



Using the Template Manager

The Template Manager helps you manage the configuration templates, that is, boilerplate forms, that contain text, Cisco IOS commands, and variables for provisioning devices. Templates can be either predefined system templates provided by Cisco, or user-defined templates. User-defined templates are modified system templates or new templates not based on an existing system template.

You can use the Template Manager to do the following:

- modify predefined system templates
- create new templates
- delete templates
- generate configurations.

Use the Network Inventory Manager to add devices and to download the configurations to devices (provision devices).

For information about using the Network Inventory Manager to add and provision devices, see [Chapter 3, “Using the Network Inventory Manager.”](#)

Accessing Default Templates

The Template Manager provides several default templates for T1 CPE devices. These default templates are stored at the following location:

`$CCNSC_HOME/common/templates/IAD2421`



Note

The `$CCNSC_HOME` environment variable contains the directory the CCNSC Subscriber Provisioning product was installed. The default is `/opt/CSCOcnscs`.

If while using the Template Manager, you corrupt the default templates, you can import the default templates from the above location. You can import the templates using the Template Config Builder.



Note

A few templates at these locations are not automatically downloaded, but can be downloaded manually. However, these templates have not been tested and may have unexpected results.

Understanding the Template Manager Window

The CCNSC Subscriber Provisioning Template Manager window displays a hierarchical view of the available templates. When you first bring up the Template Manager, the Tree View contains only the predefined system templates. However, as you add new templates, this list will grow.

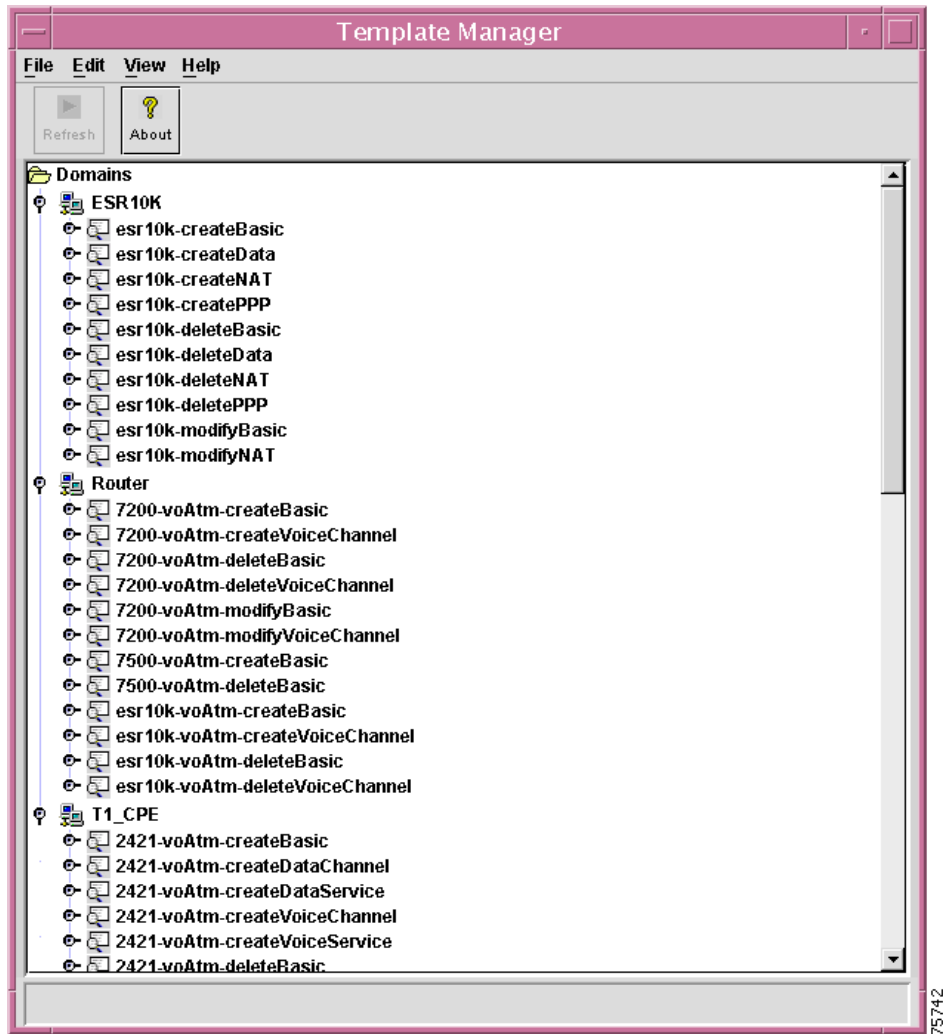


Note

The Template Manager window is the first window that appears when you click **Template Manager** in the CCNSC Subscriber Provisioning Control Center window.

The Tree View allows you to select the template for the various devices you will be provisioning. [Figure 4-1](#) shows an expanded Tree View of the CCNSC Subscriber Provisioning Template Manager window.

Figure 4-1 CCNSC Subscriber Provisioning Template Manager Window—Expanded



The Tree View displays a data tree that shows all of the domains to which you have been given access. Unlike the Network Inventory Manager data structure, the tree displayed in the Template Manager shows templates and template data objects, not network objects.







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.

The Tree View menu bar allows you to select various operations to perform from several drop-down menus. However, the operations that are enabled within these menus depend upon the template you have selected in the Template Tree. The Tree View window provides the menu options shown in [Table 4-1](#).

Table 4-1 Tree View Menu Options

Menu	Description
File	Allows you to close the Template Manager or exit CCNSC Subscriber Provisioning. The Close option stops the Template Manager process.
View	Provides two options. Refresh all of the templates in the Tree View, or only the selected templates.

The Tree View shows all of the objects that are currently defined in the Cisco Convergent Network Solution Center: Subscriber Provisioning NMS. The Network Tree uses the following conventions:

- Items in the Tree View can be preceded by a  icon to indicate that the tree can be expanded. Click the icon to expand to the next level of the hierarchy.
The  icon indicates that you have reached the lowest level in the hierarchy for that network object. You can click the icon to collapse to the previous level of the hierarchy.
- Root in the hierarchy is preceded by the  icon. The root is the highest level template folder and can be made up of any number of templates and complexes.
- Domains in the hierarchy are preceded by the  icon. A domain is a logical organization of your templates and can optionally be made up of any number of other templates. Default domains are provided for you, however, you can arbitrarily define other domains in order to provide a multilevel hierarchical structure.
- Templates in the hierarchy are preceded by the  icon. A Template, like the Service, is a logical organization of your templates. Default templates are provided for you, however, you can create other templates in order to provide a multilevel hierarchical structure.
- Template Data in the hierarchy are preceded by the  icon. Template data indicates the specific configuration that has been defined for a specific device.

To be managed by the Template Manager, a template must be listed in the Tree View. If the template is not listed, try expanding the hierarchy. If it is still not listed, it has not been defined in CCNSC Subscriber Provisioning.

Using the Context Menus

You can use the **Service Type** context menu to create new templates. To create a new template, you must right-click on a Domain icon and select **New Template**. The Enter Template Name window appears. You must enter a template name and click **OK** to add the new template or **Cancel** if you do not want to create the new template. Once created, it is added to the list of templates displayed in the Template Manager window.

You can use the **Template** context menu to do the following:

- Enter a new data name. To enter a new data name, you must right-click on a Template icon and select **New Data**. The Enter New Data Name window appears. You must enter a name for the template data file and click **OK** to add the new template data file or **Cancel** if you do not want to create the new template data file. Once created, it is added to the list of templates data file displayed in the Template Manager Window.
- Set the lock state. To set the lock state, you must right-click on a Template icon and select **Lock State and Lock, Unlock, or Override Lock**. The Confirm window appears. Click **Yes** or **No**. You can Lock or Unlock only Templates that you are using. When you set the lock state to lock, you are preventing other users from using this particular template. When you want to release it, you set the lock state to Unlock. You can override any current lock state currently set by other users.
- Delete the template. To delete a template, you must right-click on a Template icon and select **Delete**. The Confirm window appears. Click **Yes** or **No**. Once deleted, it is removed from the list of templates displayed in the Template Manager Window.

You can use the **Template Data** context menu to do the following:

- Access the Config Builder. To access the Config Builder, you must right-click on a Template Data icon and select **Config Builder**. The Template Config Builder window appears. For information about using it, see [“Using the Template Config Builder”](#).
- Delete a template data file. To delete a template data file, you must right-click on a Template Data icon and select **Delete**. The Confirm window appears. Click **Yes** or **No**. Once deleted, it is removed from the list of template data file displayed in the Template Manager Window.

Using the Template Config Builder

The Template Config Builder is where you construct templates, specifically, create the template body, identify the device for which you want to generate configuration files, and define values that are used for individual devices. You also use the Template Config Builder to preview device configurations.

The Template Config Builder provides two separate panes for constructing templates. The panes are as follows:

- **Template Body** pane (left pane)—Contains the nonchanging text and variables. The variables are replaced with the data, which is defined in the right pane, during generation of the configuration.
- **Template Data** pane (right pane)—Lists the devices for which configurations are generated and defines the values that are substituted for the variables in the **Template Body** pane.

[Figure 4-2](#) shows Template Config Builder window.

Specifying 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:
>
- 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.



Note

Do not precede reserved symbols with a backslash (\).

Following 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 dollar sign (\$) and 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 upon both the capacity of the Java Runtime Environment and your system's resources. You should be able to add several hundred attributes, however, more than 1000 attributes may cause excessive delays in calculating the spreadsheet.

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**.

Specifying Subtemplates

You can define sub templates, however, CCNSC Subscriber Provisioning does not support subtemplates at this time.

Conditionally Including a Block

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 are true, then `bodyN` is included. There can be zero or more `#elseif` clauses. 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:

```
<   less than
>   greater than
<=  less than or equal
>=  greater than or equal
==   equal
!=   not equal.
```

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.

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. CCNSC Subscriber Provisioning utilizes an embedded Tcl interpreter to parse, filter, and process these conditional control commands and expressions.

Using Expressions in a Template Body

You can use expressions in a template body. An expression combines *operands* with *operators* to produce a new value. For example, the expression `4 + 2` contains two operands, 4 and 2, and one operator, +; it evaluates to 6. The format of an expression in a template body is:

```
[expr arg1 arg2 ... argN]
```

where *arg1 arg2... argN* consists of a combination of operands, operators, and parentheses. White space are needed between the operands and operators and parentheses. Where possible, operands are interpreted as integer values. Integer values may be specified in decimal (the normal case), octal (if the first character of the operand is 0), or hexadecimal (if the first two characters of the operand are 0x). If an operand does not have one of the integer formats given above, it is treated as a floating-point number if that is possible. Floating-point numbers may be specified as the following examples: 2.1, 3., 6e4, 7.91e+16.

Operands:

Operands may be specified in any of the following ways:

- [1] As an numeric value, either integer or floating-point.
- [2] As a global template variable, using standard `<var>` notation. The variable's value will be used as the operand.

For some examples of simple expressions, suppose the variable `<a>` has the value 3 and the variable `` has the value 6. Then the command on the left side of each of the lines below will produce the value on the right side of the line:

```
[expr 3 + $<a>]      ->  6
[expr 2.1 + $<a>]    ->  8.1
[expr ($<a> + $<b>) * 5] ->  45
```

Operators:

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

- + ~ ! Unary minus, unary plus, bit-wise NOT, logical NOT.
- * / % Multiply, divide, remainder.
- + - Add and subtract.

< > <= >=	Boolean less, greater, less than or equal, and greater than or equal. Each operator produces 1 if the condition is true, 0 otherwise.
== !=	Boolean equal and not equal. Each operator produces a zero/one result.
&&	Logical AND. Produces a 1 result if both operands are non-zero, 0 otherwise.
	Logical OR. Produces a 0 result if both operands are zero, 1 otherwise.

Specifying 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).



Note

Only one row will display in CCNSC Subscriber Provisioning. Even though the Template Config Builder can display multiple devices, CCNSC Subscriber Provisioning supports only one device. *We advise not using the multiple row functionality.*

List each network element in the **Device** column by its name as defined in the Network Inventory window. This name is an arbitrary value that you define within the CCNSC Subscriber Provisioning 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.)

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 scroll bar 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 scroll bar 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.

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. However, 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** option from the **File** menu (or close and reopen the Template Config Builder).

The following tasks relate to the template data portion of the Template Config Builder:

- [Importing Data](#)
- [Missing Values](#)
- [Previewing Configurations](#).

Importing Data

To import template data from a text file, select the **Import** option from the **File** menu in the Template Config Builder window, then select the **Template Data** option from 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**. 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 is 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 of the Template Config Builder window, select a device and select the **Preview** option from the **Config** menu.



Note

Templates must be saved before you can preview the configuration; if you have not yet saved the template, you are asked if you want to. When you save the template, you must again select the device in the data pane and select the **Preview** option.

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 options from the floating menu that opens. The **Save All** option saves all text in the preview window to an ASCII file; the **Save** option saves only the text you have selected (prior to opening the **Options** menu). The **Find** option 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** option 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 select the **Commit** option from 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 window to search for text in either the body or the data portion of your template. First click in either the body or data side of the Template Config Builder window, then select the **Find** option from the **Edit** menu. Enter the text you want to locate in the **Find What** field, select the **Match Case** check box if you want the search to be case sensitive, then click **Find Next**. 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 selecting the **Rename Variable** option from the **Edit** menu
- by changing the variable with the Template Configuration Editor.

If you change a variable with the **Rename Variable** option, its value is retained. However, if you change the variable with the Template Configuration Editor, the value is lost, as described in the following example.

-
- Step 1** Open the Template Manager.
- Step 2** Create a template and a data object under it.
- Step 3** Right-click on the data object, then select **Config Builder**.
- Step 4** Enter the following in the **Template Editor** pane.
- ```
$ [num1] $ [num2]
```
- Step 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**.
- Step 6** Click **Save**.
- Step 7** Change the value **#[num1]** to **{num1}** by choosing **Rename Variable** from the **Edit** menu. Click **Save**, and notice that the **1** remains in the column.
- Step 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.
- 

## Supported Service Related Variables

CCNSC Subscriber Provisioning supports service related variables for each CCNSC Subscriber Provisioningconfigurable device and service type (voice and data, voice, and data). CCNSC Subscriber Provisioning generates their value or derives their value from a service request. However, a template can also contain non-service related variables required for configuring the device, for example an SNMP trap destination IP address.

**Note**

CCNSC Subscriber Provisioning fills in only the service related variables it supports. If you create or modify a template file that contains other variables, you must provide the values of these non-service related variables as additional input within a service request.

CCNSC Subscriber Provisioning has reserved global variable names for configuration required for voice and data service. In addition, CCNSC Subscriber Provisioning prohibits the use of any variable that starts with **System** followed by an underscore “**System\_**”; this usage is reserved for future global variable names.

**Note**

You should not modify the system variables. CCNSC Subscriber Provisioning will not substitute the values and it may cause conflicts in the future.

[Table 4-2](#) lists and describes the supported VoIPoATM service related variables. [Table 4-3](#) lists and describes the supported VoIPoPPP service related variables.

**Table 4-2 Supported Service Related Variables for VoIPoATM**

| Variable                     | Service Type | Description                                                                                                            |
|------------------------------|--------------|------------------------------------------------------------------------------------------------------------------------|
| System_atm_intf_no_list      | Basic        | Defines the list of ATM slots.                                                                                         |
| System_atm_intf_port_list    | Basic        | Defines the list of ATM interfaces.                                                                                    |
| System_COND_MGMT             | Basic        | Defines whether or not management IP is configured.                                                                    |
| System_Enable_Password       | Basic        | IAD's enable password.                                                                                                 |
| System_ima_group_list        | Basic        | Defines the list of IMA group numbers.                                                                                 |
| System_unique_ima_list       | Basic        | Defines the list of unique IMA group numbers.                                                                          |
| System_mgmt_atm_no           | Basic        | ATM slot number.                                                                                                       |
| System_mgmt_atm_port_no      | Basic        | ATM port number                                                                                                        |
| System_mgmt_ima              | Basic        | ATM IMA number.                                                                                                        |
| System_mgmt_ip_address       | Basic        | IP address.                                                                                                            |
| System_mgmt_pvc_vci          | Basic        | Virtual connection identifier.                                                                                         |
| System_mgmt_pvc_vpi          | Basic        | Virtual path identifier.                                                                                               |
| System_mgmt_subnet_mask      | Basic        | Subnet mask.                                                                                                           |
| System_IPAddress_CallAgent   | Basic        | P address of the Call Agent.                                                                                           |
| System_ServiceType_CallAgent | Basic        | Service type of the call agent, that is, the protocol employed by the call agent to control the Voice over IP gateway. |
| System_Version_CallAgent     | Basic        | Version of the protocol that is currently being used by the call agent.                                                |
| System_data_atm_intf_no_list | DataChannel  | Defines the list of ATM slot numbers.                                                                                  |
| System_data_atm_port_no_list | DataChannel  | Defines the list of ATM port numbers.                                                                                  |
| System_data_description_list | DataChannel  | Defines the list of interface descriptions.                                                                            |
| System_data_frf_version_list | DataChannel  | Defines the list of FRF versions.                                                                                      |

**Table 4-2 Supported Service Related Variables for VoIPoATM (continued)**

| Variable                       | Service Type | Description                                                              |
|--------------------------------|--------------|--------------------------------------------------------------------------|
| System_data_ima_list           | DataChannel  | Defines the list of ATM IMA numbers.                                     |
| System_data_pvc_vci_list       | DataChannel  | Defines the list of virtual connection identifiers.                      |
| System_data_pvc_vpi_list       | DataChannel  | Defines the list of virtual path identifiers.                            |
| System_data_qos_rate_list      | DataChannel  | Defines the list of quality of service rates.                            |
| System_data_qos_type_list      | DataChannel  | Defines the list of quality of service types.                            |
| System_serial_dldci_list       | DataChannel  | Defines the list of DLCI numbers.                                        |
| System_serial_intf_list        | DataChannel  | Defines the list of serial interfaces.                                   |
| System_voice_atm_no_list       | VoiceChannel | Defines the ATM slot number of each interface in the list.               |
| System_voice_atm_port_no_list  | VoiceChannel | Defines the ATM port number of each interface in the list.               |
| System_voice_channel_type_list | VoiceChannel | Defines the interface type of each interface in the list.                |
| System_voice_ima_list          | VoiceChannel | Defines the ATM IMA number of each interface in the list.                |
| System_voice_fqdn_list         | VoiceChannel | Defines the fully qualified domain name of each interface in the list.   |
| System_voice_ip_address_list   | VoiceChannel | Defines the IP address of each interface in the list.                    |
| System_voice_pvc_vci_list      | VoiceChannel | Defines the virtual connection identifier of each interface in the list. |
| System_voice_pvc_vpi_list      | VoiceChannel | Defines the virtual path identifier of each interface in the list.       |
| System_voice_qos_rate_list     | VoiceChannel | Defines the quality of service rate of each interface in the list.       |
| System_voice_qos_type_list     | VoiceChannel | Defines the quality of service type of each interface in the list.       |
| System_voice_subnet_mask_list  | VoiceChannel | Defines the subnet mask of each interface in the list.                   |

**Table 4-3 Supported Service Related Variables for VoIPoPPP**

| Variable               | Service Type | Description                                                        |
|------------------------|--------------|--------------------------------------------------------------------|
| System_Domain_Name     | Basic        | IAD's domain name.                                                 |
| System_Enable_Password | Basic        | IAD's enable password.                                             |
| System_Hostname        | Basic        | IAD's host name.                                                   |
| System_IPAddress       | Basic        | IAD's T1 serial interface's IP address towards aggregation router. |

**Table 4-3 Supported Service Related Variables for VoIPoPPP (continued)**

| Variable                          | Service Type                | Description                                                                           |
|-----------------------------------|-----------------------------|---------------------------------------------------------------------------------------|
| System_SubnetMask                 | Basic                       | IAD's T1 serial interface's IP address subnet mask towards aggregation router.        |
| System_T1_ChannelGroup_Speed      | Basic                       | T1 time slot speed (56 0r 64) towards aggregation router.                             |
| System__T1_ChannelGroup_Timeslots | Basic                       | T1 time slots range (for example, 1-10, 5-24, 1-24, etc.) towards aggregation router. |
| System_T1_Framing                 | Basic                       | T1 framing (ami or b8zs) towards aggregation router.                                  |
| System_T1_LineCode                | Basic                       | T1 line code (sf or esf) towards aggregation router.                                  |
| System_ESR10K_T1_Serial_Interface | Basic<br>Data<br>NAT<br>PPP | Serial interface's number on the aggregate router towards the IAD.                    |
| System_DHCP_Domain_Name           | DHCP                        | Domain name identifying the Ethernet subnet.                                          |
| System_DHCP_Exclude_High_IPAddr   | DHCP                        | High address of the static IP address space which is excluded from the DHCP space.    |
| System_DHCP_Exclude_Low_IPAddr    | DHCP                        | Low address of the static IP address space which is excluded from the DHCP space.     |
| System_IPAddress_Ethernet_0       | DHCP<br>Data                | IAD's Ethernet IP address.                                                            |
| System_SubnetMask_Ethernet_0      | DHCP<br>Data                | IAD's Ethernet subnet mask.                                                           |
| System_NAT_Global_IPNetmask       | NAT                         | IP netmask of the global IP address space used for NAT.                               |
| System_NAT_High_Global_IPAddr     | NAT                         | High address of the global IP address space used for NAT.                             |
| System_NAT_Low_Global_IPAddr      | NAT                         | Low address of the global IP address space used for NAT.                              |
| System_PPP_Chap_Password          | PPP                         | PPP Chap password from IAD to aggregation router.                                     |
| System_PPP_Chap_Username          | PPP                         | PPP Chap username from IAD to aggregation router.                                     |
| System_IPAddress_CallAgent        | Voice                       | Call agent's hostname or IP address.                                                  |
| System_ServiceType_CallAgent      | Voice                       | Call agents's service type (SGCP or MGCP).                                            |
| System_Version_CallAgent          | Voice                       | Call agent's version number.                                                          |

# Generating Configurations

When you are satisfied that the template will generate the configurations you want, select the devices you want to configure in the Template Config Builder window, then select the **Preview** option from the **Config** menu. The generated configuration file is displayed. To save the generated configuration file, you must select all of the text displayed in the Result Config: *<device name>* window, then click the right mouse button. Options allowing you to copy, find, save all, and save are available. Select the **Save All** or **Save** option to save selected information to the configuration file.

To download configurations to devices, use the Network Inventory Manager. See [Chapter 3, “Using the Network Inventory Manager.”](#)