

Managing Templates

Overview of Template Management

Use the Template Manager to create a new template for provisioning devices. When you click the right mouse button on a template, you are given the option to bring up the archive viewer, create a new data object for the template, change its lock state, or delete the template. When you click the right mouse button on a data object, you are given the option to open the **Template Config Builder** window or delete the data object.

A template is a configuration “boilerplate,” a form that contains the text and Cisco IOS commands that are common to a number of devices, plus placeholders for the data that vary from device to device.

Templates are constructed in two parts:

- **Template body**—holds the nonchanging text and tokens (called *attribute names*) that are replaced with data during generation of the configuration.
- **Template data**—lists the devices for which configurations are generated and defines the values that are substituted for the tokens in the template body.

Before you can define a template using the GUI developed by Cisco, the devices for which you want to generate configurations must already exist in the database. Use the **Element Manager** window to add devices. For information about using the **Element Manager** window, see the chapter “Managing Network Elements.”

When the devices have been added to the database, click the **Template Manager** button on the launch pane to open the **Template Manager** window, where you can open a **Template Config Builder** window (described later in this chapter).

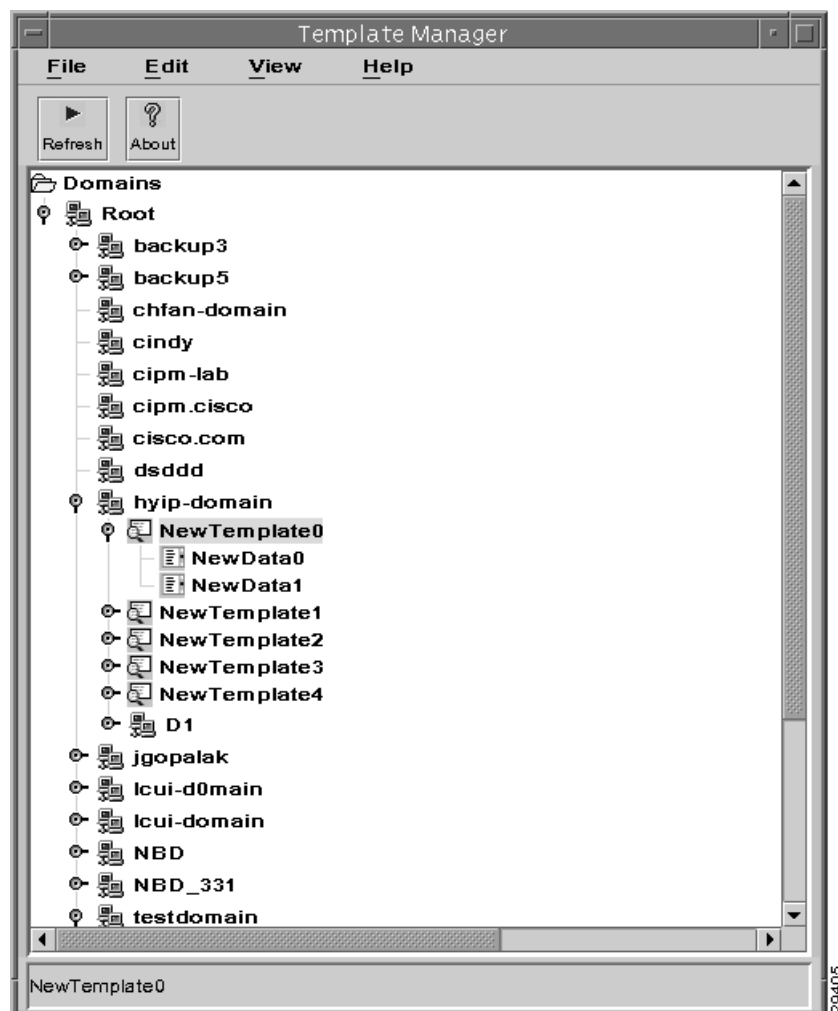
Both parts of the template are constructed in the same **Template Config Builder** window. You also use this window to preview device configurations and commit configurations to the database (as *working configs*).

However, to download the configurations to devices, you must use the **Element Manager** window, as described in the sections “Downloading to a Device” and “Downloading to Multiple Devices” in Chapter 5, “Managing Network Elements.”

Template Manager Window

To open the **Template Manager** window, click on **Template Manager** on the Main Window. This window is nearly identical in appearance to the **Element Manager** window and works in a similar way. The menu structure of the two windows is the same. For a description of the menu commands that are available in the **Template Manager** window, see the section “Element Manager Window” in Chapter 5, “Managing Network Elements.”

Figure 6-1 Template Manager window



The **Template Manager** window displays a data tree that shows all of the domains to which you have been given access. Unlike the **Element Manager** data structure, however, the tree displayed in the **Template Manager** window shows templates and template data objects, not network elements.

To create, open, modify, or delete templates and template data objects, use the floating options menu (opened by right-clicking the mouse).

To create a new template, first expand the elements in the tree to display the domain in which you want to locate your template, and then right-click on that domain's name. From the floating menu that opens, select **New Template**, give the template a name in the dialog that opens, and click the **OK** button (or click the **Cancel** button to close the dialog without creating a template).

See the section "Name Restrictions" in Chapter 5, "Managing Network Elements" for limitations on characters in names.

If you do not see the template in the data tree immediately beneath the domain to which it belongs, click on the ball to the left of the domain icon (or double-click on the icon itself) to expand the domain listing in the data tree.

Before you can do anything with your new template, you must create a template data object, which contains the values you assign to the template attributes. (Deleting this data item from the **Template Manager** window removes only the data values from the template. Body text and attribute names are stored in the template object.) Right-click the mouse on the name of your newly created template and select the **New Data** command from the menu that opens. Enter a name for this data object and click the **OK** button (or click the **Cancel** button to close the dialog without creating a data object). If you do not see the data object in the data tree immediately beneath the template to which it belongs, click on the ball to the left of the domain icon (or double-click on the icon itself) to expand the domain listing in the data tree.

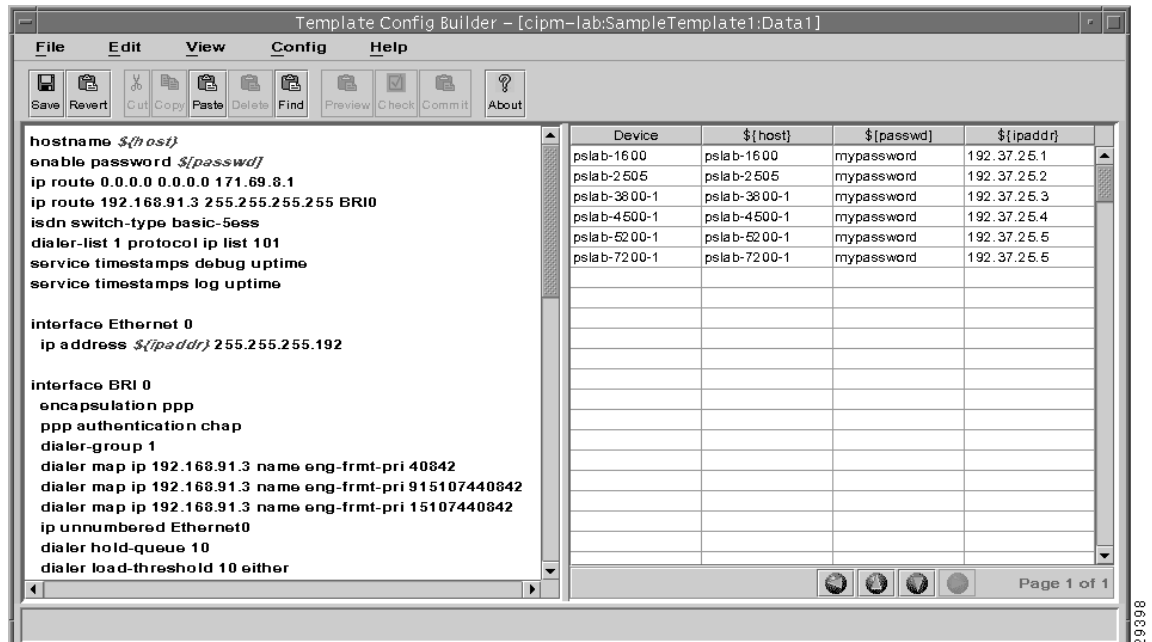
Note You can have multiple data objects associated with a single template, but any changes you make to the template body text affect all of the data objects that are associated with the template, regardless of which data file you used to open the **Template Config Builder**.

Template Config Builder

Right-click on a data object and select the **Config Builder** command from the floating menu.

The **Template Config Builder** window opens. This is where you create the template body, identify the devices for which you want to generate configuration files, and define the values that are used for individual devices.

Figure 6-2 Template Config Builder window



The template body is entered in the editable text area that takes up the left half of the **Template Config Builder** window.

Devices to be configured are listed in the pane that occupies the right half of the window. This is also where you enter the data that varies from device to device. A column appears to be correspond with each variable construct in the left pane.

Template Body

The template body contains the text that does not change from device to device, plus placeholders for the data that change for each device.

Enter the text for the body in the editable text box that occupies the left half of the **Template Config Builder** window.

Importing Text

To import the template body from some other source—either a router configuration file or a text-only file created in some other application—select the **Import** command from the **File** menu in the **Template Config Builder** window. Choose the **Template Body** command on the submenu. Respond to an alert that warns that the imported template may not match existing data.

When you have located the file in the **Import Template Body** window that opens, click the **OK** button. The selected file is imported into the **Template Config Builder**.

Attribute Names

Placeholders, or *attribute names*, must adhere to the following syntax:

- Attribute names are prefixed with a dollar sign (\$) and enclosed in braces if mandatory, square brackets if optional, or angle brackets if global:

```
${mandatory_attribute}  
$[optional_attribute]  
$<global_attribute>
```

- Banner text can include attributes, but may cause problems if it contains the following characters:

```
# >
```

For more about using these characters, refer to “Uploading from a Device” in Chapter 5, “Managing Network Elements.”

- Names within the braces or brackets must start with a letter of the alphabet (either upper or lower case), followed by any printable characters.
- A mandatory or optional attribute name can be used only once in a template body (global attributes can be used as many times as necessary).
- An attribute name (which includes the brackets and dollar sign) should contain no white space.

Here is a sample template body:

```
hostname ${host_name}  
router ospf ${process_id}  
redistribute static metric ${metric_val}  
network ${ip_subnet} ${ip_mask} area ${area_id}
```

When you have entered the template body text, save the template. This causes all attribute names (anything preceded by a *nd enclosed in braces or brackets) to be displayed as blue text.

It also causes numbered pages in the right-hand pane (the template data region of the **Template Config Builder** window) to be divided into as many columns as there are mandatory and optional attributes, plus one (for the device name). Each column is headed by a different attribute name.

The number of attributes that a single template can hold is dependent on both the capacity of the Java Runtime Environment and your system's resources. You should be able to add several hundred attributes, but more than 1000 attributes may cause excessive delays in calculating the spreadsheet. For large numbers of attributes (an access list, for example), consider using subtemplates.

The global page (which precedes page 1) is divided into as many columns as there are global attributes (plus one for the device name).

The first column heading, on both numbered and global pages is **Device**.

Subtemplates

You can reference another template (a *subtemplate*) from within a template by using the reserved word **#include**, followed by the name of the domain to which the subtemplate is attached in the **Template Manager** window, a colon, and the subtemplate's template name all in quotes (except the **#include** command):

```
#include "domainName:templateName"
```

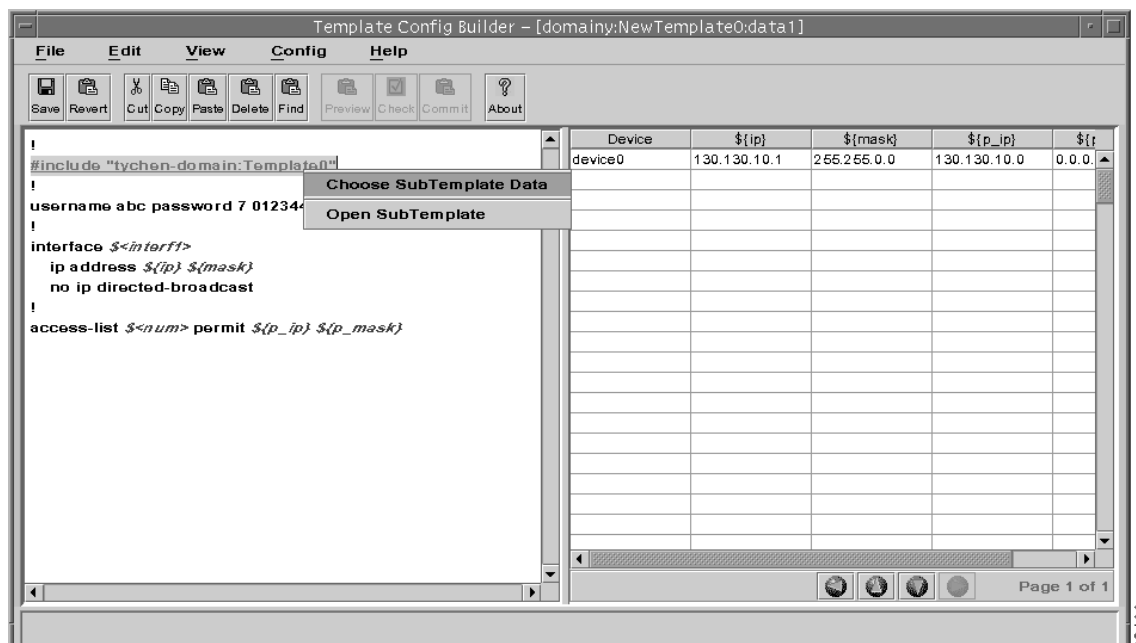
There can be no white space between the names, and no embedded carriage returns. However, there *must* be a space between **#include** and the first quote mark. See the section “Name Restrictions” in Chapter 5, “Managing Network Elements” for limitations on characters in domain names.

A subtemplate must be in the same domain, or an ancestor of the domain, in which the parent template is located in the **Template Manager** window. Subtemplate references must be in a straight line up the data tree. (You cannot refer to a template that is in a subdomain of an ancestor domain, for example.)

Subtemplates cannot reference other subtemplates.

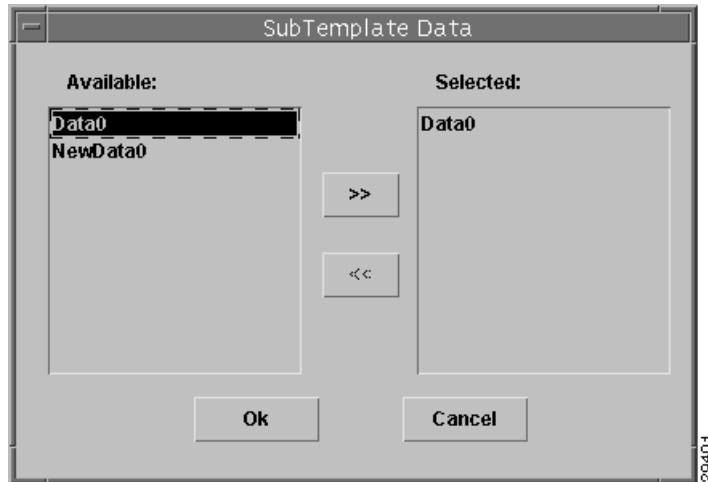
You must associate a data object with each subtemplate, by right-clicking the **#include** line. Select the **Choose SubTemplate Data** command on the floating **Options** menu.

Figure 6-3 Template Config Builder: choosing subtemplate



Then select the data object in the **SubTemplate Data** window that opens.

Figure 6-4 **Template Config Builder: SubTemplate Data window**

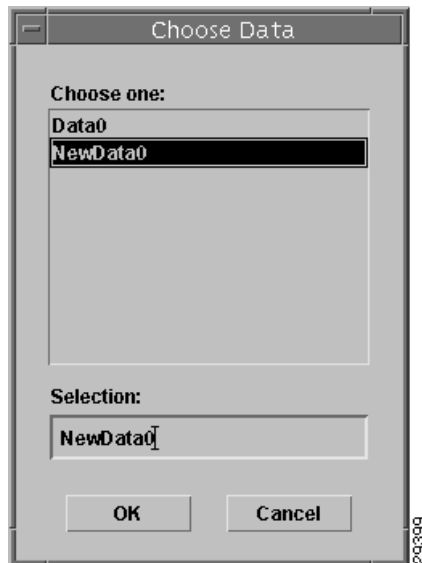


Use the center arrow buttons to move selected data from the **Available** window to the **Selected** window (or back again).

The association is stored in the database and is preserved the next time you open the **Template Config Builder**.

The **Open SubTemplate** command on the floating **Options** menu (displayed when you right-click on the **#include** line) opens the **Choose Data** window.

Figure 6-5 **Template Config Builder: Choose Data window**



When you select a data object and click the **OK** button, a new **Template Config Builder** window opens. The body text is that of the subtemplate referenced by the **#include** statement. The data is what is contained in the selected data object.

Conditional Inclusion of Template Body

A block of template body can be conditionally included. Conditional inclusion is specified in the template body text by the reserved words **#if**, **#elseif** and **#else** as follows:

```
#if {test1} {
    body1
#} elseif {test2} {
    body2
#} elseif {test3} {
    .
    .
    .
#} else {
    bodyN
#}
```

The **#if** command evaluates *test1* as an expression. The value of the expression must be a boolean value (either numeric, where 0 is false and anything else is true, or a string such as `true` or `yes` for true and `false` or `no` for false). If the expression is true, then *body1* is included in the template. If the *test1* expression is false, then *test2* is evaluated as an expression and, if it is true, then *body2* is included, and so on. If none of the expressions is true then *bodyN* is included. There can be any number of **#elseif** clauses, including zero. The *bodyN* option may also be omitted, as long as **#else** is omitted, too.

An expression consists of a combination of operands, operators, and parentheses. White space may be used between the operands, operators and parentheses. Operands may be specified in any of the following ways:

- As a numeric value, either integer or floating-point.
- As a template global variable. (A non-global variable cannot be accepted here because it may contain more than one value.)
- As a string enclosed in double-quotes.

The valid operators are listed below, grouped in decreasing order of precedence:

Boolean:

<	<i>less than</i>
>	<i>greater than</i>
<=	<i>less than or equal</i>
>=	<i>greater than or equal</i>

Each operator produces 1 if the condition is true, 0 otherwise. These operators may be applied to strings as well as numeric operands, in which case string comparison is used.

Boolean:

==	<i>equal</i>
!=	<i>not equal.</i>

Each operator produces a zero/one result.

&&	Logical AND. Produces 1 if both operands are non-zero, otherwise 0.
----	---

|| Logical OR. Produces 0 if both operands are zero, otherwise 1.

The following is an example of conditional inclusion in a template:

```
hostname myhost
interface $<interf1>
#if {$<interf1> == "s1"} {
    bandwidth 56000
#} elseif {$<interf1> == "e1"} {
    bandwidth 10000000
#} else {
    bandwidth ${default}
#}
```

The syntax of conditional inclusion strictly follows Tcl **if** commands and general Tcl expression syntax. CIPM utilizes an embedded Tcl interpreter to parse, filter and process these conditional control commands and expressions.

Archiving

To view archive information about a template and to perform operations on stored configurations, right-click on the template and select the **Archive Viewer** option. For more about the operations you can perform, refer to “Archive Viewer” in Chapter 8, “Archive Administration.”

When you check in a template, the Element Manager creates a new version to store in the archive; you can view each version, compare versions, and so on, as described in “Template Archive Viewer” in Chapter 8, “Archive Administration.”

Template Data

Once you have saved your template, the right side of the **Template Config Builder** displays columns with attribute names for titles. These columns hold the data to be used for individual devices.

Each row configures a different device. You can have a maximum of 200 rows in a data object (but a single template body can be associated with multiple data objects). Figure 6-2 shows the **Template Config Builder** window, with template body and data.

List each network element in the **Device** column by its name as defined in the **Element Manager** window. This name is an arbitrary value that you define within the Cisco IP Manager software; it may or may not be the device’s actual host name.

You can add any element in any domain to which you have access, using the following syntax:

```
domain:element
```

There can be no spaces surrounding the colon. Do not specify more than one domain (or subdomain). The entry `domainA:subdomainB:element` would be interpreted as a device named `subdomainB:element` in the domain `domainA`. The device named `element` resides in `subdomainB`; there is no device named `subdomainB:element`. Specifying `subdomainB:element` is sufficient, even though they are nested in `domainA`. (Domain names must be unique throughout the system.)

You can copy network elements from the **Element Manager** window, then paste them into your template. Select the devices to be copied and use the **Copy to Clipboard** command on the **Edit** menu. Return to the **Template Config Builder** window, click in the first open cell in the **Device** column, and choose **Paste** from the **Edit** menu.

Data Pane Is Three Dimensional

The data portion of the **Template Config Builder** window has three dimensions:

- **Rows**—Each row in the data window represents a different device. Use the vertical scrollbar on the right side of the pane to display additional rows.
- **Columns**—Each column (other than the first) represents a different attribute name in the template body. Use the horizontal scrollbar at the bottom of the grid to display additional columns.
- **Pages**—Data for each iteration of a Cisco IOS configuration line is stored on a separate page. Page numbers are displayed in the lower right corner of the window. Each time you open the **Template Config Builder** window, page 1 is displayed. Use the arrow buttons (green diamonds in black circles) at the bottom of the data pane to move forward and back through the pages. The left arrow takes you to the global page, which precedes page 1, and the right arrow takes you to the last pages; the up/down arrows move forward and back one page at a time. When the last page is displayed, clicking the down arrow adds a new, blank page.

For each mandatory or optional attribute name in the template body, enter a data value in the appropriate cell on page 1 in the data pane. Enter a data value for each unique global attribute, on the global page.

Global attributes defined in the parent template can be referenced in the body text of subtemplates. If no data value is assigned to the attribute on the global page in the subtemplate, the value assigned in the parent template is used.

If you want the configuration file to contain multiple iterations of a line that is specified once in the template body, enter one complete set of data—one value for each of the attribute names on that line—for each iteration, with each data set on a different numbered data page. Do not repeat data for non-iterative lines.

To repeat the **network...** line contained in the body of the template shown previously, for example, add one set of data for the attribute names `$(ip_subnet)`, `$(ip_mask)`, and `$(area_id)` for each repetition desired, one set per page. The first set could be included on the first page along with the data for **hostname** and the other lines that won't be repeated. But data for these *nonrecurring* lines should appear only once, on the first page.

Use the page forward button at the bottom of the window to advance to the next page. There can be an infinite number of pages.

Note While it is possible to skip over blank pages and continue to add data, this consumes memory unnecessarily and performance suffers. To remove blank pages, save the template and then force the **Template Config Builder** to reload it by selecting the **Revert** command from the **File** menu (or close and reopen the **Template Config Builder**).

Importing Data

To import template data from a text file, select the **Import** command on the **File** menu in the **Template Config Builder** window and choose the **Template Data** command on the submenu.

Data should be saved in a text file, one device per line. To supply values for fields, separate them by tabs or commas. (Do not mix separators within a file—use one or the other, or data will be lost.)

You must first create the columns in the data portion of the **Template Config Builder** window by saving the template before importing the data. If your data file specifies a value for a column that does not exist in the **Template Config Builder** window, that value is ignored.

Multiple data pages cannot be specified from a single text file.

It is possible to import data to one page from one text file, then advance to another page and import data from a second text file. However, data is always imported into the page beginning at the top left cell, regardless of device specification, and device names in the second text file are ignored.

For example, if you create a template that has data fields `${A}` and `${B}` in addition to the device column and you import the following data from a file:

```
deviceOne,dataA-1,dataB-1
deviceTwo,dataA-2,dataB-2
deviceThree,dataA-3,dataB-3
```

...then advance to page 2 and import the following text from a second file:

```
deviceTwo,dataA-2.2,dataB-2.2
```

...the data in the second file is imported into the first row of page two and the device specification is ignored. This assigns the data to `deviceOne`, not `deviceTwo`, even though the **Template Config Builder** window shows `deviceTwo` on the screen. When you exit this page and return, there is no data assigned to `deviceTwo` on this second page.

To make the data match the device, you should duplicate the device names from the first file, even if some of the devices have no second-page data.

The example would work if the second file contained the following lines:

```
deviceOne
deviceTwo,dataA-2.2,dataB-2.2
deviceThree
```

Missing Values

If a configuration line in the template body contains multiple mandatory attribute names and you provide data for some but not all of them, an error message is sent to the message console: "Some attribute values are missing in the spreadsheet." The configuration is not generated.

If no values are given for *any* of the attribute names on a template body line, the corresponding line in the configuration file is not generated.

If there is no value assigned to a global attribute, the configuration not generated and an error message is sent to the message console. (However, see the discussion of subtemplates.)

If there is no value in the template data object for an *optional* attribute, an empty string is generated for that value.

If you change the name of an attribute, data associated with the attribute is lost when you save the template. You should copy the entire data column for that attribute name before you change the attribute's name. After saving the edited template, paste the data back into the column.

Previewing Configurations

When you have entered data for a device, you can preview the configuration that is generated. On page 1 in the data pane, select one or more devices and choose the **Preview** command from the **Config** menu. (Templates must be saved before you can preview the configuration; if you have not yet saved the template, a dialog asks if you want to. When you save the template, you must again select the device in the data pane and choose the **Preview** command.)

A preview window is opened for each selected device, displaying the text of the configuration that will be generated. You can export the text from any of these windows by right-clicking anywhere inside the window and selecting one of the commands from the floating menu that opens. The **Save All** command saves all text in the preview window to an ASCII file; the **Save** command saves only the text that you selected (prior to opening the options menu). The **Find** command finds the next instance of text you specify. The text to find must appear below the current cursor position; when the find mechanism finds the last instance of specified text, that text remains highlighted. The **Copy** command copies text into the clipboard; you can then paste that text anywhere else that permits such an operation (such as into the left pane of the **Template Config Builder** window, or into any other window that has a **Paste** button).

When you close the preview window, you must then choose the **Commit** command on the **Config** menu to generate a *working config* that is retained in the database. If you do not commit the configuration, an **Object not found** error message is generated when you choose an operation that requires the device's working configuration. Select multiple devices to generate and commit configurations for a group of network elements.

Finding Data and Body Text

You can use the **Find** dialog to search for text in either the body or the data portion of your template. First click the mouse in either the body or data side of the **Template Config Builder** window, then select the **Find** command on the **Edit** menu. Enter the text you want to locate in the **Find What** field, select the **Match Case** checkbox if you want the search to be case sensitive, then choose the **Find Next** button. When text is located in the data pane, the corresponding attribute in the body pane is highlighted.

Changing Variables

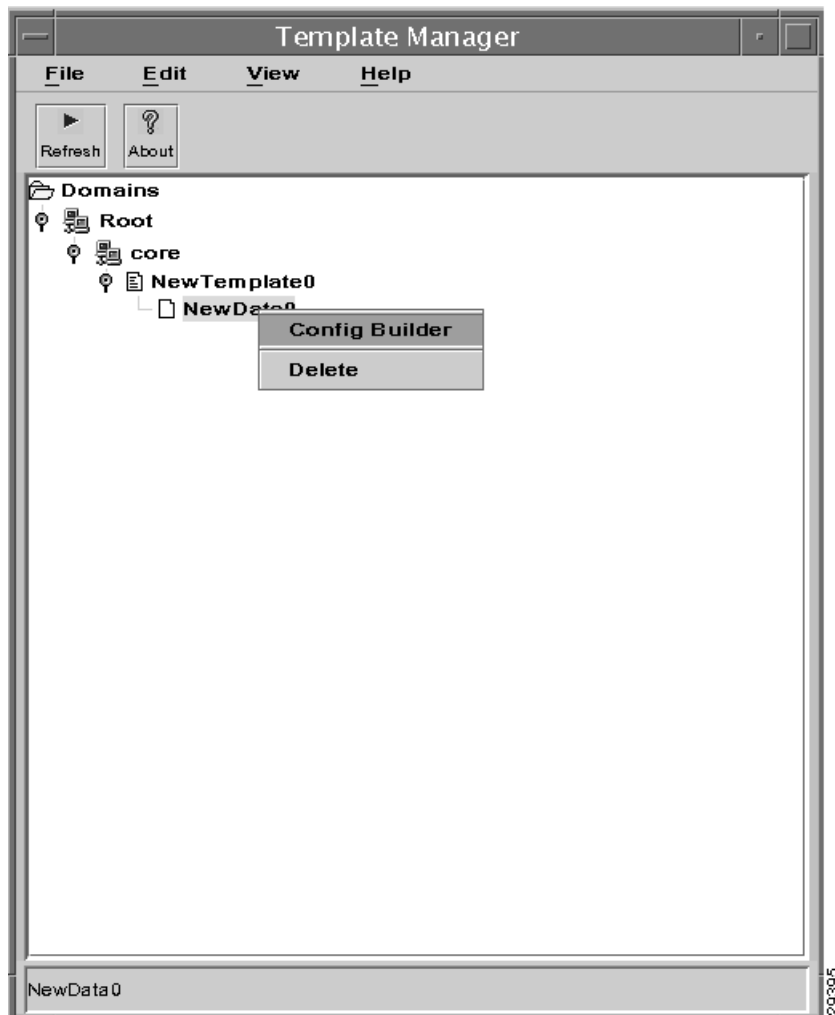
You can change a variable in the configuration template builder in two ways:

- By choosing **Rename Variable** from the **Edit** menu
- By changing the variable with the template configuration editor

If you change a variable with the **Rename Variable** command, its value is retained. However, if you change the variable with the template editor, the value is lost. An example explains this.

- 1 Open the template manager.
- 2 Create a template and a data object under it.
- 3 Right-click on the data object, and select **Config Builder**.

The preceding three steps are shown in Figure 6-6.

Figure 6-6 **Selecting Config Builder**

- 4 Enter the following in the template editor pane.

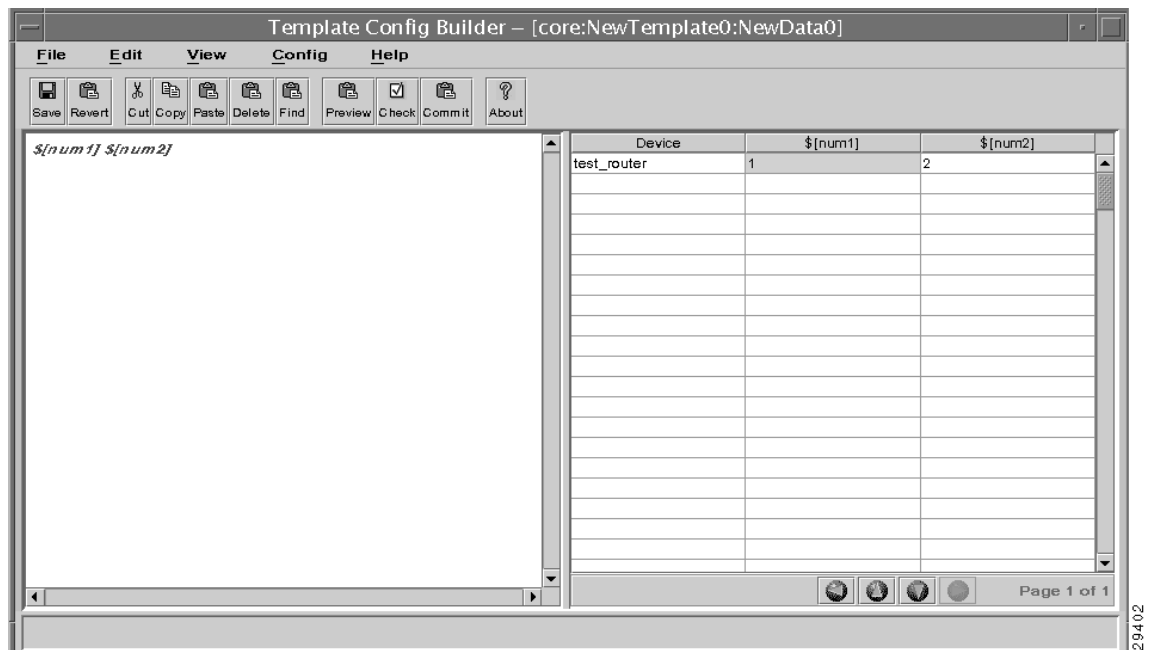
```
$[num1] $[num2]
```

- 5 Click **Save**.

Add arbitrary data to the columns. For example, in the column with the heading `$(num1)`, enter a 1 and in the column with the heading `$(num2)`, enter a 2.

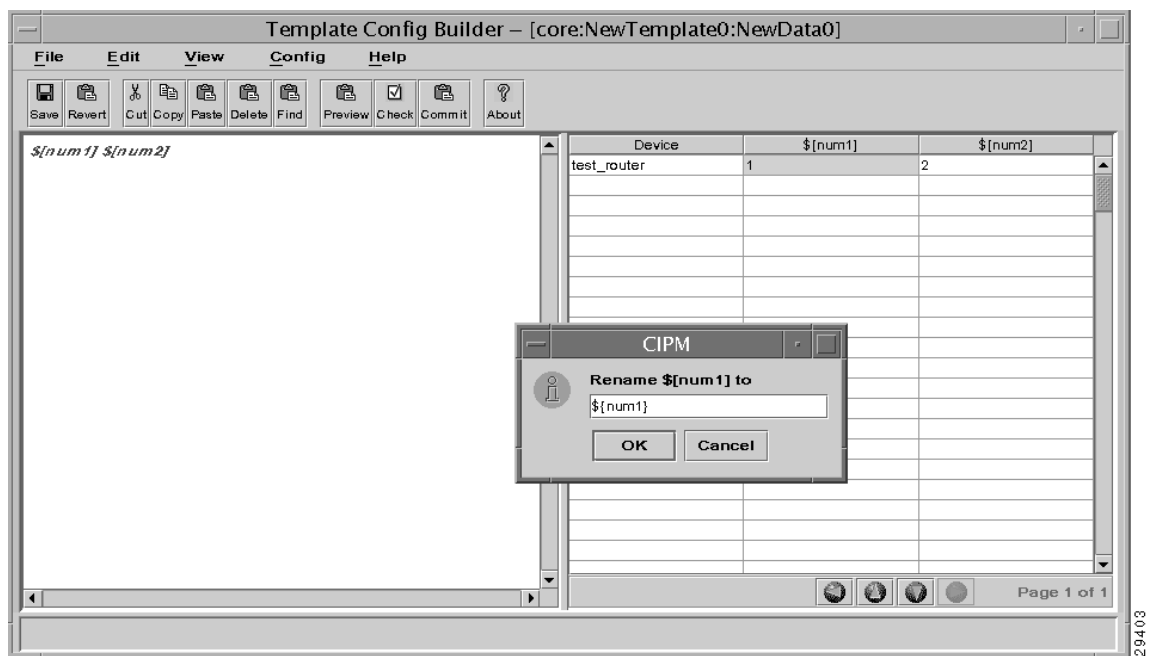
- 6 Click on save. The screen has the appearance shown in Figure 6-7.

Figure 6-7 Template Config Builder



7 Change the value `$(num1)` to `num1` by choosing **Rename Variable** from the **Edit** menu.

Figure 6-8 **Template Config Builder with Rename Variable Dialog**

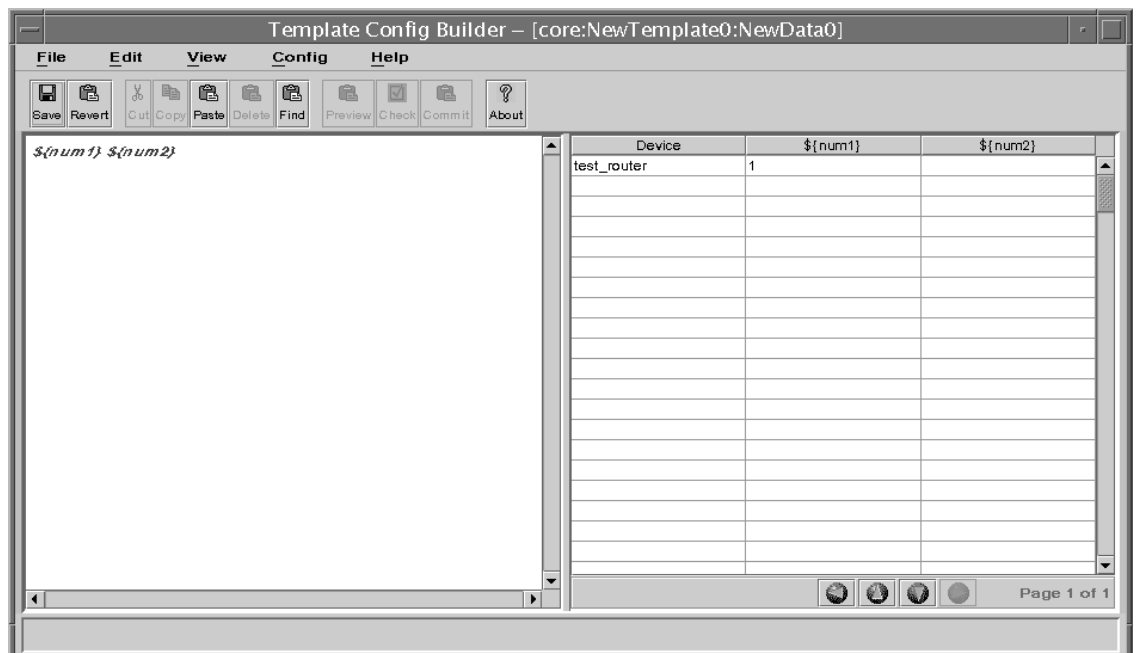


Click **Save**, and notice that the 1 remains in the column.

8 Now change the value of `§ [num2]` to `§ {num2}` by editing it in the template configuration editor (the left pane).

Click **Save**, and notice that the 2 disappears from the column.

Figure 6-9 **Template Config Builder after Editing a Variable**



Generating Configurations

When you are satisfied that the template will generate the configurations you want, select the devices you want to configure and choose the **Commit** command from the **Config** menu.

This generates configurations using the template body and data, and the configurations are written to the database as working configs.

To download configurations to devices, use the **Element Manager** window. Select one or multiple devices, right-click the mouse, and choose the **Download to Elements** command.