
# Firmware Management

This chapter describes firmware management for TR-069 compliant devices in Cisco Broadband Access Center (Cisco BAC).

This chapter includes the following sections:



## Overview

Firmware management consists of maintaining and distributing sets of firmware image files to corresponding customer premise equipment (CPE) through the Cisco BAC system. A firmware rules template associates the firmware image files to groups of devices. Cisco BAC uses the rules in the associated firmware rules template to evaluate which firmware to download to the device.

The firmware management functionality allows the administrator to view firmware information on devices, to add firmware images to the database, and to apply the image files to specific devices.

Cisco BAC supports two mechanisms for CPE firmware management:

- Policy-based firmware management through firmware rules templates.
- Direct firmware management through device operations API.

  See Firmware Rule Templates, page 6-2, and Direct Firmware Management, page 6-4, for detailed information.

In the process of firmware management, regardless of the management method used, the device is instructed to obtain a new firmware image file from a file server. Cisco BAC provides a file service in its DPE. However, you can also direct CPE to other file servers.

Firmware rules can apply firmware to devices, based on a match of any preprovisioned or discovered device parameters, including device group membership, model, type, current state, type of connectivity, and so on.

The DPE triggers the firmware download by issuing a Download RPC with the location of the file on a file server and authentication credentials, if any. Cisco BAC supports HTTP and HTTP over SSL, called SSL/TLS in this chapter, for file downloads on DPE servers.

This download can be initiated in various modes:

- Firmware-rule based, in which firmware rules may or may not allow the download of the file requested by a device, or may download a different file. Firmware rules are executed whenever a device connects to the DPE.
- The device may contact the DPE, based on periodic contact after a reboot or a specific action, such as an user-initiated upgrade by clicking in the local user interface of the device. Regardless of how the device contacts the DPE, the firmware rules, determine whether the upgrade is needed and is allowed at the time of the particular interaction.
- Proxy, in which an external application invokes the API download operation for a specific device and specifies the firmware image file location. The DPE then executes the Download RPC on the device, causing the device to download the file from the specified location.

  The download can occur in three ways:

  - Immediate, in which the DPE connects to the device and instructs the device to download the firmware.
  - On-connect, in which the DPE instructs the device to download the firmware on next contact with the device.
  - Asynchronous, in which the DPE instructs the device to download the firmware without reserving the PACE thread at RDU. The operation result is sent to the API client in an AyncOperationEvent.

The multiple instance object support is available in the firmware template, which facilitates discovering the device parameters for multiple object instances. The parameter value associated with the discovered object instances is matched with the parameter values specified in the template, and further firmware rule is applied based on the match condition. For more information on multi-instance object support, see Multi-Instance Object Support, page 4-5.

# Firmware Management Mechanisms

This section describes CPE firmware-management mechanisms provided by Cisco BAC. These comprise:

- Policy-based firmware management through firmware rules templates. See Firmware Rule Templates, page 6-2, for detailed information.
- External firmware management through proxy operations. See Direct Firmware Management, page 6-4, for detailed information.

## Firmware Rule Templates

You use rule templates to set policy-based firmware management. The firmware rules templates are XML documents written according to a published schema document. Each template must be stored in a file and uploaded into Cisco BAC.

Each firmware rules template contains one or more rules which trigger firmware updates based on specific conditions. Any number of such templates can be added to Cisco BAC, through the administrator user interface or API. The template is associated with a Class of Service object using the `COS_CWMP_FIRMWARE_RULES_FILE` property. Each device, in turn, is assigned to a Class of Service. (For information on Cisco BAC object relationships, see Cisco BAC Device Object Model, page 4-2.)

In this model, you can make convenient updates to the definition of the rules, which apply to a large number of devices. When the rules template is updated, CPE that are indirectly associated with the template through the Class of Service are managed according to the new policy.

When the device establishes a connection with Cisco BAC, its firmware and configuration are automatically synchronized based on the configuration and firmware rules cached at the DPE. First, the firmware rules are executed, and if appropriate, the device firmware is updated. Then, the device configuration is synchronized.

Earlier to this release, the only file type supported by Cisco BAC was "1 Firmware Upgrade Image". With this release of Cisco BAC, firmware download is enhanced to allow upgrade of CPE devices with different vendor specific TR-069 download file types. Now vendor can define file type in both internal and external firmware rule tag.

Cisco BAC provides a two-stage firmware rule processing. First, the templates are processed at the RDU where template constructs such as conditionals and substitutable parameters are interpreted (See Using Template Constructs with Firmware Rule Templates, page 6-13).

This processing allows customization of rules for devices based on data available at the RDU (such as device properties and grouping). This data could be preprovisioned by using the API. Data previously discovered from the device and stored at the RDU can also be used in constructing the templates.

After the template has been processed, the resulting rules are sent to the DPEs in the device's provisioning group. The rules, in turn, can have dynamic matching criteria, which enable further granularity in firmware rules policy.

To determine if a firmware update is needed, the rules engine at the DPE evaluates the firmware rules. Firmware rules allow a firmware update to be triggered on match of:

- Inform event types
- Device RequestDownloadRPC arguments
- Inform parameter values
- Any other device parameter values
- MaintenanceWindow time

**Note** You can use the MaintenanceWindow option to schedule firmware downloads to a device. For details, see Device Contact During MaintenanceWindow, page 5-12.

Together, these rules provide a powerful mechanism to create policy for managing firmware. For example, an administrator can write a rule that forces all devices of a certain model with a certain current firmware version to upgrade to a different firmware, during a specific service time window.

The DPE logs an entry for all cases of firmware selection by using rules. It also logs an entry if none of the rules match. This logging mechanism can be useful to track devices that have no firmware image file associated with them or if the device firmware is simply up to date.

Cisco BAC uses the XML schemas that are defined in various files to generate instructions for device configurations. Table 6-1 lists these files and their locations.

*Table 6-1 Files Used in Firmware Rules Template Processing*

| File | Purpose | Options Available in Cisco BAC |
|---|---|---|
| Firmware Rules Template Samples | Defines device configuration | Sample templates |
| | **Sample templates are located at:**<br>*<BPR_HOME>/rdu/samples/cwmp* | |
| Firmware Rule Template Schema | Validates firmware rules template syntax | Default template schemas |
| | **Default template schemas are located at:**<br>• Firmware rules template schema <BPR_HOME>/rdu/templates/cwmp/schema/FirmwareTemplateSchema.xsd<br>• Common template schema <BPR_HOME>/rdu/templates/cwmp/schema/CommonTemplateConstructs.xsd | |
| Parameter Dictionary | Validates firmware rules template content | Default dictionaries |
| | **Default dictionaries are located at:**<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr069-cwmp-dictionary.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr098-cwmp-dictionary.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr104-cwmp-dictionary.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr106-cwmp-dictionary.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr196-cwmp-dictionary-v1.1.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr196-cwmp-dictionary-v2.0.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr196-cwmp-dictionary-IGD-v1.1.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/tr196-cwmp-dictionary-IGD-v2.0.xml<br>• <BPR_HOME>/rdu/templates/cwmp/dictionary/basic-cwmp-dictionary.xml | |
| Parameter Dictionary Schema | Validates parameter dictionary syntax | Default dictionary |
| | **Parameter dictionary schema is located at:**<br>Schema for TR-069, TR-098, TR-104, TR-106, TR-181, and TR-196 dictionaries <BPR_HOME>/rdu/templates/cwmp/schema/TemplateDictionarySchema.xsd | |

# Direct Firmware Management

The Cisco BAC device operations API allows the OSS to execute operations on individual devices. Among other operations, Cisco BAC provides standard CWMP RPC operations.

Managing firmware using device operations API gives the OSS precise control over operations performed on CPE. The OSS issues specific API calls which correspond to remote procedure calls (RPCs) necessary for CPE firmware update.

For example, a Download RPC can be invoked on the device using a corresponding API call. This command contains the URL of the firmware image file that the device should download and, if necessary, authentication credentials.

For detailed information on device operations, see CWMP Device Operations, page 14-1.

### File Service

In the process of firmware management and regardless of which management method is used, the device is instructed to obtain a new firmware file from a file server. Cisco BAC provides a file service in its DPE servers. However, CPE can also be directed to other file servers if necessary. For the various configuration options that Cisco BAC supports, see CWMP Service Configuration, page 12-1.

# Managing Firmware Files

Firmware file management comprises the management of firmware image files and firmware rule template files. This functionality allows the administrator or applications by using the API to add, delete, or replace firmware image files and firmware rule template files, and view and search firmware image files and file information.

You can manage firmware through the administrator user interface, or through the API. To manage firmware image file and firmware rule templates on the administrator user interface, choose **Configuration > Files**.

Firmware rule template files determine the firmware image for a device. These files are stored in the RDU database with the file type `Firmware Rules Template`.

Firmware image files are stored in the RDU database with file type `Firmware File`. Each firmware image file has a firmware version that is specified by using the `Firmware Version` attribute. The DPE uses the firmware version information to evaluate firmware rules.

**Note** While firmware images are managed using the central server (RDU), they are automatically distributed or deleted from the appropriate DPEs.

Firmware file management allows the following operations for each file type:

*Table 6-2 Firmware File Management Operations*

| Firmware Image File | Firmware Rules Template |
|---|---|
| Add | Add<br>You can add a firmware rule template file to the system only if it is valid; otherwise, Cisco BAC displays error messages explaining the type of error in the template. |
| Delete<br>You cannot delete an existing firmware image file if it is referenced in a firmware rules template.<br>To successfully delete a firmware image file, remove the reference to the firmware file in the firmware rules template. | Delete<br>You cannot delete a firmware rules template if it is referenced by a Class of Service.<br>To successfully delete a firmware rules template, remove the reference to the Class of Service. |
| Retrieve contents | Retrieve contents |
| Retrieve file attributes, such as size, name, and properties | - |

*Table 6-2 Firmware File Management Operations (continued)*

| Firmware Image File | Firmware Rules Template |
|---|---|
| Replace contents and/or modify file attributes/properties<br><br>You can replace an existing firmware image file even if the file is associated to a firmware file template using the API. The administrator user interface, however, notifies of existing associations before replacing a firmware image file.<br><br>When a firmware image file's contents are replaced, IGS regenerates firmware rules for each affected device and IGS distributes them to the DPEs in the device's provisioning group. Subsequently, when the device contacts the DPE, new rules are executed. | Replace file contents, modify file attributes and properties, or both |
| Search by name, suffix, regex or file type | - |
| - | View templates in tabular form from the administrator user interface<br><br>Templates appear in tabular form only if they do not include conditionals. |

# Authoring Firmware Rules Templates

Firmware rule templates in Cisco BAC are based on an XML schema file located at <*BPR_HOME*>/rdu/templates/cwmp/schema/FirmwareTemplateSchema.xsd.

Firmware rules execute after processing an Inform from the device. They are also triggered after the device issues a RequestDownload RPC.

A firmware rules template comprises:

- **FirmwareTemplate**—The root element, which can contain a Prerequisites tag, and one or more named FirmwareRule elements.

  Firmware rules are processed in order and, once a firmware rule matches, further rules are not processed.

  - **TemplateVersion**—A TemplateVersion attribute has been introduced to support the different firmware file types and the download of large firmware files. In the FirmwareTemplate configuration file, the template version must be set to 3.0 to enable multi-instance object support feature. If the template version is not provided or is set to 1.0, the new features will not work as expected.

- **Prerequisites**—Contains conditions which must be met before the rules listed in the firmware rule template are processed. Prerequisites may contain zero or one MaintenanceWindow, and zero or more Expressions.

  You can enable Expressions and MaintenanceWindow for firmware templates just as it is done for configuration templates.

  - MaintenanceWindow—Describes the time range within which the firmware rule template is valid for processing. For detailed information, see Prerequisites, page 5-9.

  - Expression—Zero or more expression for evaluating this rule; has the same syntax and definition as the Expression defined for the *FirmwareRule* element. For detailed information, see Expression and Regular Expression, page 6-7.

- **FirmwareRule**—Each FirmwareRule element provides:
  - Expression—Zero or more expressions for evaluating this rule. The rules triggers a firmware update if all expressions in a given rule match. For detailed information, see Expression and Regular Expression, page 6-7.
  - InternalFirmwareFile or ExternalFirmwareFile—One of the two must be specified.
  - InternalFirmwareFile is used if the Cisco BAC file service is utilized, and the firmware image file has been added to the system using the API or the administrator user interface.
  - ExternalFirmwareFile provides information for firmware image files located on external file servers.

  For detailed information, see Internal Firmware File vs. External Firmware File, page 6-10.

# Expression and Regular Expression

Expressions provide a conditional way to apply the firmware rules based on value match. If the value is a range, Regular Expression (Regex) is used for defining the firware rule in firmware rule template. Regular expression provides a conditional way to apply the firmware rules based on Regex pattern match. Each expression or Regex must provide a *ParameterName*, *InformParameterName* or *RpcArgumentName* tag, one or more *Value* tags and an *Operator* tag.

The value of the event returned from the CPE, and the condition specified for the event determines whether the firmware rule is applied to the CPE. The firmware rules are defined in the firmware template with the appropriate event, Regex pattern associated with the event, and the condition for applying the firmware rule.

Regex support is also available in the configuration templates prerequisites. For more information on configuring prerequisites, see Configuring Prerequisites, page 5-13.

Cisco BAC processes these tags in sequence to match and assign a firmware image file to the CPE.

### ParameterName

Specifies the name of a TR-069 parameter. The *ParameterName* is validated by using the relevant parameter dictionary.

The DPE may have this parameter available from Inform or from prior GetParameterValues RPCs calls within the same session. If the parameter value is not available in the session when processing the rule, the DPE queries the device for the missing parameter values before proceeding with evaluation of the rules.

For additional information, see Parameters, page 5-6.

### InformParameterName

Specifies the name of an Inform parameter that is not described in the parameter dictionary. This entry is not validated.

For example, the expression in the following example evaluates to *true* if the device `Inform.EventCode` includes either of the specified values:

```
<Expression>
    <InformParameter>Inform.EventCode</InformParameter>
    <Value>1 BOOT</Value>
```

```
    <Value>3 SCHEDULED</Value>
    <Operator>match</Operator>
</Expression>
```

Cisco BAC supports the following parameter names in the InformParameter tag:

- Inform.DeviceId.Manufacturer
- Inform.DeviceId.ManufacturerOUI
- Inform.DeviceId.ProductClass
- Inform.DeviceId.SerialNumber
- Inform.EventCode

For information on the values for Inform.EventCode, refer to the DSL Forum's Technical Report on TR-069.

#### Value

Specifies data for the Parameter. One or more possible values might be listed for a given parameter. The data type for a value is validated by using a dictionary when possible.

In the expression, if you need to set a value that contains comma, then specify each item of the value separated by a comma in separate value tag. For example, the comma-separated value *12.4(22)T, RELEASE SOFTWARE* (fc1) can be set as in the following expression:

```
<Expression>
<ParameterName>InternetGatewayDevice.DeviceInfo.SoftwareVersion</ParameterName>
<Value>12.4(22)T</Value>
<Value>RELEASE SOFTWARE (fc1)</Value>
<Operator>match</Operator>
</Expression>
```

You can also set the value as empty.

#### RpcArgumentName

Specifies the name for a parameter that the device reports. The possible values are RequestDownload.FileType and RequestDownload.FileTypeArg*

- RequestDownload.FileType indicates the type of file that the device is requesting to download.
- RequestDownload.FileTypeArg* indicates any argument that the device might have provided in a download request message. The asterisk (*) denotes the actual argument name.

  See Example 6-3.

#### Operator

Evaluates the Parameter and the Value. In evaluating the expression, the *Operator* may be one of the following:

- match—Specifies that the device parameter value must match at least one of the values in a case-sensitive comparison.
- matchIgnoreCase—Specifies that the device parameter value must match at least one of the values in a case-insensitive comparison.
- matchAll—Specifies that the device parameter value must match all of the values in a case-sensitive comparison.
- matchAllIgnoreCase—Specifies that the device parameter value must match all of the values in a case-insensitive comparison.

- noMatch—Specifies that the device parameter value must not match any of the values in a case-sensitive comparison.
- noMatchIgnoreCase—Specifies that the device parameter value must not match any of the values in a case-insensitive comparison.

#### *Example 6-1 Expression - match InformParameterName*

In the following sample expression, the match condition indicates that the subsequent rules are valid when the *InformParameter* `Inform.EventCode` has exactly the value *1 BOOT*. The device reports this value in the Inform message when it contacts the autoconfiguration server (ACS).

```
<Expression>
<InformParameterName>Inform.EventCode</InformParameterName>
          <Value>1 BOOT</Value>
          <Operator>match</Operator>
</Expression>
```

#### *Example 6-2 Expression - match RpcArgumentName (RequestDownload.FileType)*

In the following sample expression, the match condition indicates that the subsequent rules should be in effect when the RPCArgumentName `RequestDownload.FileType` matches exactly the value *1 Firmware Upgrade Image*.

```
<Expression>
    <RpcArgumentName>RequestDownload.FileType</RpcArgumentName>
    <Value>1 Firmware Upgrade Image</Value>
    <Operator>match</Operator>
</Expression>
```

#### *Example 6-3 Expression - match RpcArgumentName (RequestDownload.FileTypeArg)*

In the following sample expression, the match condition indicates that the subsequent rules should be in effect when the RPCArgumentName `RequestDownload.FileTypeArg.Version` matches the value *1.1*.

**Note** The CWMP specification defines `Version` to be a FileTypeArg that you can use if File Type is Web Content.

```
<Expression>
    <RpcArgumentName>RequestDownload.FileType</RpcArgumentName>
    <Value>2 Web Content</Value>
    <Operator>match</Operator>
</Expression>
<Expression>
    <RpcArgumentName>RequestDownload.FileTypeArg.Version</RpcArgumentName>
    <Value>1.1</Value>
    <Operator>match</Operator>
</Expression>
```

#### *Example 6-4 Expression - noMatch ParameterName*

In the following sample expression, the match condition indicates that the subsequent rules should be in effect when the *Parameter* `InternetGatewayDevice.DeviceInfo.SoftwareVersion` does not match the software version *1.02*.

```
<Expression>
   <ParameterName>InternetGatewayDevice.DeviceInfo.SoftwareVersion</ParameterName>
```

```
   <Value>1.02</Value>
   <Operator>noMatch</Operator>
</Expression>
```

While specifying the Regex pattern, the operator tag may be one of the following:

- regexMatch—Indicates a condition that is set to ensure that the firmware rule is applied only when the device parameter value matches the Regex pattern.
- regexNoMatch—Indicates a condition that is set to ensure that the firmware rule is applied only when the device parameter value does not match the Regex pattern.

***Example 6-5 Expression - regexMatch***

In the following sample expression, the regexMatch condition indicates that the firmware rule comes into effect when the *Parameter* `Inform.EventCode` matches Regex pattern `^[0-9]\s\w`.

```
<Expression>
    <InformParameterName>Inform.EventCode</InformParameterName>
          <Value>^[0-9]\s\w</Value>
          <Operator>regexMatch</Operator>
</Expression>
```

If the condition is specified as regexNoMatch, the firmware rule is applied only when the `Inform.EventCode` does not match the Regex pattern `^[0-9]\s\w` specified in the value tag.

**Note** Under expression, if the operator is regexMatch or regexNoMatch, ensure that only one value tag is used.

# Internal Firmware File vs. External Firmware File

The Internal and External firmware image file elements define whether the firmware image files are located within the Cisco BAC file server or at remote file server.

## InternalFirmwareFile

The InternalFirmwareFile element describes the filename of the firmware image that was added to the RDU and automatically distributed to DPEs and the delivery transport method employed to download the firmware image to a device. It comprises:

- *FileName*—Specifies the name of the file in the RDU database.
- *DeliveryTransport*—Specifies service HTTP 1 or service HTTP 2 transfer.

  You should configure the corresponding file service (HTTP 1 or HTTP 2) on the DPE. For configuration details, see the *Cisco Broadband Access Center 3.8 DPE CLI Reference*.

  If you define multiple services on the DPE with the same transport, for example, two HTTP, the DPE chooses the first one to service the device.
- *FileType*—Specifies the vendor specific download FileType element. This element is optional. The default value is 1 Firmware Upgrade Image. The template version must be set to 2.0 to enable this feature.
- *Location*—Specifies the location where the files are stored. Possible values are Cache and File System. The default value is Cache. Specifying the location is optional and to enable this, you must set the template version to 2.0.

- *FileSize*—Specifies the size of the file. This is mandatory when the location is File System. The FileSize can also be set to 0.

**Examples**

```
<InternalFirmwareFile>
        <FileName>sample-firmware-image1.bin</FileName>
        <DeliveryTransport>service http 1</DeliveryTransport>
        <FileType>X CISCO_COM DLCImage</FileType>
        <Location>File System</Location>
        <FileSize>1234</FileSize>
</InternalFirmwareFile>
```

## ExternalFirmwareFile

The ExternalFirmwareFile element describes the name of a firmware image file located at a remote server. It comprises:

- *FileURL*—Specifies the URL for a firmware image file located at remote location.
- *FileSize*—Specifies the size of the firmware image file to be downloaded. The FileSize can also be set to 0.
- *AuthenticationCredentials*—Specifies the username and password to use if HTTP authentication is enforced by the file server. The username and password are transmitted using the Download RPC to the device.
- *FileType*—Specifies the vendor specific download FileType element. This element is optional. The default value is 1 Firmware Upgrade Image. The template version must be set to 2.0 to enable this feature.

**Note** Ensure that you use SSL/TLS for CWMP to avoid transferring passwords in clear text.

You can use substitutable parameters to make the template-processing engine derive a device-specific username and password from the device record when processing firmware rules at the RDU.

**Examples**

```
<ExternalFirmwareFile>
        <FileURL>http://imageserver.isp.com/sample-firmware-image.bin</FileURL>
        <FileSize>3449</FileSize>
        <AuthenticationCredentials>
            <HttpUserName>test</HttpUserName>
            <HttpPassword>changeme</HttpPassword>
        </AuthenticationCredentials>
        <FileType>X CISCO_COM DLCImage</FileType>
</ExternalFirmwareFile>
```

# Sample Firmware Rules Template

The following is an example of a firmware template that supports the new feature like multi-instance object support and features supported in earlier versions of firmware template.

```
<FirmwateTemplate templateVersion="3.0">
    <Prerequisites>
```

```
            <MaintenanceWindow>
                <StartTime>01:00:00</StartTime>
                <Duration>5:00</Duration>
            </MaintenanceWindow>
            <Expression>
                <InformParameterName>Device.DeviceInfo.EventCode</InformParameterName>
                <Value>1 BOOT</Value>
                <Operator>match</Operator>
            </Expression>
            <Expression>
                <ParameterName>Device.DeviceInfo.HardwareVersion</ParameterName>
                <Value>W3GFP-100-Rev 6.0</Value>
                <Operator>matchIgnoreCase</Operator>
            </Expression>
            <Expression>

<ParameterName>Device.Services.FAPService.{i}.REM.UMTS.GSM.Cell.{i}.BSIC</ParameterName>
                <InstanceConfiguration>
                    <Instance>
                        <Path>Device.Services.FAPService.</Path>
                        <Value>last()</Value>
                    </Instance>
                    <Instance>
                        <Path>Device.Services.FAPService.{i}.REM.UMTS.GSM.Cell.</Path>
                        <Value>compare(LAC lessThanEquals 65534)</Value>
                    </Instance>
                    <MatchCondition>OR</MatchCondition>
                </InstanceConfiguration>
                <Value>18</Value>
                <Operator>matchAllIgnoreCase</Operator>
            </Expression>
        </Prerequisites>
        <FirmwareRule name="InternalFileRule">
            <Expression>
                <InformParameterName>Device.DeviceInfo.SoftwareVersion</InformPaameterNam>
                <Value>5.4.0</Value>
                <Operator>match</Operator>
            </Expression>
            <Expression>

<ParameterName>Device.Services.FAPService.{i}.REM.UMTS.GSM.Cell.{i}.BSIC</ParameterName>
                <InstanceConfiguration>
                    <Instance>
                        <Path>Device.Services.FAPService.</Path>
                        <Value>compare(DeviceType equalsIgnoreCase fap)</Value>
                    </Instance>
                    <Instance>
                        <Path>Device.Services.FAPService.{i}.REM.UMTS.GSM.Cell.</Path>
                        <Value>1</Value>
                    </Instance>
                    <MatchCondition>OR</MatchCondition>
                </InstanceConfiguration>
                <Value>18</Value>
                <Operator>matchAllIgnoreCase</Operator>
           </Expression>
            <InternalFirmwareFile>
                <FileName>sample-firmware-image.bin</FileName>
                <DeliveryTransport>HTTP</DeliveryTransport>
                <Location>File System</Location>
                <FileSize>1234</FileSize>
                <FileType>X CISCO_COM DLCImage</FileType>
            </InternalFirmwareFile>
        </FirmwareRule>
        <FirmwareRule name="ExternalFirmwareRule">
```

```
        <Expression>
            <InformParameterName>Device.DeviceInfo.SoftwareVersion</InformParameterName>
            <Value>5.4.1</Value>
            <Operator>match</Operator>
        </Expression>
        <ExternalFirmwareFile>
            <FileURL>http://10.10.10.10:889/sample-firmware-image.bin</FileURL>
            <FileSize>3449</FileSize>
            <AuthenticationCredentials>
                <HttpUserName>test</HttpUserName>
                <HttpPassword>changeme</HttpPassword>
            </AuthenticationCredentials>
            <FileType>X CISCO_COM DLCImage</FileType>
        </ExternalFirmwareFile>
    </FirmwareRule>
</FirmwareTemplate>
```

# Using Template Constructs with Firmware Rule Templates

You can use the Cisco BAC template-processing mechanism to generate customized configurations for a large numbers of CPE by using a small number of templates. The use of template constructs enables this mechanism.

Template constructs are any of the **tc:include**, **tc:if**, and **tc:choose** conditional statements, which are used in conjunction with Cisco BAC properties. The template processor processes the constructs when generating instructions for a device. The instructions are then forwarded to the DPE and cached there.

FirmwareRules, on the other hand, are tags within a firmware rule template and describe the firmware image to send to the device. Firmware rules may include expressions that are evaluated when the device contacts Cisco BAC to retrieve configuration. If the expressions evaluate to *true*, the device is instructed to download a specific firmware image file.

Firmware rules can apply firmware to devices based on a match of any preprovisioned or discovered device parameters, including device group membership, model, type, current state, type of connectivity, and so on. This processing is done at the RDU, with the preprovisioned or discovered data available at the central server.

This release supports the following general classes of template constructs:

- Parameter substitutions—Rule content can be inserted based on parameter values stored on device records or other objects within the Cisco BAC data model.

  See Using Parameter Substitution, page 6-14 for more information.

- Includes—One template may include another.

  See Using Includes, page 6-14 for more information.

- Conditional expressions—Rule content can be inserted based on evaluation of conditional statements.

  See Using Conditionals, page 6-15 for more information.

You use XML tags with the prefix **tc** to specify these template constructs.

**Note** Elements prefixed by **tc** are generic constructs that are the same for firmware rule templates and configuration templates.

Cisco BAC firmware rules constructs are based on the XML schema defined in the file located at:

- <BPR_HOME>/rdu/templates/cwmp/schema/FirmwareTemplateSchema.xsd
- <BPR_HOME>/rdu/templates/cwmp/schema/CommonTemplateConstructs.xsd

**Note** The XML namespace of the Cisco BAC Common Template constructs is defined as `xmlns:tc='urn:com:cisco:bac:common-template'`.

# Using Parameter Substitution

Values from the Cisco BAC property hierarchy are substituted into a template by using the `VAR()` construct, in order to produce firmware rules specific to a given device. The `VAR()` construct can appear in an XML element value or element attribute. It can also be used to substitute full or partial values.

The following list describes the constructs that Cisco BAC supports for parameter substitution:

- Cisco BAC property value into XML element content
- Cisco BAC property value into XML element attribute
- Default value
- XML partial element content
- Values with special characters

For syntax and specific examples, see Using Parameter Substitution, page 5-17.

# Using Includes

You use Include files to build a set of reusable template snippets. These files are useful for defining options that are common across many classes, without having to duplicate the options in several templates.

You can include the content of a particular file to a template by using the **tc:include** construct. After inserting the content of included files into the host template, the Parameter Dictionary specified in the host template, validates the content of the resulting template.

**Note** If included templates use objects and parameters that are not defined in the same dictionary as the host template, parameter validation fails during instruction generation.

The **tc:Include** element specifies the *href* attribute, where *href* identifies the name of the Cisco BAC template file that is included in the host template. Use double quotation marks (") when using an Include directive in a template.

**Note** When one template is included in another, the parameter dictionary and prerequisite tags from the included template are ignored. The schema of the firmware template enforces the location of an Include tag within a firmware template.

For syntax and specific examples, see Using Includes, page 5-18.

# Using Conditionals

Cisco BAC supports powerful conditional expressions in template constructs to enable ultimate configuration customization. You can use these conditional expression constructs to include or exclude blocks of text within a template. These construct elements are **tc:if**, **tc:choose**, and **tc:when**. For detailed information and specific examples on conditionals, see Using Conditionals, page 5-19.

By using conditionals, you can also enable devices to bypass firmware upgrade. If a device does not match the required conditions specified in the firmware rules template, then the device bypasses an upgrade. See Example 6-6.

### *Example 6-6 Firmware Upgrade Bypass*

The following is an example of a firmware rule template that describes a firmware upgrade bypass using the **if** construct.

If `checkVersion` is set to *true*, the rules check the software version on the device; if the version does not match, firmware upgrade is bypassed. If `checkVersion` is set to *false*, the rules do not check for the software version and the device gets instructions on firmware download.

```
<tc:Template xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:tc="urn:com:cisco:bac:common-template" xmlns="urn:com:cisco:bac:firmware-template"
xsi:schemaLocation="urn:com:cisco:bac:common-template CommonTemplateConstructs.xsd">
<FirmwareTemplate>
    <ParameterDictionaries>
        <ParameterDictionary>tr069-cwmp-dictionary.xml</ParameterDictionary>
    </ParameterDictionaries>
    <!--  Upgrade rule: if software version is 0.00.22, direct the device to download
sample-firmware-image.bin -->
    <!--  devices that do not have software version 0.00.22 , will bypass firmware
upgrade. -->
<FirmwareRule name="LinksysWAG54G2Rule">
    <tc:if test="equals(VAR(name=/cpe/checkVersion,defaultValue=false), true)">
        <Expression>
         <ParameterName>InternetGatewayDevice.DeviceInfo.SoftwareVersion</ParameterName>
         <Value>0.00.22</Value>
         <Operator>matchIgnoreCase</Operator>
        </Expression>
    </tc:if>

    <InternalFirmwareFile>
        <FileName>sample-firmware-image.bin</FileName>
        <DeliveryTransport>service http 1</DeliveryTransport>
    </InternalFirmwareFile>
</FirmwareRule
```