following format for entering information into the Command Template tab. This is similar to a command instance descriptor (CID), where you can designate the options that should be evaluated on execution.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$Attribute_name$</code></td>
<td>Designation for an attribute in a workflow template.</td>
</tr>
<tr>
<td><code>$Task_name:Attribute_name$</code></td>
<td>Designation for an attribute in a task.</td>
</tr>
</tbody>
</table>

**Workflow Call Task**

The Workflow Call task starts another independent workflow, called a child workflow. The parent workflow passes parameters to each subworkflow invocation. Using subflows lets you isolate the tasks. The Workflow Call task has two additional tabs in its properties sheet:

- Assign Attributes, where you specify the attributes to be used by the child workflows.
- Target Template, where you enter the child workflows that will be called by the parent workflow.

**Note**

Child workflows are not visible through the API. You must interact directly with the parent workflow.

The parent workflow only runs when the subworkflow has finished.

A subflow does the following:

- The parent workflow passes parameters to the child workflow. If the parent workflow is aborted or deleted, the child workflow is also aborted or deleted.
- The child workflow is runs separately, and has its own scope for attributes. Its output is directed to the parent workflow.
The parent workflow waits for the child workflow to terminate, and only runs when the child workflow has finished. If the child workflow stops, the parent workflow also stops.

The maximum nesting depth is 16, and rollback is performed at all depths.

In the following example, a user created a workflow call task named **Workflow Call**. Figure D-5 shows the Assign Attributes tab, which instructs the task to use the parent attributes as input to this task.

**Figure D-5 Assign Attributes Tab for a Workflow Call Task**

Use the following format for entering information into the Assign Attributes tab.

<table>
<thead>
<tr>
<th>Format</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribute_in_child_workflow=attribute_in_parent_workflow</td>
<td>Designation for attributes taken from a parent workflow.</td>
</tr>
<tr>
<td>attribute_in_child_workflow=task_in_parent_workflow:attribute_name</td>
<td>Designation for attributes taken from a particular task in the parent workflow.</td>
</tr>
</tbody>
</table>

The child workflows that will be called by this task are displayed in Figure D-6.
Using the Workflow GUI Client to Create Workflows

Subflow Task

The Subflow task is used to embed and run another workflow in a workflow instance. It is similar to the Workflow Call task except that in the Subflow task, the workflows are actually embedded in each other. The maximum workflow nesting depth is defined in the registry. The default value is 16.

To configure this type of task, see the instructions in Workflow Call Task, page D-10.

Lock and Unlock Task

The Lock and Unlock tasks allow you to protect a component from concurrent use by multiple workflows. Both tasks have an additional tab in its properties sheet, Resource Names, where you list the resources to be locked (and unlocked). A lock can be owned by only one workflow at a time.

Activities not related to workflows can access a resource even when it is locked by a workflow. Participation in the locking process is optional.

Resources are automatically locked during rollback, but the system prevents deadlocks before they occur. Upon detecting an imminent deadlock, the lock operation belonging to the workflow with the least progress fails. A failed lock might or might not abort the workflow.

Figure D-7 gives an example of a lock task.
Using the Workflow GUI Client to Create Workflows

In the text area, enter the names of the resources to be locked. Each resource name should be on a separate line.

You can also pass resource names as parameters to other workflows. This is similar to a command instance descriptor (CID), where you can designate the options that should be evaluated on execution.

**Activation Tokenizer Task**

The Activation Tokenizer task supports repetitive looping, where multiple values for an attribute are passed to the workflow. The wizard activation packs multiple values into an attribute using an “|” delimiter between values, and places them in a temporary internal attribute that can be used within the looping flow. For example, if you had four devices, you could designate them as follows:

```
MyDevices=dev1|dev2|dev3|dev4
```

The loop code would do four loops and with each loop, get a new attribute value for a device.

**Prime Network Activation Workflow Template Attributes**

All Prime Network activation workflows contain the following attributes. The AA prefix identifies them as Activation workflow attributes. They are all of type String.

| Attribute Name           | Description                                                                 | Values: Free form | Default: “|” |
|--------------------------|-----------------------------------------------------------------------------|-------------------|------------|
| AA_NSA_ActivationInfo    | Text that is displayed in the Info column of the Activation History window. It is updated in the workflow to reflect a user-friendly message about the status of the workflow execution (for example, “See Workflow output for more details”). |                   |            |
| AA_NSA_capabilities      | Internal value set by the workflow designer to reflect the capabilities of the workflow. | add or remove plus set of values delimited by |             |
|                          |                                                                            | Default: { add | remove }                     |            |

---

*Figure D-7 Resource Names Tab for Lock Task*

In the text area, enter the names of the resources to be locked. Each resource name should be on a separate line.

You can also pass resource names as parameters to other workflows. This is similar to a command instance descriptor (CID), where you can designate the options that should be evaluated on execution.

**Activation Tokenizer Task**

The Activation Tokenizer task supports repetitive looping, where multiple values for an attribute are passed to the workflow. The wizard activation packs multiple values into an attribute using an “|” delimiter between values, and places them in a temporary internal attribute that can be used within the looping flow. For example, if you had four devices, you could designate them as follows:

```
MyDevices=dev1|dev2|dev3|dev4
```

The loop code would do four loops and with each loop, get a new attribute value for a device.

**Prime Network Activation Workflow Template Attributes**

All Prime Network activation workflows contain the following attributes. The AA prefix identifies them as Activation workflow attributes. They are all of type String.

| Attribute Name           | Description                                                                 | Values: Free form | Default: “|” |
|--------------------------|-----------------------------------------------------------------------------|-------------------|------------|
| AA_NSA_ActivationInfo    | Text that is displayed in the Info column of the Activation History window. It is updated in the workflow to reflect a user-friendly message about the status of the workflow execution (for example, “See Workflow output for more details”). |                   |            |
| AA_NSA_capabilities      | Internal value set by the workflow designer to reflect the capabilities of the workflow. | add or remove plus set of values delimited by |             |
|                          |                                                                            | Default: { add | remove }                     |            |
Using the Workflow GUI Client to Create Workflows

Running, Viewing, Aborting, and Deleting Workflows

These topics explain how to perform workflow operations using saved templates:

- Previewing and Running Workflow Templates, page D-15
- View Workflow Status, page D-15
- View Workflow Output, page D-16
- Abort Workflow, page D-16
- Delete Workflow, page D-17

### Table D-3 Workflow Attributes Supplied in Activation Workflows

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Description</th>
<th>Values: (see Description)</th>
<th>Default: &quot;&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA_NSA_coWFId</td>
<td>Maintains an association between related workflows.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>A workflow containing the initial <strong>add</strong> operation should have a &quot;&quot; value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>If a <strong>remove</strong> is later performed, its AA_NSA_coWFId value</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>should be the same as the <strong>add</strong> workflow AA_NSA_coWFId value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA_NSA_CustomerName</td>
<td>Stores the customer name (requires per-service activation input).</td>
<td>Value: Free form</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>Default: &quot;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA_NSA_CreateTime</td>
<td>Stores the workflow creation time (in long format, stored as a string for BQL). Does not require user input because activations and deactivations manage this information. The time is in long format and stored as a string for BQL.</td>
<td>Value: Free form</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>Default: &quot;&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA_NSA_operation</td>
<td>Set by Activation component to <strong>add</strong> for an activation and <strong>remove</strong> for a deactivation.</td>
<td>Values: <strong>add</strong> or <strong>remove</strong></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>Default: <strong>add</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA_NSA_purge</td>
<td>Controls the purging behavior for a workflow instance record in the tablespace.</td>
<td>Values: <strong>none</strong>, <strong>false</strong>, <strong>pending</strong>, or <strong>true</strong></td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>Default: <strong>none</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA_NSA_serviceActivationId</td>
<td>Set by Activation component to the value of the IMetaDataListEntry ID. It is only stored during the workflow execution. This attribute is used by Activation History and Clone Activation feature to retrieve a more user-friendly display name (from Service.xml metadata file).</td>
<td>Default: <strong>none</strong></td>
<td></td>
</tr>
<tr>
<td>AA_NSA_type</td>
<td>Set by Activation component to <strong>NSA</strong> to distinguish these workflows from non-Prime Network workflows. Is used internally and cannot be modified.</td>
<td>Default: <strong>NSA</strong></td>
<td></td>
</tr>
<tr>
<td>AA_NSA_UserName</td>
<td>Stores the username at the time of activation or deactivation; no user input required.</td>
<td>Values: Free form</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td></td>
<td>Default: &quot;&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Previewing and Running Workflow Templates

Note
When you run a workflow, you may be shown a warning message. This is a global controls that is set from the Administration GUI client. For more information, see the Cisco Prime Network 4.0 Administrator Guide.

Use the following BQL command to preview or run a workflow that has been deployed to the gateway. This command returns the ID of the new workflow. (There is no option to run workflows from the Prime Network Administration GUI client.)

In this example, the preview value is set to false, and the task is fully executed. This is the default.

```
<?xml version="1.0" encoding="UTF-8"?>
<command name="RunWorkflow">
  <param name="templateOid">
    <value>{[WorkflowTemplate(Name=test.template)]}</value>
  </param>
  <param name="workflowAttributes">
    <value>
      <IWorkflowStringAttribute>
        <ID type="Oid">{[WorkflowAttribute(Name=param1)]}</ID>
        <Value type="String">name</Value>
      </IWorkflowStringAttribute>
      <IWorkflowIntegerAttribute>
        <ID type="Oid">{[WorkflowAttribute(Name=param2)]}</ID>
        <Value type="Integer">3</Value>
      </IWorkflowIntegerAttribute>
    </value>
  </param>
  <param name="preview">
    <value>false</value>
  </param>
</command>
```

If the application settings (global settings in the Administration GUI) require input of each user's device access credentials, you need to add the following parameters to the command. If these parameters are not added, the credentials specified during VNE creation will be used.

```
<param name="DEVICE_USER_NAME"><value>user-name</value></param>
<param name="DEVICE_PASSWORD"><value>user-password</value></param>
```

This example sets the preview value to true.

```
<value>true</value>
</command>
```

View Workflow Status

Use the following BQL command to get the status of an executed workflow.

Users with Administrator privileges can view the status of a workflow can be checked from the Administration GUI client.
<command name="Get">
  <param name="oid">
    <value>({Workflow(Id=114)})</value>
  </param>
  <param name="rs">
    <value>
      <entry name="depth">100</entry>
      <entry name="register">false</entry>
      <entry name="cachedResultAcceptable">false</entry>
      <key name="requiredProperties">
        <key name="*">
          <entry name="*"/>
        </key>
      </key>
    </value>
  </param>
</command>

View Workflow Output

The following BQL command displays the output of a workflow. You can use this command when you are testing a workflow and after you execute it. Administrators can also view workflow output from the Administration GUI client by right-clicking a workflow and choosing Show Output.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<command name="GetWorkflowOutput">
  <param name="oid">
    <value>({Workflow(Id=workflow-id)})</value>
  </param>
</command>
```

This command returns the output as a string; for example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<IMO>
  <Output type="String">
    ...
    [progress messages and script testing results]
    ...
  </Output>
</IMO>
```

Abort Workflow

Workflows can be manually aborted using the BQL command below. Users with Administrator privileges can also abort running workflows from the Administration GUI client by right-clicking a running workflow and choosing Abort.

A workflow will automatically abort if any of its tasks are aborted. When a workflow aborts, a workflow rollback occurs. This means all Execute BQL tasks that have already run will roll back. The rollback is performed according to actions specified in the command when it was created using Command Builder. Scripts roll back in the reverse order of their execution. (You cannot use commands created with Command Manager.)

Rollback can be disabled for specific BQL tasks by setting the RollbackEnabled value task attribute to false in the respective BQL task. This is useful for a BQL task executing a script that does not have an appropriate rollback, or a BQL task executing a gateway command.
If you want Prime Network Events to display an event when a Workflow is aborted, add the following line to the preActivateScript in the Callback Scripts tab:

```javascript
thisWorkflow.getRootTask().abort();
```

For more information, see Prime Network Workflow application online help for **Engine Behavior > Workflow State Transitions**.

**Note**

Gateway commands do not support rollback. Rollback scripts must be specified in the workflow templates.

The following BQL command stops a workflow that has been executed from the gateway. This includes stopping a running workflow, and rolling back all activation scripts that have been executed. Messages are returned only if the command fails.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<command name="AbortWorkflow">
    <param name="oid">
        <value>{[Workflow(Id=workflow-id)]}</value>
    </param>
</command>
```

**Delete Workflow**

This BQL command deletes an inactive workflow (that is, one that has already been completed or aborted) from the Prime Network database. Users with Administrator privileges can also delete workflows from the Administration GUI client by right-clicking a running workflow and choosing **Delete**.

**Note**

Delete operations performed from the Workflow GUI client will delete templates, not workflow instances.

Messages are only returned if the command fails.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<command name="Delete">
    <param name="oid">
        <value>{[Workflow(Id=workflow-id)]}</value>
    </param>
</command>
```

If successful, this command returns nothing.
Viewing Workflow History Using Prime Network Events

You can view the history of workflows by opening Prime Network Events and clicking the Provisioning Tab, as shown in Figure D-8.

Figure D-8 Prime Network Events - Provisioning Tab

You can also view the properties of the event, as shown in Figure D-9.
The Description area of the Provisioning Event Properties window displays the details of the execution of the workflow, including all scripts, script rollbacks, and log messages.

Creating Custom Tasks for Workflows

These topics explain how to extend the Workflow Engine with custom tasks and Prime Network Workflow callbacks:

- Writing the Code for New Custom Tasks, page D-20
- Packaging the New Tasks for Deployment to Devices, page D-21
- Deploying the Custom Tasks to Devices, page D-22
Appendix D  Creating Workflows and Wizards Using the Workflow Editor and Activation Features

Writing the Code for New Custom Tasks

Custom Tasks

The procedure for developing custom tasks is explained in the LiquidBPM documentation. The following is the definition of a simple task class:

```java
package samples;
import com.dralasoft.workflow.Key;
import com.dralasoft.workflow.SynchronousTask;

public class CustomTask1 extends SynchronousTask {
    public CustomTask1(Key _key) {
        super(_key);
    }

    public void perform() {
        System.err.println("Hello from CustomTask1");
    }
}
```

Custom Task Panel Factories

A TaskPanelFactory implementation class can be assigned to each custom task class. This allows you to create custom property sheets for the custom class. This factory class should implement the interface `com.dralasoft.gui.common.ext.TaskPanelFactory`. A simple way to do this is to extend `com.dralasoft.gui.common.ext.DefaultPanelFactory` and override some of its methods.

Workflow Callbacks Class

It is possible to develop a class that implements the Prime Network Workflow callbacks. This class should implement the interface `com.sheer.client.workflowstudio.IWorkflowEditorCallbacks`, which includes the following methods:

```java
/**
 * Called before the deploy template action
 * @param templateName
 * @return true if deploy template action should proceed
 */
public boolean preDeployTemplate(String templateName);

/**
 * Called after the deploy template action
 * @param templateName
 */
public void postDeployTemplate(String templateName);

/**
 * Called before the delete template action
 * @param templateName
 * @return true if delete template action should proceed
 */
public boolean preDeleteTemplate(String templateName);

/**
 * Called after the delete template action
 * @param templateName
 */
public void postDeleteTemplate(String templateName);
```
/**
 * @param templateName
 * @return true if this template should be displayed, false if not
 */
public boolean shouldDisplayTemplate(String templateName);

Packaging the New Tasks for Deployment to Devices

Classes and resources must be packaged into JAR files to be deployed on the server. A JAR file can contain multiple tasks and an optional workflow callback implementation.

In addition to the class files, each JAR file must contain a descriptor file named extension-config.xml. This file contains the XML block or blocks that define the tasks’ appearance in the task palette, the tasks’ custom panel factories, and, optionally, the Prime Network Workflow callback implementation.

Custom Tasks

In addition to the custom tasks class files, task icons should be included in the JAR file. Task icons should measure 16 by 16 pixels and must be placed in the JAR file in the subdirectory, com/dralasoft/gui/common/images/16x16.

Custom Task Panel Factories

The full class name of the task panel factory should be added in an element called task-panel-factory-class inside the custom-task element. If this element is not specified, com.dralasoft.gui.common.ext.DefaultPanelFactory is used for the custom task.

Workflow Callbacks Class

Adding an editor-callbacks-class element to the extension-config element configures a class that implements the Prime Network Workflow callbacks. This element should be added to the extension-config.xml file in only one of the JAR files (if the element is present in more then one descriptor, one element’s value would be used arbitrarily).

The following example of the contents of a task descriptor file defines two task types, one of which has a task panel factory, and an editor callback implementation:

```xml
<extension-config>
  <editor-callbacks-class>
    samples.EditorCallbacksImpl
  </editor-callbacks-class>

  <custom-task>
    <class-name>samples.CustomTask1</class-name>
    <label>Custom Task 1</label>
    <icon>task1.png</icon>
    <tooltip>Custom Task 1</tooltip>
    <menu-display>true</menu-display>
    <toolbar-display>true</toolbar-display>
    <task-panel-factory-class>
      com.sheer.client.workflowstudio.TestTaskPanelFactory
    </task-panel-factory-class>
  </custom-task>

  <custom-task>
    <class-name>samples.CustomTask2</class-name>
    <label>Custom Task 2</label>
    <icon>task2.png</icon>
```
Deploying the Custom Tasks to Devices

To deploy JAR files:

---

**Step 1** Confirm that no workflows are currently running because this procedure will restart the Workflow engine.

**Step 2** On the Prime Network gateway, create a directory called \( \text{NETWORKHOME/dralasoft_extensions} \).

**Step 3** Copy the JAR files to the \( \text{NETWORKHOME/dralasoft_extensions} \) directory.

**Step 4** Run the script \( \text{NETWORKHOME/Main/scripts/installDralasoftExtensions.pl} \).

---

**Note** Run this script each time you want to add a JAR file, remove a JAR file, or replace an existing JAR file with a new version.

The script uninstalls and reinstalls JAR files so that the set of installed JAR files is the same as the contents of the directory \( \text{NETWORKHOME/dralasoft_extensions} \). We therefore recommend that you keep files with a .jar extension here even after they have been deployed.

**Step 5** When the script is done, launch Prime Network Workflow. The deployed JAR files are automatically downloaded, and the new tasks are included in the task toolbar.

---

Using the Activation Wizard Builder

**Note** The Activation and Workflow features are only available in upgrades to Prime Network 4.0. Because Workflow and Activation will be deprecated in the future, you should use Transaction Manager instead.

These topics explain how users with Configurator or Administrator privileges can use the Activation Wizard Builder (AWB) to create an activation wizard, which can be accessed by network operators from Prime Network Vision, and used to run activations on managed devices.

Activation wizards provide an alternative to deploying workflows via BQL.

- **What is the Activation Wizard Builder (AWB)?**, page D-23
- **AWB Window**, page D-23
- **Steps for Creating an Activation Wizard**, page D-24
- **Creating a New Activation Wizard**, page D-25
- **Adding an Activation Wizard to Prime Network Vision**, page D-29
- **Synchronizing the Wizard and a Workflow When a Workflow is Modified**, page D-32
What is the Activation Wizard Builder (AWB)?

The AWB allows you to create a GUI interface for deploying activations, in the form of a wizard. When you create an activation wizard, you select a workflow template that defines the activation to be deployed to the managed devices. You then create the wizard screens and designate which workflow attributes should appear on each screen. You can configure attribute display names, wizard loops, tool tips, and conditions that determine the wizard content.

When you are done, the AWB validates the wizard definition. Then you can upload the wizard to the gateway and view it in Prime Network Vision. You can also change the configuration of existing wizards by downloading them from the gateway into AWB.

AWB Window

If Prime Network is installed as a standalone product, launch the AWB as follows:

b. Click Cisco Prime Network Activation Wizard Builder.

If Prime Network is installed in suite mode with Cisco Prime Central, launch Workflow by selecting Tools > Activation Wizard in the Prime Network Vision GUI client.

Note

The Activation and Workflow features are only available in upgrades to Prime Network 4.0. Because Workflow and Activation will be deprecated in the future, you should use Transaction Manager instead.

Figure D-10 shows the AWB window. In this example, the window is displaying a wizard that has two pages and one unassigned attribute.
Appendix D  
Creating Workflows and Wizards Using the Workflow Editor and Activation Features

Using the Activation Wizard Builder

Table D-4  Important Activation Actions

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Open]</td>
<td>Opens a wizard’s file set in AWB.</td>
</tr>
<tr>
<td>![Import]</td>
<td>Imports a wizard (pages and activation XML files) from a local machine into AWB.</td>
</tr>
<tr>
<td>![Export]</td>
<td>Exports a wizard (pages and activation XML files) from the AWB to your local machine.</td>
</tr>
<tr>
<td>![Validate]</td>
<td>Validates the current wizard’s definition.</td>
</tr>
<tr>
<td>![Preview]</td>
<td>Previews the XML of the current wizard.</td>
</tr>
<tr>
<td>![Download]</td>
<td>Downloads a wizard’s files from the gateway.</td>
</tr>
<tr>
<td>![Upload]</td>
<td>Uploads the current wizard’s files to the gateway and adds it to Prime Network Vision’s Activation menu.</td>
</tr>
<tr>
<td>![Manage]</td>
<td>Manages the wizard properties and directory.</td>
</tr>
<tr>
<td>![Synchronize]</td>
<td>Synchronizes a wizard with a workflow that has been updated.</td>
</tr>
</tbody>
</table>

**Steps for Creating an Activation Wizard**

**Note**

The Activation and Workflow features are only available in upgrades to Prime Network 4.0. Because Workflow and Activation will be deprecated in the future, you should use Transaction Manager instead.

These steps describe how to create an activation wizard.
Appendix D  Creating Workflows and Wizards Using the Workflow Editor and Activation Features

Using the Activation Wizard Builder

<table>
<thead>
<tr>
<th>Figure D-11  Steps for Creating an Activation Wizard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a new wizard using an existing workflow (from your local machine)</td>
</tr>
<tr>
<td>Create wizard pages, and select attributes and rules for each page</td>
</tr>
<tr>
<td>Add page iterations (loops), if desired</td>
</tr>
<tr>
<td>Add groups of attributes that will appear, based on other attribute values (groups), if desired</td>
</tr>
<tr>
<td>Validate the wizard and upload it from your local machine to the gateway</td>
</tr>
<tr>
<td>Test the wizard from Prime Network Vision</td>
</tr>
</tbody>
</table>

For more information about:
- Creating a new wizard and validating it, see Creating a New Activation Wizard, page D-25.
- Adding the new wizard to the GUI client, see Adding an Activation Wizard to Prime Network Vision, page D-29.

**Creating a New Activation Wizard**

Use the following procedure to create new Prime Network activation wizards. This procedure requires at least one workflow that is stored on the gateway.

**Tip**

A best practice is to create a wizard design plan based on the workflow attributes. It will make the wizard creation process much simpler.

**Note**

The Cisco Developer Network (CDN) has some scripts that you can use as examples for using the framework. Other activation scripts are only available through Cisco Advanced Services.

**Step 1**
Verify that a workflow is stored on the gateway. (The workflow must be based on an Activation workflow template—that is, the workflow must have been created using File > New Activation Workflow).

**Step 2**
Launch the AWB.
If Prime Network is installed as a standalone product, launch the AWB as follows:
- Go to http://gateway-ip:6080/ana/services/install/install/webstart.html
- Click Cisco Prime Network Activation Wizard Builder.
If Prime Network is installed in suite mode with Cisco Prime Central, launch Workflow by selecting **Tools > Activation Wizard** in the Prime Network Vision GUI client.

The workflows on the gateway are loaded into the Workflows area in the AWB window.

**Step 3** Right-click the workflow and choose **Create Wizard**. AWB retrieves the workflow attributes and places them in the Unassigned Workflow Attributes area. These attributes define attributes appearance and behavior in the wizard. The first screen (Page1) is displayed in the screen area.

**Step 4** Complete the attributes for the first screen.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Read-only attribute name.</td>
</tr>
<tr>
<td>Display Name</td>
<td>Attribute display name in wizard.</td>
</tr>
<tr>
<td>Type</td>
<td>Attribute type. Note the following:</td>
</tr>
<tr>
<td></td>
<td>- Macro—Populated based on a macro that is supplied with Prime Network or you create.</td>
</tr>
<tr>
<td></td>
<td>- Combo—Creates a drop-down list of values (combo box).</td>
</tr>
<tr>
<td></td>
<td>- IMO Property—Allows selection of an object in the Prime Network Vision GUI.</td>
</tr>
<tr>
<td>Initialization</td>
<td>Attribute initialization sequence, determined by the Type.</td>
</tr>
<tr>
<td>Macro</td>
<td>Macro Config—List of available macros.</td>
</tr>
<tr>
<td></td>
<td>Macro Parameters—Lists attributes that can be selected to use as macro inputs.</td>
</tr>
<tr>
<td></td>
<td>Returns Single Value—If checked, the macro always returns a single value. The wizard field is a text box (not a combo box).</td>
</tr>
<tr>
<td></td>
<td>Enable Default Value—If checked, populates the value with the default value defined in Carry Over From Previous Iteration or Default Value Box fields.</td>
</tr>
<tr>
<td></td>
<td>Carry Over From Previous Iteration—If checked and wizard loops exist, initializes the attribute with value entered in the first loop. If not checked, enter the default value in box.</td>
</tr>
<tr>
<td>Combo</td>
<td>ID/Value—Combo ID and values.</td>
</tr>
<tr>
<td></td>
<td>Enable Default Value—If checked, populates the value with the default value defined in Carry Over From Previous Iteration or Default Value Box fields.</td>
</tr>
<tr>
<td></td>
<td>Carry Over From Previous Iteration—If checked and wizard loops exist, initializes the attribute with value entered in the first loop. If not checked, enter the default value in box.</td>
</tr>
<tr>
<td>IMO Property</td>
<td>IMO Type—Prime Network Information Model Object name.</td>
</tr>
<tr>
<td></td>
<td>Depth and Property Name—IMO type property to be retrieved.</td>
</tr>
<tr>
<td></td>
<td>Macro Name—List of available macros</td>
</tr>
<tr>
<td></td>
<td>Returns Single Value—If checked, the macro always returns a single value. The wizard field will be a text box and not a combo box.</td>
</tr>
<tr>
<td></td>
<td>Parameter Mapped To—Parameters that will be provided to the macro.</td>
</tr>
<tr>
<td></td>
<td>Enable Default Value—If checked, populates the value with the default value defined in Carry Over From Previous Iteration or Default Value Box fields.</td>
</tr>
<tr>
<td></td>
<td>Carry Over From Previous Iteration—If checked and wizard loops exist, initializes the attribute with value entered in the first loop. If not checked, enter the default value in box.</td>
</tr>
<tr>
<td>Required</td>
<td>Indicates whether the attribute is required.</td>
</tr>
</tbody>
</table>
Using the Activation Wizard Builder

Repeat Step 4 for additional screens.

**Step 5**

If you want to add loops (iterative presentations of the wizard screens), complete the following steps.

a. Click the **Loops** tab. **Figure D-12** shows the two Layer 3 VPN wizard loops. One loop consists of three pages (1,2,3) and the other contains one page (4). Both loops are required.

**Figure D-12** **AWB Loops Tab**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Validation</td>
<td>A regular expression that will be used to validate the operator entry. Examples include:</td>
</tr>
<tr>
<td></td>
<td>• ([0]+(100[1-9][0-9]</td>
</tr>
<tr>
<td></td>
<td>• ([Vv]([Ii]([Rr]</td>
</tr>
<tr>
<td>ToolTip</td>
<td>Text displayed when user hovers the mouse over the attribute name.</td>
</tr>
<tr>
<td>Example</td>
<td>(If Type=String) Sample text for the attribute. This advises users of the permissible entry format. (The text appears in light gray and is removed as soon as the operator clicks the field.)</td>
</tr>
<tr>
<td>Shown When Disabled</td>
<td>If checked, the attribute field always appears in the wizard, even if it is disabled (normally not done).</td>
</tr>
<tr>
<td>Hidden</td>
<td>Never displays the attribute in the wizard. (Useful for unused parameters or to maintain compatibility with older workflows.)</td>
</tr>
</tbody>
</table>

1. Loops tab
2. Loop actions
3. Loop attributes
b. Click **Add Loop**, then define the loop parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>User-defined loop name. Double-click the cell to enter a name.</td>
</tr>
<tr>
<td>Screen</td>
<td>Designates the screens to be included in the loop.</td>
</tr>
<tr>
<td></td>
<td>1. Double-click the cell, then click the ellipsis [...] on the right side of the cell to display the Screens dialog box.</td>
</tr>
<tr>
<td></td>
<td>2. Choose the screens from the drop-down list for each Order number.</td>
</tr>
<tr>
<td></td>
<td>3. Click <strong>OK</strong>.</td>
</tr>
<tr>
<td>Iterations</td>
<td>Name of attribute that will collect the number of wizard iterations completed by operator when they launch wizard. (This value is passed to the workflow.)</td>
</tr>
<tr>
<td>Required</td>
<td>If checked, indicates the loop is required.</td>
</tr>
</tbody>
</table>

**Step 6**

If you want to define attribute groups (attributes that will appear based on the values of another attribute), complete the following steps.

a. Click the **Groups** tab.

b. Click **Add Grouping**, and define the group parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>User-defined loop name. Double-click the cell to enter a name.</td>
</tr>
<tr>
<td>When</td>
<td>Read-only field used for a conditional display (defined in Any of the Values parameter).</td>
</tr>
<tr>
<td>Equal</td>
<td>Determines criteria for matching with Any of the Values field (defined in the Any of the Values parameter).</td>
</tr>
<tr>
<td>Any of the Values</td>
<td>Defines the parameters that cause the loop (per Name field) to appear:</td>
</tr>
<tr>
<td></td>
<td>1. Double-click the cell, then click the ellipsis on the right side of the cell.</td>
</tr>
<tr>
<td></td>
<td>2. In the Any of the Values dialog box, complete the following parameters:</td>
</tr>
<tr>
<td></td>
<td>- When Field—Determines the current attribute conditional display (attributes are defined in Screens tab).</td>
</tr>
<tr>
<td></td>
<td>- Equal To—If checked, group attributes will appear if the attribute value is equal to the value specified in ID/Value. (If unchecked, group attributes will appear if it is not equal.)</td>
</tr>
<tr>
<td></td>
<td>- ID/Value—Value the attribute must match (or not match) in order to be displayed.</td>
</tr>
<tr>
<td></td>
<td>3. Click <strong>OK</strong>.</td>
</tr>
</tbody>
</table>
Using the Activation Wizard Builder

Appendix D      Creating Workflows and Wizards Using the Workflow Editor and Activation Features

Step 7  Validate the wizard by clicking Wizard > Validate Wizard. The validation command checks the wizard’s internal model, screens, groups, loops, and XML. If an error occurs, an explanatory message is displayed.

Step 8  Save the wizard to the gateway by clicking Wizard > Upload to Gateway. See Adding an Activation Wizard to Prime Network Vision, page D-29.

After you upload the wizard to the gateway, you can make it available to users in Prime Network Vision.

Adding an Activation Wizard to Prime Network Vision

This procedure uploads a wizard to the gateway and identifies how the wizard will appear in the Activation menu in Prime Network Vision.

Once you add an activation, only users with Administrator privileges can delete activations and activation templates from the Prime Network Administrator GUI client. Executed activations are automatically purged from the Prime Network database according to the purging settings set by the administrator. For more information, see the Cisco Prime Network 4.0 Administrator Guide.

To add an activation wizard to Prime Network Vision:

Step 1  From the Wizard menu, choose Upload to Gateway. AWB performs a validation and displays the results.

Step 2  In the Upload Wizard Metadata to Server window, expand the wizard directory to where you want to upload the wizard.

If you want to upload the wizard to a new directory or to a new wizard name, you must create the directory first.

a. To create a new category, click Create Category, enter the new category name in the Create New Category dialog box, then click OK. The new category appears in the Upload Wizard Metadata to Gateway.
Using the Activation Wizard Builder

b. To create a new wizard name, click **Create Wizard**, enter the following wizard attributes in the Edit Wizard Properties dialog box.
   - Wizard Name—Enter the new wizard name.
   - Comment—Enter comments, if desired.
   - Version—Enter a version number, if desired.

c. Click **OK**. The new category appears in the Upload Wizard Metadata to Gateway window.

*Figure D-13 Creating a New Wizard Category and Wizard in the Network AWB Directory*

Step 3 Highlight the wizard in the Upload Wizard Metadata to Gateway window and click **OK**.

If you selected a new wizard, the upload occurs immediately, and you will see an Upload successful message.

If you selected a wizard with a wizard already associated to it, the Upload Wizard Metadata to Gateway will ask you to choose an option (as shown in Figure D-14).
**Running an Activation and View the Results**

Activations can be run from the Prime Network Vision GUI client. When you run an activation, you may be prompted for your credentials, or Prime Network may display a warning message. These are global settings that are controlled from the Administration GUI client.

**Note**
The Cisco Developer Network (CDN) has some scripts that you can use as examples for using the framework. Other activation scripts are only available through Cisco Advanced Services.

**Step 1**
From the Vision main menu, choose **Activation > Activation**. This opens a menu that lists the activations that the user can run, depending on their user access role.

**Step 2**
Expand the tree and highlight the activation wizard you want to run, and click **Next**.

**Step 3**
Enter all of the required data. You can only run activations on devices that are within your device scope.

**Step 4**
Check your entries and preview your changes:

1. Click the User Input tab and check all of the values you entered.
2. Click the Preview Configuration tab, which displays and validates the CLI commands that will be run on the device. It also highlights any errors so that you can make corrections to your input.
Step 5  Run the activation.

**Note** If you are prompted to enter your credentials, they will be used for every subsequent execution of a command or activation in the same GUI client session. If you want to change the credentials, click **Edit Credentials**.

Step 6  View the output:

a. Select the activation in the Activation History window, right-click and choose **Show Output**. The information presented is similar to the data displayed in Step 4 except it reflects the real runtime results.
   - Workflow Output—The sequence of commands that were run on the devices.
   - CLI Output—The actual CLI commands that were executed for the selected activation (for activations with an **Add** operation and a **Done** state).

b. If you want to view the output at a later time, export the activation to a local drive by clicking **Export to File**. We recommend that you do not change the file type.

### Synchronizing the Wizard and a Workflow When a Workflow is Modified

If you update a workflow template that is called by a wizard, you must synchronize the wizard with the updated template using the following procedure.

Step 1  Verify that the updated workflow template has been deployed to the gateway (using Prime Network Workflow).

Step 2  From the Wizard menu, choose **Download from Gateway**.

Step 3  In the Download Wizard Metadata from Gateway dialog box, expand the wizard tree, choose the wizard you want to download and click **OK**.

Step 4  After the wizard is downloaded to the AWB, from the Wizard menu, choose **Sync with Workflow**. The Wizard and Workflow Synchronization window displays the changes between the workflow and the wizard.

Step 5  When the AWB discovers the changes, it will display a dialog box asking for confirmation to merge the changes. Click **Yes** and **OK** to merge the workflow changes into the wizard.

Step 6  After the changes are merged, define the merged items. (Use the same instructions that are described in **Creating a New Activation Wizard**, page D-25.)

Step 7  Validate the wizard by clicking **Wizard > Validate Wizard**. The validation command checks the wizard’s internal model, screens, groups, loops, and XML. If an error occurs, an explanatory message is displayed.

Step 8  Save the wizard to the gateway by clicking **Wizard > Upload to Gateway**.

Step 9  In Upload Wizard Metadata to Gateway, choose the overwrite option to update the old workflow template with the new workflow template’s settings.

Step 10  Test the wizard from Prime Network Vision.