



## Creating Commands and Command Scripts to Perform Device Configuration Operations

---

Prime Network provides two tools for creating commands and scripts that perform device configuration operations: Command Builder and Command Manager. Commands can range from simple show commands to command scripts that contains wizards with multiple pages and input methods, such as check boxes and drop-down lists. After you create command scripts, you can add them to the Vision GUI client. Users with the required privileges can run the command scripts by right-clicking an NE's **Commands** menu. To get started, see these topics:

- [How is Command Manager Different From Command Builder?, page 4-1](#)
- [Checking Global Command Settings Before You Begin Using Command Manager, page 4-2](#)
- [Using Command Manager to Create and Execute Device Configuration Command Scripts, page 4-3](#)
- [Using Command Builder to Create Device Configuration Command Scripts, page 4-21](#)

### How is Command Manager Different From Command Builder?

Command Manager uses the Command Builder back-end, but unlike Command Builder, it also provides a command repository that contains all of the device-level commands and command scripts on the Prime Network gateway. Command Manager also allows you to execute commands on multiple devices at one time. You can organize commands by assigning versions to commands, or group commands into user-specified packages. For bulk configurations, you can do parameter substitution by importing data from a file.

Command Manager supports creating commands and command scripts at the network element level (IManagedElement). Therefore, the command repository only displays scripts that are available at the device level. To create commands at other levels, use Command Builder.

# Checking Global Command Settings Before You Begin Using Command Manager

While Command Manager and Command Builder let you specify the required user access level for commands, Prime Network has some global settings which will also affect authentication and authorization:

- Whether users must enter their credentials before running command scripts
- Whether users can schedule device configuration jobs
- Whether a warning message is displayed when users run command scripts

These settings are controlled from the Administration GUI client and are described in the following sections.

## Controlling Who Can Run Command Scripts

Users can execute commands from the Vision GUI client when a device is in their scope and they have sufficient privileges. You specify the required privileges when you create the command's basic information. However, the following two settings in the Administration GUI client will also affect command operation. (These are in the Administration GUI client under **Global Settings > Security Settings > User Account Settings**.)

Global Setting	How It Affects Command Manager and Command Builder
Execution of Commands	If enabled ("Ask for user credentials when running device configuration operation" check box is checked), requires users to enter their credentials when they execute a command from a device's right-click Commands menu in the Vision GUI client, or when commands are run from Command Manager. This only applies to commands that are executed <i>immediately</i> (it does not apply to scheduled commands). Provisioning and Audit events display an additional column that lists the username. This setting is disabled by default.
Job Scheduling	<p>Enables and disables the global per-user job authorization mode. This affects Command Manager jobs, and commands that are <i>scheduled</i> using the device right-click Commands menu in the Vision GUI client.</p> <p>If the mode is enabled, job scheduling privileges are controlled by the setting in the individual user accounts.</p> <ul style="list-style-type: none"> <li>• If this mode is enabled and a user is granted privileges, the user can schedule jobs across the product.</li> <li>• If this mode is enabled and a user is not granted privileges, the job scheduling features in the user's GUI clients are disabled.</li> </ul> <p>If the global per-user authorization mode is disabled, all users can schedule jobs; the setting in the users's account is ignored. This mode is disabled by default (all users can schedule jobs).</p>

For more information on these global settings, see the [Cisco Prime Network 5.1 Administrator Guide](#).

## Displaying a Warning Message When Users Run Commands

Prime Network can be configured to display a warning message whenever users execute commands. This affects Command Manager jobs, and any commands that are executed using the device right-click Commands menu in the Vision GUI client.

Users must acknowledge the message before proceeding. This feature is disabled by default. To enable it and specify a message, choose **Global Settings > Commands** in the Administration GUI client, enable the feature, and enter the message text. When users run commands, they will have to acknowledge the warning message or Prime Network will not allow them to proceed.

# Using Command Manager to Create and Execute Device Configuration Command Scripts

These topics describe how to use Command Manager to create device configuration commands and command scripts:

- [What is Command Manager?, page 4-3](#)
- [Steps for Creating and Deploying Commands with Command Manager, page 4-4](#)
- [Command Manager Window and Repository, page 4-5](#)
- [Creating and Testing a Command or Command Sequence, page 4-7](#)
- [Running Commands Using Input Stored in a File, page 4-18](#)
- [Controlling Command Manager Jobs, page 4-19](#)
- [Moving, Importing, and Exporting Commands and Packages, page 4-19](#)
- [Deleting Commands and Packages, page 4-20](#)
- [Deleting Commands and Packages from the Prime Network Command Repository, page 4-21](#)

## What is Command Manager?

Like Command Builder, Command Manager provides a GUI framework for creating programmable SNMP or Telnet commands. Commands can include multiple input screens and multiple input methods—text fields, check boxes, drop-down lists, and so forth. Commands can include parameters that are fixed. You can also perform bulk configurations using XLS or CSV files to populate the parameter values

Command Manager uses the Command Builder back-end, but it provides the following additional features:

- View the command repository, which contains all device-level commands available on the system
- Execute commands on multiple devices
- Use input from files to provide parameter substitution
- Apply versions to individual commands
- Create command packages

However, Command Manager commands can only be applied at the network element (IManagedElement) level, while Command Builder commands can be associated with any existing object group (IMO), type, or instance. This is why only device-level commands are listed in the command repository.

Once a command is created, it is added to the central repository and can be made available to other NEs of the same family, NE type, or software version, and can be executed from the Vision GUI client by users according to their device scope and privileges. Users with Configurator privileges can also create simple command sequences, although command sequences can only be run from the command repository (not from the Vision GUI client).

Command information is stored in XML files that can be organized into packages. The Command Manager repository always contains these two packages:

- **Default**—Contains all commands that are provided with the Prime Network product. Out-of-the-box commands in the Default package cannot be moved or exported.
- **Production**—Contains any user-created commands that were migrated from another installation. For a new installation, the Production package will be empty.

As you create new commands, you can place them in new packages and reorganize them as needed. You can also export and import command packages. Commands installed from Device Packages are listed only if Prime Network is restarted.

Command Manager jobs are purged according to the Prime Network job purging settings. Commands information (including rules, parameters, and script implementations) are stored on the gateway and are not purged; only Command Manager *jobs* are purged.

Commands created with Command Manager can be used by Transaction Manager. Commands can also be integrated into external configuration applications using the Prime Network API. For information on how to do this, see the [Cisco Prime Network Integration Developer Guide](#).

## Steps for Creating and Deploying Commands with Command Manager

The following table lists the steps you must follow to create a command or command sequence.

	Step Description	See:
Step 1	Check the settings that may affect who can run commands you create, and whether a warning message is displayed when users run a command.	<a href="#">Checking Global Command Settings Before You Begin Using Command Manager, page 4-2</a>
Step 2	Create a package for the command, giving it a name that is meaningful for its contents (such as a device type). We recommend not using the Default package to keep user-created commands separate from out-of-the-box commands (for ease of management).	<a href="#">Creating a New Package, page 4-7</a>
Step 3	Provide a name for the command (that will appear in the NE right-click menu) and specify the required user access level (this is the command's basic information).	<a href="#">Defining a Command's Basic Information, page 4-8</a>

	Step Description	See:
<b>Step 4</b>	<p>Define the properties that determine the format and structure of the command input form. These are called the <i>Execution Parameters</i>.</p> <p>For each parameter, define the type of input the user must provide (a string, a choice from a drop-down list, a checked check box, and so forth). A command can have multiple parameters, or multiple input pages (tabs), each with their own parameters.</p> <p>If you use Bean Shell scripts in the next step, you can use the script logic to populate and validate the parameter values.</p>	<a href="#">Defining Execution Parameters for a Command, page 4-9</a>
<b>Step 5</b>	<p>Specify the devices and software versions that the command can be run on, along with the scripts that should run when a user executes the command. These are called <i>Command Implementations</i>. Scripts can use:</p> <ul style="list-style-type: none"> <li>• Bean Shell over Telnet, SNMP, or TL1; or</li> <li>• Cisco Prime Network Macro language (over Telnet only).</li> </ul> <p>You can also specify rollback scripts and the conditions that indicate the command has failed.</p>	<a href="#">Defining the Implementations (Scripts) for a Command, page 4-12</a>
<b>Step 6</b>	Test the command locally by previewing each implementation.	
<b>Step 7</b>	Create the command.	
<b>Step 8</b>	Test the command to see how it will appear to the user, review the script execution, and execute the command (which deploys the command to applicable devices).	<a href="#">Previewing a Command Script and Checking Command Job Results, page 4-16</a>
<b>Step 9</b>	Check the command job status.	<a href="#">Controlling Command Manager Jobs, page 4-19</a>

## Command Manager Window and Repository

Launch the Command Manager from Change and Configuration Management, as shown in the following figure. You may have to expand the title bar by clicking the arrow at the top left.



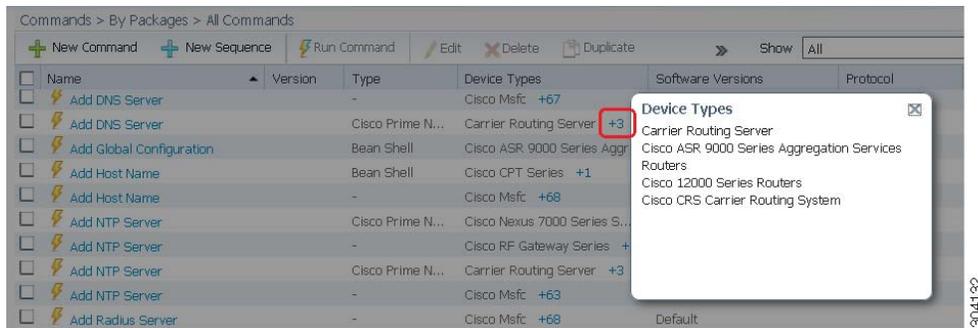
When you select **Command > Command Repository**, Prime Network displays all of the device-level commands in the system. The Default package contains the out-of-the-box commands provided with Prime Network. The Production package is only populated in upgrade scenarios and contains the commands that were migrated from the older deployment. Commands can be executed from the repository or from the devices (in the Vision GUI client). Commands displayed in the repository are filtered according to the user's access level.

The command repository only lists device-level commands (that is, commands at the IManagedElement level). For commands at other levels, use Command Builder.

Commands are indicated by a yellow lightning bolt and sequences by a blue lightning bolt. You can get the following information from the repository:

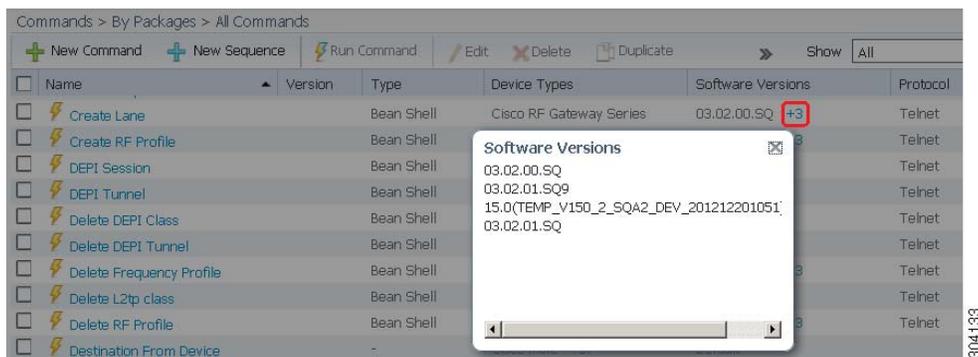
- Find out whether the command can be executed on different device types. This is indicated by a hyperlink in the Device Type column, as shown in the following figure. In this example, the Add DNS Server command can be executed on all CRS, ASR 9000, and 12000 routers. This also indicates that if those devices appear in a Vision GUI client map, the Add DNS Server command can be executed from the device's right-click Commands menu.

**Figure 4-1** Command Manager Repository Showing Applicable Device Types for a Command



- Find out if the command can be executed on multiple software versions (for a single device type). This is indicated by a hyperlink in the Software Version column. In the following example, the Create Lane command can be executed on a Cisco RF Gateway Series device that is running any of the software versions that are listed.

**Figure 4-2** Command Manager Window Showing Applicable Software Versions for a Command



- Find out the number of jobs associated with the command. This is indicated by a hyperlink in the # Of Jobs column which displays the Command Manager Jobs page. Depending on the user access role the jobs are displayed for the selected command.
- Find out the last run result of the command. This is indicated by a hyperlink in the Last Run column, as shown in the following figure. In the following example, the details of each job and its result associated with the command Device Log is displayed.

Task ID	Device Name	Command Name	Status
44044	10.104.120.188	Device Log	X
44050	10.104.120.189	Device Log	X

Users with Administrator and Configurator privileges can perform all operations on all commands—editing, executing, deleting, moving, and so forth. All other users can perform these operations if they have the required privileges for the command, and the device is in their scope.

## Creating and Testing a Command or Command Sequence

Commands and sequences must be created within a package. A best practice is to create a package with a name that describes the contents, such as the device series the command can be run on.

Commands can be duplicated and edited, but only if they are user-created commands. You cannot duplicate out-of-the-box commands that are packaged with Prime Network.

When you create a command (by clicking **Create** in the Command Implementations area), the command is saved to the database. It is not deployed to any NEs until you go to the repository, select the command, and click **Run Command**.

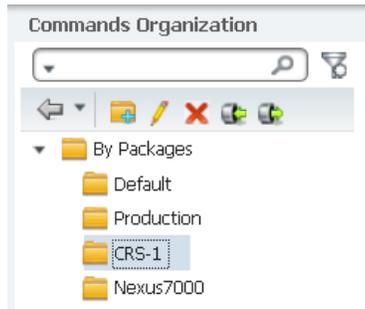
These topics provide the steps required to create, test, and deploy commands and command sequences:

- [Creating a New Package, page 4-7](#)
- [Creating a New Command, page 4-8](#)
- [Creating a Command Sequence, page 4-15](#)
- [Previewing a Command Script and Checking Command Job Results, page 4-16](#)

## Creating a New Package

Use the package mechanism to organize your command scripts. Naming packages by device type is a good practice, as shown in the following figure. If you need to create a new package, click the New Package icon on the upper left side of the window and provide a name and optional description.

Figure 4-3 Command Packages



If you delete any of the user-created packages (such as CRS-1 or Nexus7000 in the previous figure), all commands associated with package are moved to the Default package. This ensures that there will be no problems if the command is part of a scheduled job.

New commands that are installed from a Device Package are not listed in the command repository unless Prime Network is restarted. (Remember that the command repository only lists device-level commands; that is, commands at the IManagedElement level.)

To create a new command, proceed to the next section. To create a command sequence, see [Creating a Command Sequence](#), page 4-15.

## Creating a New Command

These are the steps for creating a new command script with Command Manager:

- [Defining a Command's Basic Information](#), page 4-8
- [Defining Execution Parameters for a Command](#), page 4-9
- [Defining the Implementations \(Scripts\) for a Command](#), page 4-12

### Defining a Command's Basic Information

The basic information for a command includes the name that will appear in the Vision GUI client (in the NE right-click menu) the user authorization that is required to use the command.

---

**Step 1** Select a package from the Commands Organization area.



**Tip** Do not use the Default package because it contains all commands provided by Prime Network. This makes it easier to your manage user-created commands.

---

**Step 2** Click **New Command** to open the command wizard builder.

**Step 3** Configure the command's basic information in the Command Info area.

Command Info Field	Description
Command Title	The name of the command. This is what will be displayed in the Vision GUI client when a user right-clicks an NE and chooses <b>Commands</b> .
Menu Category	The menu under which the command should be listed. This specifies where the command is displayed when a user right-clicks the NE in the Vision GUI client. For example, for a command named <i>new-command</i> : <ul style="list-style-type: none"> <li>• <b>Commands</b>—Would render <b>Commands</b> &gt; <i>new-command</i></li> <li>• <b>Commands/Custom</b>—Would render <b>Commands</b> &gt; <b>Custom</b> &gt; <i>new-command</i></li> </ul>
Version	A number or letter which helps you track instances of the same command. A best practice is to change this when you upgrade a device software image or add a new NE model.
Visible in menu	Whether users can see the command in the Vision GUI client. <b>Tip</b> Do not make commands visible until you have tested them.
Description	Command description. A best practice is to fill this field because it will make it easier to identify user-created commands from those that are supplied with Prime Network.
Authorization	User access level required to run the command. (The user will be permitted to execute the command only if the device is in their scope.)

**Step 4** Click **Next** to begin entering the command's implementation parameters.

## Defining Execution Parameters for a Command



### Note

If you are creating a command that requires no input—for example, a simple show command—skip this section and proceed to [Defining the Implementations \(Scripts\) for a Command, page 4-12](#).

The Execution Parameters determine the structure and format of the input form and the data that users must supply. Each parameter has an associated input type which determines how information is entered (for example, typing in a number or string, checking a check box, choosing from a drop-down list). You can specify:

- Required parameters and optional parameters.
- Parameters with default values which can be visible or hidden in the input form.

A command can have one parameter, multiple parameters, and multiple input pages (tabs) with their own parameters.

- Step 1** Define your first parameter by clicking **New Parameter** in the Execution Parameters area, and provide the basic information for the execution parameter.

Execution Parameters Field	Description
Title	The name of the parameter. This is what will be displayed in the command dialog when the user runs the command.
Variable Name	Parameter name. This entry must be unique and can contain only letters, numbers, hyphens (-), and underscores (_).
Tooltip	A tooltip (maximum of 256 characters).

- Step 2** Define the input type for the parameter.



**Tip** Use TextArea for multi-line parameter values (e.g., license, banner, and so on). Do not use it to configure multiple commands in a single parameter. Doing so will download all commands to the device, but the command output will only be partial.

For this type of user input:	Choose this type:
A type conversion	String
	Long
A number	Integer or float
Multi-line string that will be referenced as a single attribute (for example, information that can be cut and pasted (including end-of-line characters <CR><LF>))	TextArea
An IP address or IP subnet mask	IPSubnet
	IPAddress
A drop-down list of choices (see the following instructions)	Combo

If you choose **Combo**, an Add button appears in the wizard. Click **Add** to create choices for the combo box.

Add Combo Options	Description	Example	
Value	A value that will be associated with the option; for example, <b>1</b> .	<b>1</b>	<b>2</b>
Label	The description of the entry that is displayed in the selection list (drop-down list) of the input form, such as <b>Up</b> .	<b>Up</b>	<b>Down</b>

The example in the previous table would create a combo box with the choices Up and Down.

**Step 3** Define the rest of the execution parameter. The choices depend on what you specified in the Type field.

Execution Parameters Field	Description
default	The default value (a number, enabled, etc.). To force the user to enter a value (without specifying a default), leave it blank and choose Required.
Display width	The maximum width in characters.
Visible	Whether the parameter will be visible in the command dialog. If you uncheck the box, the parameter is hidden from the user but can still be used in the command with a default value. To provide a constant value but keep it hidden from the user, uncheck the visible check box and provide a default value in the Default area.
Required	Specifies the input as mandatory. Mandatory input is designated by a red asterisk in the command.
Group	Indicates that the parameter belongs to a specified group. Parameters in a group are displayed together inside a box, with the group name as their title. For example, you could create a group called Authentication with two parameters: Username and Password. Username and Password would be enclosed in a box entitled Authentication.

**Step 4** If you want the input to be either validated or populated when the user runs the command, click the **Validate** tab.

- On Populate—When Run Command is clicked, the BeanShell script should populate the parameters values according the script logic.
- On Validate—When Preview or Execute Now is clicked, the BeanShell script should validate the parameters values according the script logic.



**Note** When creating or editing On Populate or On Validate BeanShell scripts, you must restart the Vision GUI client for the changes to take effect.

**Step 5** Click **OK**, and the new parameter is added to the list of Execution Parameters.

**Step 6** Create more parameters as needed by repeating [Step 1](#) through [Step 5](#).

**Step 7** If you want to create additional pages (tabs) or adjust the order of parameters, use the tools illustrated in the following figure.



304140

**Step 8** When you have finished creating the parameters, click **Next** to begin entering the Command Implementations (the scripts that will be executed and the devices they can run on).

### Defining the Implementations (Scripts) for a Command



**Note** All scripts are run at the network element level (IManagedElement).

The Command Implementations are the scripts that are run when the user executes the command. Implementations can use Bean Shell (Telnet or SNMP) or Cisco Prime Network Macro language (Telnet only). You can also specify a rollback script and the conditions that indicate the command has failed.

When you create an implementation you also define which device types the command applies to. You can choose different degrees of granularity:

- All devices
- Device family (such as routers)
- Device series (such as Cisco 7200 routers)
- Device model (Cisco 7207)

However, if you choose all devices, you cannot specify multiple software versions.

When a command is deployed, it overrides any inherited command from a higher level and automatically applies to elements below it. For example, the following tables shows how command versions can be overwritten:

Chronological Actions	Results in these devices using this version:		
	7200 routers	7207 router	6509 switch
1. Version 1 is applied to all devices (family)	1	1	1
2. Version 2 is applied to only the Cisco 7207 (model)	1	2	1
3. Version 3 is applied to all Cisco 7200 routers (series)	3	3	1
4. Version 4 is applied to all switches (family)	3	3	4

**Step 1** Define the scripts that will run by clicking **New Implementation** in the Implementations area.

**Step 2** In the General tab, provide the following script information.

Command Implementations Field	Description
Name	A meaningful name for the implementation, such as a device type, device series, or technology. For example, one profile could be <i>command-AllDevices</i> , another could be <i>command-Cisco7600</i> . <b>Note</b> A best practice is to associate the profile name with the devices you are choosing.
Device Types	The NEs that this command can be run on.
Software Version	The device software versions that are available for the chosen devices. If you select a device series or a non-Cisco device, this choice is disabled. If you subsequently update the software version on a device, you must also update the command.
Type	The scripting language to be used for the command: <ul style="list-style-type: none"> <li>• CiscoCisco Prime Network Macro—Simple mode. (See <a href="#">Prime Network Macro Language</a>, page C-1.)</li> <li>• BeanShell—Programmable mode; full scripting language. (See <a href="#">BeanShell Commands</a>, page C-11.)</li> </ul>
Protocol	The protocol to use. If the command has multiple profiles, they must use the same protocol. <ul style="list-style-type: none"> <li>• Telnet</li> <li>• SNMP (BeanShell only)</li> <li>• TL1 (BeanShell only)</li> </ul> <p>Currently TL1 protocol is applicable only for CPT devices.</p> <b>Note</b> You can create commands using only those protocols that are enabled on the device type in the Vision GUI.
Timeout	The timeout value for the command, in milliseconds. If a timeout occurs, the command continues to run in the background and the result is displayed in the command execution result screen.

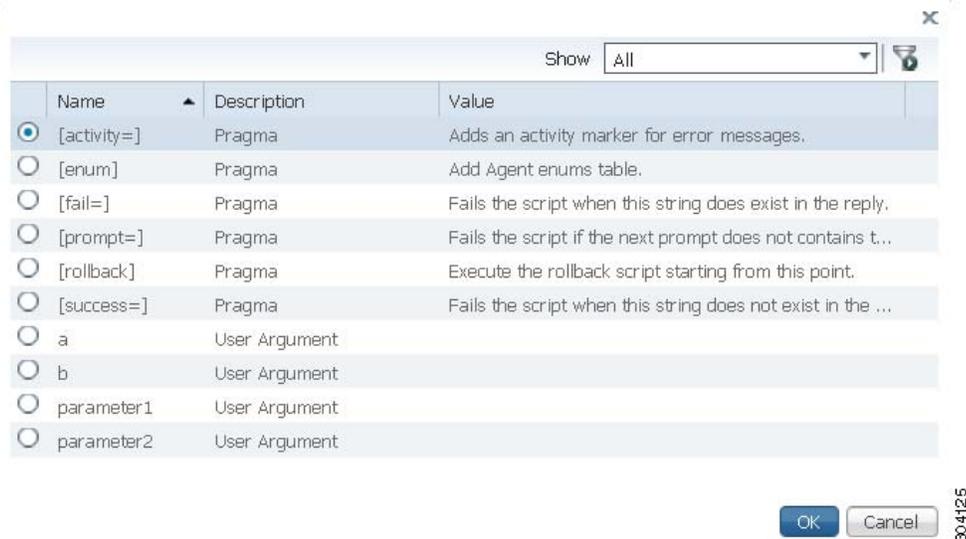
**Step 3** Click the Script tab to define the command script, using these guidelines:



**Tip**

A best practice is to create the script in a text editor, then cut and paste it into the Script area of the Script tab.

- To view all user-defined and built-in parameters in the Command Builder application, position the cursor in the Script or Rollback field and press **Ctrl-Spacebar**. A dialog box is displayed that lists all available arguments (containing both the user-defined input argument and the built-in properties of the IMO context). Also see [Supported Pragmas, page C-4](#).



Select an entry from the list and then click **OK** to add it to the Script or Rollback field.

- Enclose pragmas with square brackets: [...].
- To enter a required carriage return in the command line, enter the escape sequence **&cr**.
- It is possible to use multiple pragmas in a single line, in which case all pragmas are analyzed. If the same type of pragma is repeated, only the last one is used.

Field	Description
Script	The Telnet or script lines to be sent to the NE. The script lines can contain optional inline directives (pragmas) for finer granularity control. For more information about the supported pragmas, see <a href="#">Supported Pragmas, page C-4</a> .
Rollback	(Optional) The rollback script that is used when the command fails. <b>Note</b> If the rollback script fails, no additional actions are performed.
Failure Condition	(Optional) A general failure condition that applies to all script lines. To specify a failure condition: <ol style="list-style-type: none"> <li>Check the Failure Condition check box.</li> <li>In the Failure Condition field, enter the text that is to be looked for during script execution. If the specified text appears in the reply, the command is aborted.</li> </ol>

**Step 4** When you are finished with the implementation, click **OK**.

**Step 5** Create more profiles as needed by repeating [Step 2](#) through [Step 4](#).

- Step 6** Click **Create** when you are done. The newly created command is displayed in the command repository table. (It is not yet deployed to devices; that happens when you select **Run Command**.)
- Step 7** Test the command locally before deploying it. See [Testing the Command Locally Before Deploying It to Other NEs, page 4-35](#)

## Creating a Command Sequence

Users with Configurator privileges can create command sequences. Sequences can only be executed from the command repository (not from the Vision GUI client). You cannot specify different devices for the different commands in a sequence.

- Step 1** Choose a package. If you need to create a new package, click the New Package icon on the upper left side of the window and provide a name and optional description.



**Tip** Do not use the Default package because it contains all commands provided by Prime Network. This makes it easier to your manage user-created commands.

- Step 2** Click **New Sequence** to open the Create Command Sequence wizard.
- Step 3** Configure the sequences's basic information in the Command Sequence Info area. Note that there is no authorization level field; only users with Configurator privileges can create and run sequences. There is also no menu field because sequences can only be run from the command repository, not from the Vision GUI client.

Command Info Field	Description
Command Sequence Title	The name of the sequence. This is what will be displayed in the repository.
Command Sequence Name	The sequence name used and saved in the database.
Version	A number which helps you manage multiple instances of the command. A best practice is to change this when you have a command deployed to specific devices, and you want to update it for those devices.
Description	Command description. A best practice is to fill this field because it will make it easier to identify user-created commands from those that are supplied with Prime Network.

- Step 4** Add the commands to the sequence in the Define Sequence area by clicking **Add Commands**. A dialog opens that lists all device-level commands in the repository (from all packages, including the Default package).
- Step 5** Select the commands you want to add to the sequence and click **Add**.
- Step 6** Configure the commands in the sequence. (You cannot edit the command contents.)
- Ensure that each command is also configured with a rollback sequence in the Create Command Sequence area. You can add a rollback sequence from the command repository by clicking the **Add** link.
  - Adjust the command order.

- Step 7** Click **Create** when you are done. The newly-created command is displayed in the command repository and can be executed from the repository.
- Step 8** Test the command locally. See [Previewing a Command Script and Checking Command Job Results](#), page 4-16.
- 

## Previewing a Command Script and Checking Command Job Results

When you preview a command, you can see how it will appear to users. The preview feature allows you to see the expected script output without performing any real command executions.

Once you execute a command with **Create Execution Job** (at the end of the following procedure) Prime Network creates a command job and displays a dialog that contains a hyperlink to the job details. To check the job status for completed and schedule jobs, select **Command > Jobs** from the Change and Configuration Management menu.

- 
- Step 1** Select the command or sequence in the repository and click **Run Command**.
- Step 2** In the Select Device page, choose one of the following options:
- **By Devices**—Choose this option to select the device(s) that you want to run the command.
  - **By Groups**—Choose this option to select the device group(s) that you want to run the command. There must be at least one device added to a device group for the command to run on that group. If a device is added to multiple device groups that are selected, the command will run only once on that device. For information on how to set up a device group, see the [Setting Up CCM Device Groups](#) section.
- Step 3** From the command input forms that are generated, enter the required input.
- Step 4** To verify the parameters that you entered and see what the executed scripts will look like, click **Preview**.



---

**Note** Preview action is disabled for commands that use TL1 protocol.

---

- Step 5** If the result is satisfactory, click **Create Execution Job** to:
- For a command, deploy it to the selected NEs.
  - For a command sequence, save it to the repository.

- Step 6** Click the job hyperlink to view the job results. To get job details, select the job and click the hyperlink under the Last Run Result column. If the job was for a command sequence, as in the following example, you can check the results of each command in the sequence.

The screenshot displays the Command Manager Jobs interface. A job named 'sanitycommand-05-14-2013 02:26:51' is selected. The job details show it is 'Completed' with a 'Partial\_success' result. A table below lists the tasks and their statuses:

Task ID	Device Name	Command Name	Status
57461	10.77.204.36	sanitycommand	X
57462	10.77.204.43	sanitycommand	X
57463	10.78.179.144	sanitycommand	✓

- If the job failed or was only partially successful, click the red X under Status to get details about the failure. By hovering over the Device Properties icon, you can get quick device details about the reachability of the device.

The screenshot displays the Command Manager Jobs interface. A job named 'sanitycommand-05-14-2013 02:26:51' is selected. The job details show it is 'Completed' with a 'Partial\_success' result. A table below lists the tasks and their statuses:

Task ID	Device Name	Command Name	Status
57461	10.77.204.36	sanitycommand	X
57462	10.77.204.43	sanitycommand	X
57463	10.78.179.144	sanitycommand	✓

- If the job was successful, click the Transaction Log tab to see what scripts were sent to the device. Click the Debug Trace tab to view the information that was received from the device.

## Enabling E-mail Notification for Command Scripts

Command Manager enables you to send automatic e-mail notifications about the status of jobs to users while executing the command or job scripts. To enable email notifications for the Command Manager, you need to add the SMTP host name in E-mail Settings pane of the Configurations Settings page in CCM.

- Step 1** Choose **Commands > Command Repository**, select the command, and click **Run Command**.
- Step 2** In the **Run Command** window, click the **Create Execution Job** pane, select the **Email Id(s)** checkbox, and enter the email id of the recipient, for example, user1@cisco.com. You can enter comma separated email ids for multiple recipients.
- Step 3** Click **Create Job**. The e-mail is sent to the recipient with the following information:

- Job ID and Job spec ID
- Job name and type
- Job schedule time
- Job completion time
- Name of the device on which the job is executed
- Command name
- Status of the job executed

**Step 4** Choose **Commands > Jobs**, select the job in the **Command Manager Jobs** page, and click **Edit**.

**Step 5** In the **Edit Job** window, click **Create Execution Job**.

**Step 6** In the **Email Id(s)** text box, enter the email ids of the recipient and click **Save**. You can enter comma separated email ids for multiple recipients. The e-mail is sent to the recipient with the following information:

- Job ID and Job spec ID
- Job name and type
- Job schedule time
- Job completion time
- Name of the device on which the job is executed
- Command name
- Status of the job executed

## Running Commands Using Input Stored in a File

For bulk configurations, you can create an XLS or CSV files with parameter values and import the file into the command.

**Step 1** Create a file that contains the parameter values.

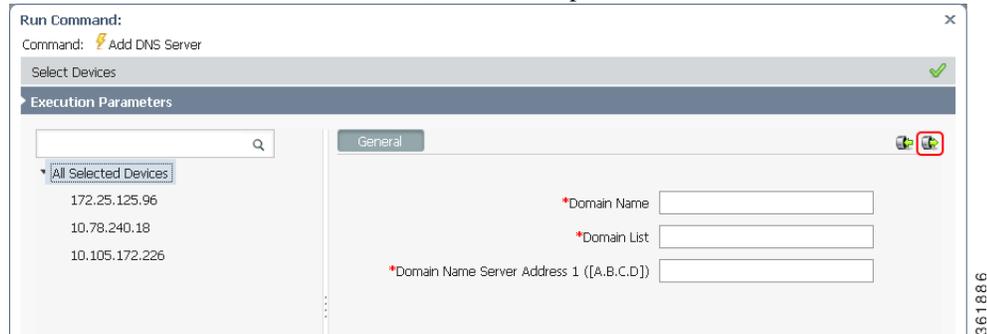
- Select the command from the repository and click **Run Command**.
- Select the devices you want to configure and click the **Export** icon at the top right of the screen.
- In the Export Execution Parameters dialog, click **OK** to continue with the export operation.
- When your browser displays its file management dialog, download the file. The file is created with the name *command.csv*.

**Step 2** Populate the *command.csv* file with your desired attribute values. One column is dedicated to each attribute.

**Step 3** Run the command using your *command.csv* file as input.

- Select the command in the repository and click **Run Command**.
- Select the devices you want to configure. They must match the devices in your spreadsheet.

- c. In the Execution Parameters area, click the import icon.



- d. Navigate to the file that contains your input, and click **OK**.
- e. In the Command Execution Job area, specify when the job needs to be executed (on daily, weekly, monthly, or on a specif time interval) and the e-mail ID(s) to which to send a notification after the scheduled job is complete. Click **Create Job**.

**Step 4** Run the command.

## Controlling Command Manager Jobs

Users with Configurator privileges can perform all job actions from the **Command Manager Jobs** window. This includes cancelling jobs, suspending jobs, rescheduling jobs, and editing job properties (such as changing the selected devices). Other users can only manage jobs that they own.

Command Manager jobs are automatically deleted according to the Prime Network job purging settings. By default, these jobs are never purged.

For examples of how to view job results, see [Previewing a Command Script and Checking Command Job Results](#), page 4-16.

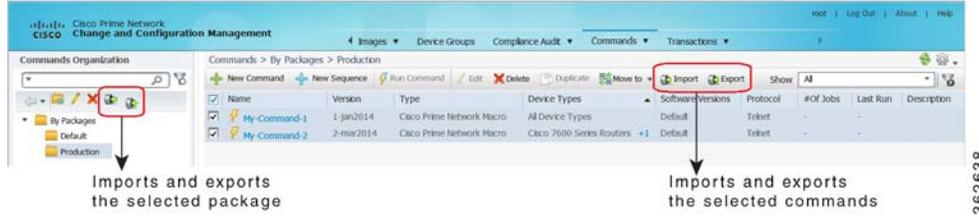
## Moving, Importing, and Exporting Commands and Packages

Command Manager allows you to move, export, and import user-created commands. User-created commands are any commands that do not reside in the Default repository (which contains all commands that were provided with the Prime Network product).

To move user-created commands to a different package, select the commands in the repository, click the Move To icon, and choose the package.

You can also export and import user-created commands and packages. Exporting commands is useful when you want to clean up the repository or if you create commands that you plan to use at a later time. [Figure 4-4](#) shows from where you should launch an import or export operation, depending on whether you are working with packages or individual commands.

Figure 4-4 Importing and Exporting Commands and Packages



Exported items, whether they are individual commands or a package of commands, are saved in a zip file on the client machine or on the gateway server. On the gateway server, commands are stored in *NETWORKHOME/Command\_Archive*. If you do not supply a name in the export dialog box, Command Manager uses the following naming scheme:

Export ed Item	Default Name
Individual command	<i>command-name_date_time</i>
Group of commands	<b>Commands</b> _date_time
Package	<i>package-name_date_time</i>

Files you export to your local system are saved in your browser's download directory.

Import operations will query you for the location of the zip file that contains the commands or package. If you try to import a command that already exists in the command repository, you are prompted to overwrite the command or exit.

## Cancelling Command Manager Jobs

To cancel the job that is in the running state:

- 
- Step 1** Select the job in the **Command Manager Jobs** window.
  - Step 2** Click **Cancel**.  
The **Job Status** displays as Cancelled and the **Lastrun Status** displays as Cancelled with a hyperlink in the **Command Manager Jobs** window.
  - Step 3** Click the Cancelled hyperlink in the **Lastrun Status** field to display the **Command Manager Job Result** window.  
The job result display the information about successful, unsuccessful, and cancelled tasks.
- 

## Deleting Commands and Packages

The following procedures explain how to:

- Remove a command from the Vision GUI client
- Remove a command or package from the command repository

## Removing a Command from the Vision GUI Client

If you want to remove a user-created command from the Vision GUI client, you can hide it without removing it from the repository. You cannot hide out-of-the-box commands that are provided with Prime Network.

**Note**

---

This operation may take some time depending on the number of NEs that will be affected.

---

To remove a command from the Vision GUI client:

---

- Step 1** Select the command in the repository and click the **Edit** icon.
- Step 2** In the Command Info page, deselect the Visible in Menu check box.
- Step 3** Click **Save**. The command is hidden in the Vision GUI client.
- 

## Deleting Commands and Packages from the Prime Network Command Repository

You can only delete user-created commands and packages. Out-of-the-box commands cannot be deleted. In addition, user-created commands can only be deleted if they are *not* part of:

- A scheduled job
- A command sequence

When a command is successfully deleted, it is removed from the repository and the Vision GUI client, and all related command data is deleted from the gateway and database.

When you delete a *package*, none of the user-created commands in the package are deleted. Instead, the commands are moved to the Default package, from where you can delete them.

To delete a command, make sure that you are working from the right package in the Commands Organization area. Then choose the command and click the Delete button above the commands table.

To delete a package, select the package and use the Delete button in the Commands Organization area.

# Using Command Builder to Create Device Configuration Command Scripts

These topics explain how to use Command Builder to create commands and make them available to other NEs:

- [What is Command Builder?](#), page 4-22
- [Steps for Defining a New Command](#), page 4-22
- [Command Builder Window](#), page 4-23
- [Creating a Command Using Command Builder](#), page 4-24
- [Testing and Publishing Commands](#), page 4-35
- [Exporting Commands with Command Builder](#), page 4-38
- [Importing Commands with Command Builder](#), page 4-38
- [Deleting Commands](#), page 4-39

## What is Command Builder?

Command Builder is an integral part of the Prime Network Vision GUI. It is used to create custom commands to be executed on specified devices. These commands can be tested and then published, which enables a wider scope of NEs to use them. You can publish commands to NEs of the same family, NE type, or software version.

Command Builder commands execute a programmable sequence of SNMP or Telnet command lines. Commands can be created using the BeanShell scripting language or the Prime Network Macro Language. Commands can be associated with any existing object group (IMO), type, or instance. This enables access to the live information of the network object with which the command is associated.

Only users with Administrator or Configurator privileges can create commands using Command Builder.

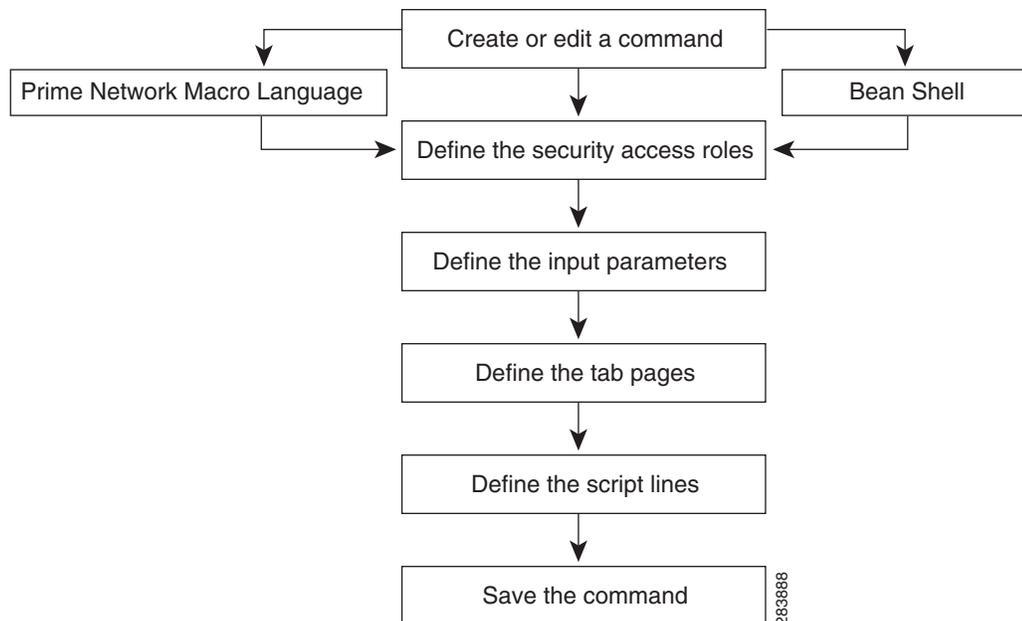
Commands can be used in a variety of ways:

- Using Command Builder, add the commands to the Prime Network Vision GUI client. Users will be able to run the commands by right-clicking an NE. When running the command, the user will be presented with a dialog for entering the input parameters that were defined during the command creation process.
- As building blocks in the creation of transactions (workflows) which can then be deployed to devices using Transaction Manager.
- Use the Prime Network API to integrate the commands into external configuration applications. For information on how to do this, see the [Cisco Prime Network Integration Developer Guide](#).

## Steps for Defining a New Command

These steps describe how to create a new command definition using Command Builder and the order in which the steps must be performed.

**Figure 4-5** Steps for Defining a New Command



At any time after the command has been defined, it can be tested, executed, and published to a wider scope of managed elements and network elements.

For more information, see [Creating a Command Using Command Builder, page 4-24](#).

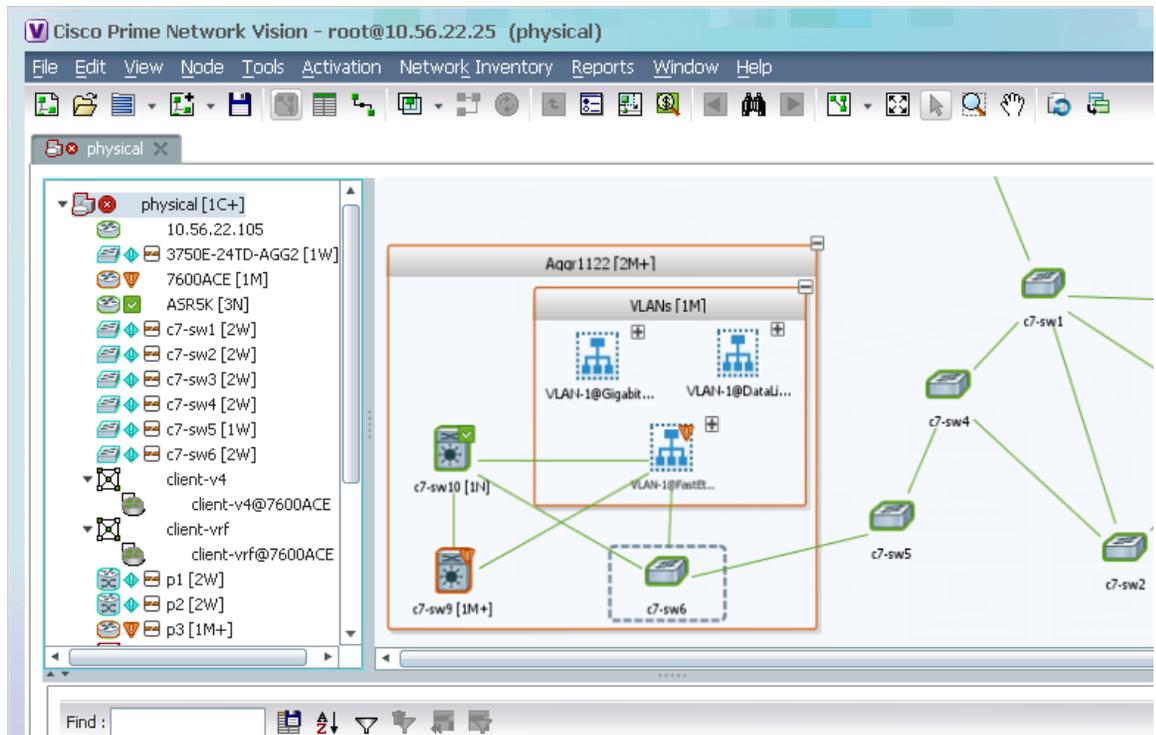
## Command Builder Window

Command Builder is launched by right-clicking a specific managed element or a selected object within a managed element and selecting **Management > Command Builder**.

The Command Builder window lists all existing commands that are available to the NE. (The technology-based commands that are packaged with Prime Network are not editable).

The commands that are listed depend on the NE type and the user's access privileges.

**Figure 4-6** Command Builder Window



**Table 4-1** Command Builder Window

Column	Description	Example:
Name	Name for the new command (also the filename).	ShowUsers
Menu Caption	The title that will appear in the command's GUI window when it is launched from an NE.	List of Current Users

Table 4-1 Command Builder Window (continued)

Column	Description	Example:
IMO Context	The inventory object associated with this command.  <b>Note</b> Commands are associated with a selected object within a managed element, enabling the command to use the object properties in the command definition. For example, if you select a port object, the portAlias and port status properties are automatically made available to the command.	IManagedElement
Local	Specifies whether the command is inherited from a higher level (false) or is defined locally on the selected managed element (true).	false

Table 4-2 identifies the buttons that appear in the Command Builder window.

Table 4-2 Command Builder Window Icons

Button	Name	Description
	New Element	Creates a new command definition on the local client machine.
	Edit Element	Edits an existing command definition on the local client machine. When you edit an existing command, the changed command is saved as a local instance on the client machine. As with a new command, you can then publish it to make it available to other NEs.  You can also edit commands that are packaged with Prime Network (for example, OAM commands that are launched from the right-click <b>Commands</b> menu). This is done by creating a clone of the command, which you can then edit.
	Delete Element	Deletes a command that exists on the local client machine (has not been published).
	Export Element	Saves a full command definition on the gateway.
	Import Element	Imports a command definition from the gateway to a local client machine. If the command already exists, it is overwritten (updated).
	Hierarchy Manager	Controls the NEs from which the command can be executed.
	Run Command	Previews and executes a command. After you create, modify, or update a command, we recommend that you wait a few seconds before executing it. If the gateway is busy, waiting a few seconds allows sufficient time for Command Builder to obtain the correct version from the registry.

## Creating a Command Using Command Builder

When you create a new command, you first create a local instance of the command. After you have tested the command, you can publish it to make it available to other NEs.

To create a command:

- Step 1** Launch Command Builder by right-clicking a managed element and choose **Management > Command Builder**. The window lists all existing commands that are available to the NE. (The technology-based commands that are packaged with Prime Network are not editable).
- Step 2** Click the **New Element** icon in the toolbar to open the New Command dialog box.
- Step 3** Enter the following information for your new command:

Field	Description
Name	Name for the new command (also the filename), such as <b>ShowUsers</b> .
Menu Caption	The title that will appear in the command's GUI window when it is launched from an NE, such as <b>List of Current Users</b> .
Menu Visible	Indicates whether or not the command should appear in the Prime Network Vision GUI client. For example, if you only want to use execute the command via the API, do not check the check box.
Menu Path	(If visible) The name that will appear in the NE's right-click menu (such as <b>Commands</b> ).
Context IMO	The object associated with this command. The object is within a managed element. All subobjects that do not appear in the inventory tree (such as parameter groups of a port) are listed in a drop-down list.
Timeout	The timeout value for the command, in milliseconds. The default value is 120000 milliseconds (2 minutes). If a timeout occurs, the command continues to run in the background and the result is displayed on the Run Command screen.
Language	The scripting language to be used for the command: <ul style="list-style-type: none"> <li>• Prime Network Macro—Simple mode. (See <a href="#">Prime Network Macro Language, page C-1</a>.)</li> <li>• BeanShell—Programmable mode; full scripting language. (See <a href="#">BeanShell Commands, page C-11</a>.)</li> </ul>
Protocol	The protocol to use: <ul style="list-style-type: none"> <li>• Telnet</li> <li>• SNMP (BeanShell only)</li> <li>• TL1 (BeanShell only)</li> </ul>
Warning Visible check box	This check box and its accompanying message text field appear only if the system is set to generate a message upon command execution warning the user that the command could potentially interrupt network services. You can deselect this check box to override this setting for the current command. When the user executes this command, no message will be shown.
Warning Text	Shows the message that will be displayed to users upon command execution. You can change the message as required. The text field is disabled if the Warning Visible check box is deselected

- Step 4** Click **Next**. The Command Authorizations dialog box is displayed.
- Step 5** Select the security access roles that are authorized to execute the command.

**Step 6** Click **Next**. The User Input Arguments dialog box is displayed.

You can define any number of input parameters. These parameters determine the structure and format of the input form. The input form is automatically generated when the command is executed.

You can use either of two types of command parameters: built-in parameters and user-defined parameters, both of which are replaced in runtime. All parameters (both built-in and user-defined) are available during command editing via a selection list. For more information about Prime Network Macro Language, see [Prime Network Macro Language](#), page C-1.

**Step 7** To add a new argument, click **New**. The Add/Edit User Argument dialog box is displayed ([Figure 4-7](#)).

**Figure 4-7 Add/Edit User Argument for Command Dialog Box**

The screenshot shows a dialog box titled "Add/Edit User Argument for Example Command." with a close button (X) in the top right corner. Inside the dialog, there is a text prompt: "Please complete the following metadata information for the argument." Below this are several input fields: "Name:" (text box), "Caption:" (text box), "Type:" (dropdown menu showing "String" and a "..." button), "Width:" (text box with "15"), "Visible:" (checkbox checked), "Tooltip:" (text box), "Default:" (text box), and "Required:" (checkbox unchecked). At the bottom right of the main area is an "Advanced" button. At the very bottom of the dialog are "OK" and "Cancel" buttons. A vertical number "281725" is visible on the right side of the dialog box.

**Step 8** Enter the required information:

Property	Explanation
Name	Parameter name. This entry must be unique and can contain only letters, numbers, hyphens (-), and underscores (_).
Caption	Parameter display name. This entry is displayed in the Command Builder execution window.

Property	Explanation
Type	<p>The type of input that will be required upon execution of the command. Only input values that are valid for the selected type are accepted. These values are validated during runtime.</p> <ul style="list-style-type: none"> <li>• String</li> <li>• Integer</li> <li>• IPSubnet</li> <li>• Combo—For more information about defining a Combo field type, see <a href="#">Defining a Combo Field Type for a Command Builder Command, page 4-30</a>.</li> <li>• IP Address</li> <li>• Float</li> <li>• Long</li> <li>• Text Area—Multi-line string field type into which you can cut and paste information, including end of line characters &lt;CR&gt;&lt;LF&gt;. This is referenced in the script as a single attribute.</li> </ul> <p> <b>Note</b> This feature should be used only for multi-line parameter values (e.g., license, banner, and so on). It is not intended for configuration of multiple commands in a single parameter. Doing so will download all commands to the device, but present only a partial output in the command builder output</p>
Width	Field width, in number of characters. Relevant for the Command Builder execution window.
Visible	<p>Indicates whether this parameter appears in the Command Builder execution window. Check this check box to display the parameter, or uncheck the check box to hide the parameter from the user. If the argument is hidden, it can still be used in the command (with its default value).</p> <p><b>Note</b> When the parameter is not visible and has been assigned a default value, it can serve as a constant argument.</p>
Tooltip	Specifies a tooltip for the command parameter. This string is displayed for the parameter field in the input form (see <a href="#">Testing the Command Locally Before Deploying It to Other NEs, page 4-35</a> ). The tooltip can be a maximum of 256 characters.
Default	A default value for the parameter.
Required	Indicates whether the argument is mandatory or optional. The mandatory arguments are displayed in bold font in the input form (see <a href="#">Testing the Command Locally Before Deploying It to Other NEs, page 4-35</a> ).

Click the Advanced tab to open the Add Argument Advanced Controller dialog box.

The Advanced option enables you to do additional actions on the parameter values using BeanShell.

- On Populate—The BeanShell script, executed when you click Run Command, can be used to populate the parameters values according the script logic.
- On Validate—The BeanShell script, executed when you click Preview or Execute Now, can be used to validate the parameters values according the script logic.



**Note** When creating or editing On Populate or On Validate BeanShell scripts, you must restart the client for the changes to take effect.

**Step 9** Click **OK**. The newly created argument is displayed in the User Input Arguments dialog box.

**Step 10** To change sequence in which arguments appear in the input form, select an argument and click **Move Up** or **Move Down**.

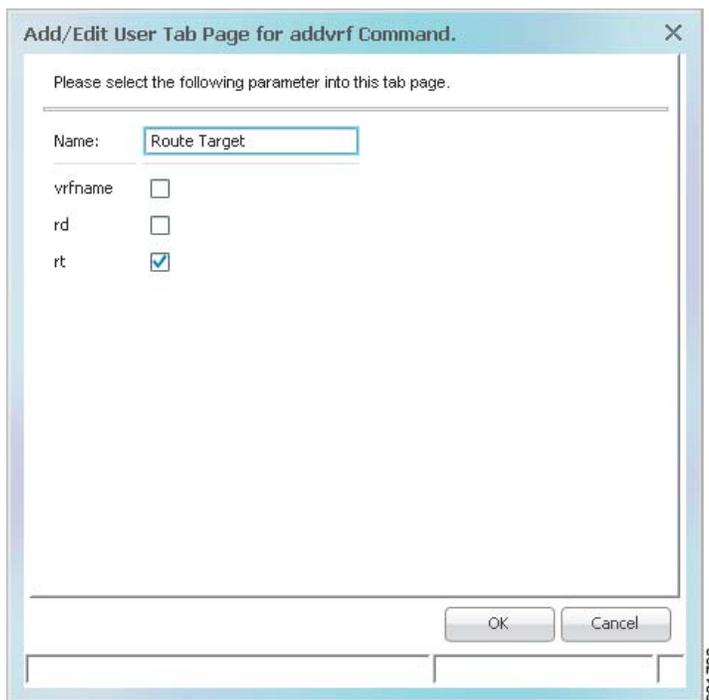
**Step 11** Click **Next**. The Tab Pages dialog box is displayed.

Use this feature if you want your input form to have different tabs. By default, all of the parameters are displayed in a single tab, General.

**Step 12** If you want to use tabs, do the following:

- Click **New**. The Add/Edit User Tab Page dialog box is displayed.

**Figure 4-8 Add/Edit User Tab Page for Command Dialog Box**



- Enter a name for the Tab and select the required parameter.

You can define a maximum of 20 tabs for a command. The name of the tab can contain a maximum of 20 characters.

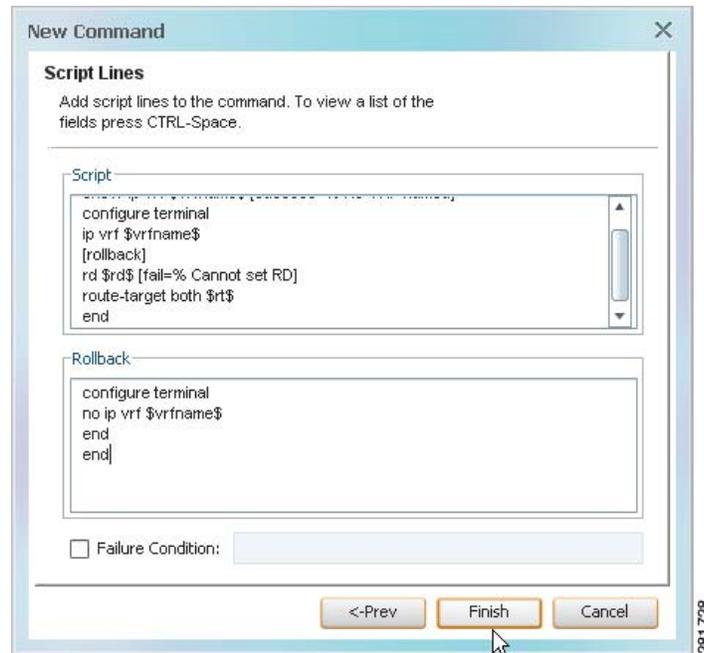
- Repeat the steps for additional tabs.

- Step 13** Click **Next** to define the command script, which can be either Prime Network Macro Language scripts or Beanshell scripts. For more information, see [Command Manager and Command Builder: Macro Language and Beanshell Reference, page C-1](#). The Script Lines dialog box is displayed, enabling you to add or edit a script line, as shown in [Figure 4-9](#).



**Note** An efficient way to do this is to create the script in a text editor, then cut and paste it into the New Command dialog box.

**Figure 4-9** Script Lines Dialog Box



- Step 14** Enter the required information, using the following guidelines:
- To view all user-defined and built-in parameters in the Command Builder application, position the cursor in the Script or Rollback field and press **Ctrl-Spacebar**. A dialog box is displayed that lists all available arguments (containing both the user-defined input argument and the built-in properties of the IMO context). Select an entry from the list and then click **OK** to add it to the Script or Rollback field.
  - Pragmas are enclosed with square brackets: [...].
  - It is possible to use multiple pragmas in a single line, in which case all pragmas are analyzed. If the same type of pragma is repeated, only the last one is used.
  - If carriage returns are required in the command line, enter the escape sequence **&cr**.

Field	Description
Script	The actual Telnet script lines sent to the NE. The script lines can contain optional inline directives (pragmas) for finer granularity control. For more information about the supported pragmas, see <a href="#">Supported Pragmas, page C-4</a> .
Rollback	(Optional) The rollback script that is used when the command fails. <b>Note</b> If the rollback script fails, no additional actions are performed.
Failure Condition	(Optional) A general failure condition that applies to all script lines. To specify a failure condition: <ol style="list-style-type: none"> <li>1. Check the Failure Condition check box.</li> <li>2. In the Failure Condition field, enter the text that is to be looked for during script execution. If the specified text appears in the reply, the command is aborted.</li> </ol>

**Step 15** Click **Finish**. The Create Command dialog box is displayed.

A LED indicates the progress or status of the command as it is being saved to the registry:

- Blue—The command definition is being saved.
- Green—The command has been created or updated successfully.
- Red—Command Builder failed to create or update the command.

**Step 16** Click **Close** when the command has been successfully saved. The newly created command is displayed in the Command Builder table.

See [Testing the Command Locally Before Deploying It to Other NEs, page 4-35](#) to test the command.

See [Publishing Commands to Other NEs, page 4-35](#) to publish the command.



**Note** If you plan to publish the command to multiple network elements so you can work on commands in bulk, be sure to delete your local copy of the command after it is published. Otherwise, when you choose multiple managed elements, the command will not be listed.

## Defining a Combo Field Type for a Command Builder Command

If you want to use a combo box in your input form, choose **Combo** in the Add/Edit User Argument dialog box ([Step 8](#) of the previous procedure), a Browse button is enabled. Use it to create a selection list (drop-down list) of valid options that will be displayed.

This example creates a combo box with Up and Down choices.

**Step 1** Select **Combo** in the Type field of the Add/Edit User Argument for x Command dialog box.

**Step 2** Click **Browse**. The Selection List dialog box is displayed.

**Step 3** Enter the required information:

Field	Description
Value	A value that will be associated with the option; for example, <b>1</b> .
Label	The description of the entry that is displayed in the selection list (drop-down list) of the input form, such as <b>Up</b> .

**Step 4** Click **Add**.

**Step 5** Repeat Steps 3 and 4 as needed until you have added all entries (such as **2** and **Down**).

**Step 6** Click **Close**.

## End-to-End Example of Creating a Macro Language Command

This topic provides an example of how to create the command, **addvrf**, using Prime Network Macro Language.

- 
- Step 1** Select the element that you want to configure this command for in the Prime Network Vision tree pane, context pane, or Inventory window.
- Step 2** Right-click the element, then choose **Management > Command Builder**. The Command Builder wizard is displayed.
- Step 3** Click **New**. The New Command dialog box is displayed.
- Step 4** Define the command identification information:
- Name—**addvrf**
  - Caption—**Add VRF**
  - Menu Visible—**Checked**
  - Menu Path—**VRF Commands**
  - Context IMO—**Automatically displayed**
  - Timeout—**180000**
  - Language—**Prime Network Macro**
  - Protocol—**Telnet** is selected by default
- Step 5** Click **Next**. The Command Authorizations dialog box is displayed.
- Step 6** Select the security access role **Administrator**.
- Step 7** Click **Next**. The User Input Arguments dialog box is displayed.
- Step 8** Click **New**. The Add/Edit User Argument for **addvrf** Command dialog box is displayed.

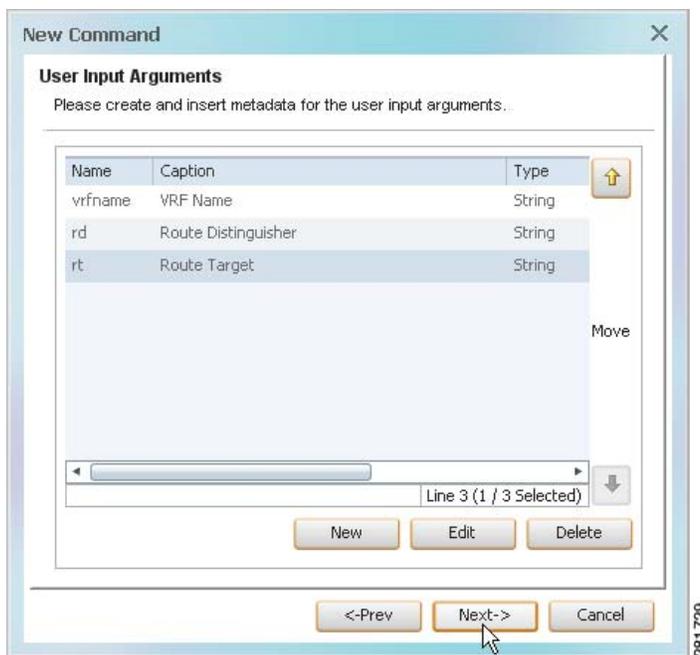
**Step 9** Using [Table 4-3](#), enter the information for the parameters.

**Table 4-3** *addvrf Command Parameters*

Field	Parameter 1	Parameter 2	Parameter 3
Name	vrfname	rd	rt
Caption	VRF Name	Route Distinguisher	Route Target Both
Type	String	String	String
Width	15	15	15
Visible	Select	Select	Select
Tooltip	Enter VRF name	Enter Route	Enter Target
Required	Checked	Unchecked	Checked

**Step 10** Click **OK** in the Add/Edit User Argument dialog box after each parameter. The new user-defined input parameters are displayed in the User Input Arguments dialog box ([Figure 4-10](#)).

**Figure 4-10** *User Input Arguments Dialog Box*



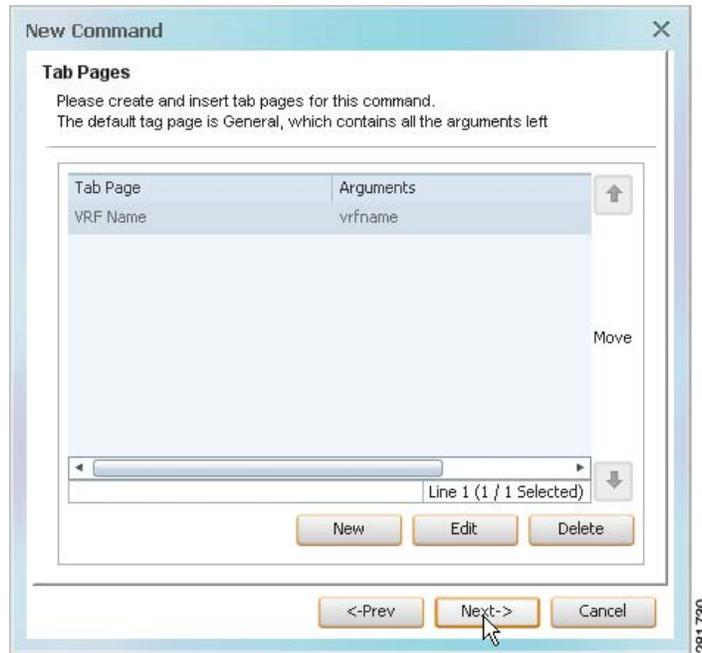
**Step 11** Click **Next**. The Tab Pages dialog box is displayed.

**Step 12** Click **New**. The Add/Edit User Tab Page dialog box is displayed.

**Step 13** Enter a VRF Name and select **vrfname**.

**Step 14** Click **OK**. The newly added tab pages are displayed in Tab Pages dialog box.

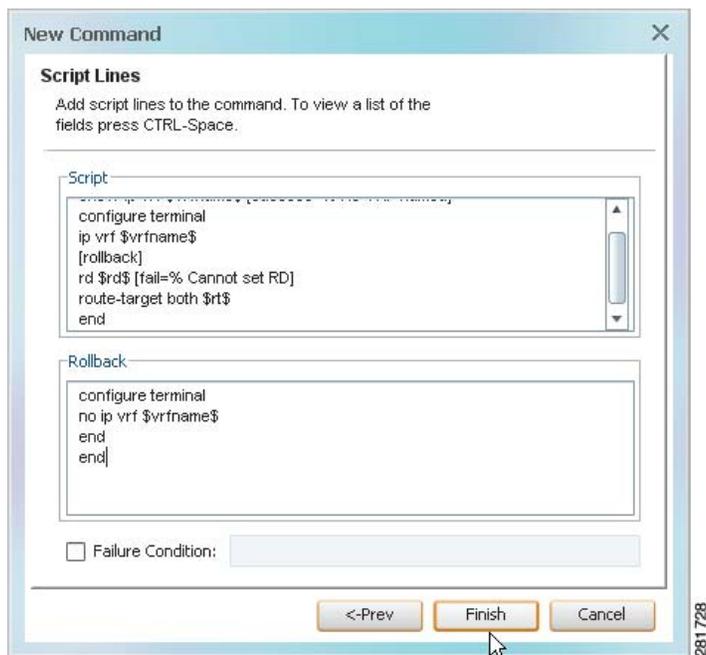
**Figure 4-11** Tab Pages Dialog Box



**Step 15** Click **Next**. The Script Lines dialog box is displayed, enabling you to add the script lines.

**Step 16** Add the script lines, as shown in [Figure 4-12](#).

Figure 4-12 Script Lines Example



- Step 17** Click **Finish**. The Create Command dialog box is displayed. LEDs indicate the progress or status of the command as it is being saved to the registry.
- Step 18** Click **Close** when the command is complete. The newly created **addvrf** command is displayed in the Command Builder table.
- Step 19** Select the **addvrf** command in the Command Builder table.
- Step 20** Run the command in one of the following ways:
- Click **Run Command** in the toolbar.
  - Choose **Tools > Run Command**.
  - Right-click the command, then choose **Run Command**.

The input form is generated and displayed. The Route Distinguisher and Route Target are displayed in General tab and vrfname is displayed in a separate tab called VRF Name.

In the General tab, the Route Target is a mandatory field (bold font) and Route Distinguisher is an optional field.

- Step 21** Enter values in the VRF Name, Route Distinguisher, and Route Target fields.
- Step 22** Click **Preview** to see how the command, including variables, looks before it is executed.
- Step 23** Click **Execute Now** to view the results of the **addvrf** command as it is being executed.
- Step 24** Close the input form to return to the Command Builder window.

## Testing and Publishing Commands

The following topics describe how to test and publish commands:

- [Testing the Command Locally Before Deploying It to Other NEs, page 4-35](#)
- [Publishing Commands to Other NEs, page 4-35](#)

### Testing the Command Locally Before Deploying It to Other NEs

Before you publish a command and apply it to other NEs, test it on your local NE using the following procedure. Once you have done this, you can apply it manually to multiple managed NEs, or publish it and make it available to a larger group of NEs (by family, NE type, or software version).

- 
- Step 1** From the Command Builder window, right-click the command and choose **Run Command**. Command Builder generates your input form.
  - Step 2** Complete the input form and click **Preview**.
  - Step 3** Click the **Result** tab to verify the command and variables before executing the command on the NE.
  - Step 4** If the command is successful and you want to execute it on the local NE, click **Execute Now**.



**Note** You might be prompted to enter your device access credentials. Once you have entered them, these credentials will be used for every subsequent execution of a command in the same GUI client session. If you want to change the credentials, click **Edit Credentials**. The Edit Credentials button will be disabled if the command is scheduled to run at a later time or if it is an SNMP command.

---

The queued commands can be terminated, if required, by clicking **Abort**.

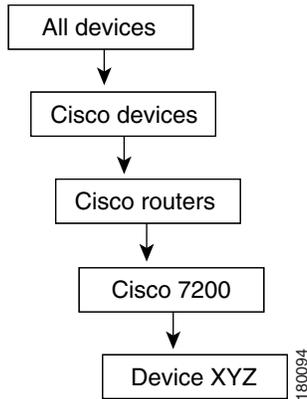
Alternatively, if you are authorized by the Administrator to schedule jobs, you can use the Scheduling tab to schedule the command to run at a later time. You can check the job results by selecting **Tools > Scheduled Jobs** from the Prime Network Vision Tools menu.

---

Every command that is executed is logged in the Prime Network event database. The command's execution history can be viewed using Prime Network Events. For more information about Prime Network Events, see the [Cisco Prime Network 5.1 User Guide](#).

### Publishing Commands to Other NEs

Publishing is the process of applying a command to a wider scope of managed NEs, so that the other NEs can use the new command. The Command Builder Hierarchy Manager dialog box controls the publishing of commands across the *inheritance hierarchy*. [Figure 4-13](#) shows an example of an inheritance hierarchy. In this example, the top level of the hierarchy is All devices and the lowest level of the hierarchy is Device XYZ.

**Figure 4-13 Inheritance Hierarchy Example**

When a command is published to a node in the hierarchy, it overrides any inherited command from a higher level and automatically applies to all its children. For example, if a command is published to Cisco 7200, it overrides any variant of this command it inherited from a higher level and is assigned to all NEs of type Cisco 7200 in the system. NEs above the Cisco 7200 class are not affected.

**Note**

A user with a Configurator role can add and publish commands on all NEs regardless of their assigned device scopes.

To publish a command:

**Step 1** From the Command Builder window, right-click the command and choose **Hierarchy Manager**.

**Note**

The names displayed in the VNE Hierarchy Location column depends on the registry settings. It is determined by the VNE schema (which controls the NEs that are managed and modeled by Prime Network). For example, a user-friendly name is displayed if a hierarchy path has been defined in the registry.

Rows are displayed in descending order with the top row representing the highest level of the hierarchy and the bottom row representing the lowest level of the hierarchy. The following information is provided for each level:

- **Exist**—If checked, indicates that a local variant of the command exists for that VNE hierarchy location.
- **Registry Key**—The hierarchy path as defined in the registry.

Table 4-4 describes the tools that are displayed in the Hierarchy Manager dialog box.

**Table 4-4 Hierarchy Manager Dialog Box Tools**

Button	Description
	Copies the command from a selected node in the hierarchy so that it can be pasted onto another node in the hierarchy. A copy icon is displayed to the left of the selected node.
	Cuts the command from a selected node in the hierarchy so that it can be moved to another node in the hierarchy. A cut icon is displayed to the left of the selected node.

Table 4-4 Hierarchy Manager Dialog Box Tools (continued)

Button	Description
	Pastes the command that was copied or cut from a selected node in the hierarchy onto another node in the hierarchy. A paste icon is displayed to the left of the selected node.
	Deletes the command from the selected node in the hierarchy. <b>Note</b> If the command has been deleted from all nodes, it is removed from the list in the Command Builder window.
	Saves a full command definition to a file that can later be imported to another managed element. For more information, see <a href="#">Exporting Commands with Command Builder, page 4-38</a> .
	Performs one of the following actions, depending upon whether or not a version of the command definition already exists in Command Builder: <ul style="list-style-type: none"> <li>For new command definitions, imports a full command definition to a managed element. For more information, see <a href="#">Importing Commands with Command Builder, page 4-38</a>.</li> <li>For existing command definitions, replaces the existing command definition with an updated version of the command definition.</li> </ul>

**Step 2** Select the node in the hierarchy from where you want to Copy/Cut the command.

**Step 3** Click **Copy** or **Cut** in the toolbar to copy or cut the command.

**Step 4** Select the node in the hierarchy where you want to publish the command.

**Step 5** Click **Paste** on the toolbar to paste the command. The command is published to the selected node in the hierarchy.



**Note** If you plan to publish the command to multiple network elements so you can perform commands in bulk, be sure to delete your local copy of the command after it is published. Otherwise when you choose multiple NEs, the command will not be listed.

If you select a group of network elements but the command is not listed, it is likely due to one of the following reasons:

- The command was not published to at least one of the network elements. You must publish the command to that network element using the hierarchy manager.
- One of the network elements has a local copy of the command. You must delete the local copy of the command using the hierarchy manager.



**Note** If a network element has more than one command instance, the command instance that is executed will be the one in the lowest hierarchy node.

## Exporting Commands with Command Builder

When you export a command, a file containing the full command definition is created on the client machine. This is normally done when you want to manually import the command file to other NEs rather than use the Hierarchy Manager and the publishing procedure. The file is not copied to any other NEs; it is created so it can be imported by other NEs.

- 
- Step 1** In the Command Builder window, right-click the command and choose **Export Element**.
  - Step 2** Command Builder lists all versions of the command. In other words, if you have a 7206 NE with one version, but all other 7200 NEs have a different version, Command Builder would display both versions because both have been saved on your client machine. Check the version you want to export and click **OK**.
  - Step 3** Browse to the directory where you want to save the command.
  - Step 4** In the File name field, enter a name and click **Save**. The command is saved in XML format on the local client machine.
- 

## Importing Commands with Command Builder

Importing commands makes new commands available to an NE. For example, if you created a new command for NE 1, you could export the command from NE 1, and then import it to NE 2. The export is performed by launching Command Builder from NE 1, and the import is done by launching Command Builder from NE 2, as long as NE 2 is the same NE type.

If the import operation will overwrite a command that already exists, one of the following will happen:

- For a local command, Prime Network overwrites the existing definition.
- For a published command, Prime Network generates an error message which states that the command already exists. You must delete the existing command instance using the procedure described in [Deleting Commands, page 4-39](#).

You can import multiple commands using this procedure as long as all of the commands are associated with the same VNE.

To import commands:

- 
- Step 1** In the Command Builder window, choose **Tools > Import Element**.
  - Step 2** Browse to the directory that contains the commands that you want to import.
  - Step 3** Select the commands that you want to import. To select multiple commands, press **Shift** or **Ctrl** while choosing the commands.
  - Step 4** Click **Open**. The Import Elements dialog box is displayed.  
If you select multiple files, Command Builder presents an Import Element dialog box for each command.
  - Step 5** In the Import Element dialog box, select the VNE hierarchy location for the specified command.
  - Step 6** Click **OK**. The Command Builder window is displayed.
  - Step 7** Click **Close**. The commands are imported and displayed in the Command Builder window.
-

## Deleting Commands

The procedure for deleting commands depends on whether the command is local or has been published.

- Local commands can be deleted from the Command Builder window by right-clicking the command and choosing **Delete**.
- Published commands must first be deleted from the applicable nodes using the Hierarchy Manager (right-click the command, then choose **Hierarchy Manager**). You must delete all instances from all applicable nodes.

You can delete multiple commands at a time as long as all of the commands are associated with the same VNE. You may want to keep a local copy of the command before you delete it.

