CHAPTER 4

CPE Management Overview

This chapter describes the management of customer premises equipment (CPE) by using the CPE WAN Management Protocol for Cisco Broadband Access Center (BAC). The following are the sections covered in this chapter:

Overview

Cisco BAC communicates with CPE through the CPE WAN Management Protocol (CWMP) according to parameters described in the TR-069 and other related data model specifications. CWMP encompasses secure management of CPE, including:

- Autoconfiguration and dynamic service provisioning
- Firmware management
- Device diagnostics
- Performance and status monitoring

Cisco BAC supports devices based on the TR-069, TR-098, TR-104 and TR-106 standards. This support includes Ethernet and ADSL gateway devices, wireless gateways, VoIP ATAs, and other devices compliant with CWMP. This release also provides for runtime-extensible data models to support any upcoming data-model standards or any vendor-specific data models.

Cisco BAC Device Object Model

The Cisco BAC device object model is crucial in controlling the configuration and firmware rules that are generated as instructions for the DPE to manage devices. This process occurs at the RDU, and is controlled through named attributes and relationships.

The main objects in the device object model are:

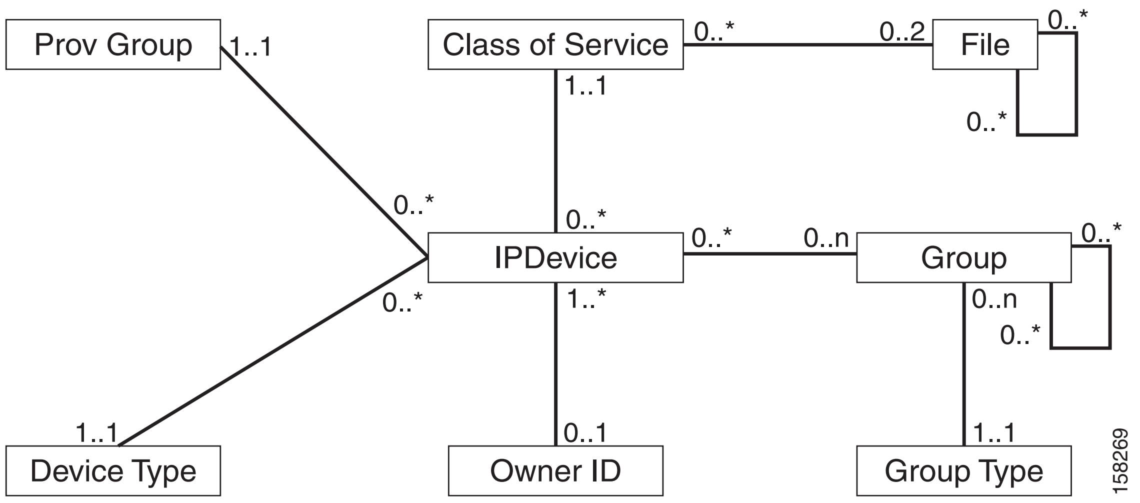
- IPDevice—Represents a network entity that requires provisioning.
- Owner ID—Represents an external identifier for a subscriber.
- Device Type—Represents the type of the device.
- ProvGroup—Represents a logical grouping of devices serviced by a specific set of DPEs.
- Class of Service—Represents the configuration profile to be assigned to a device.
- File—Serves as a container for files used in provisioning that include templates and firmware images.
- Group—Is a customer-specific mechanism for grouping devices.

Common among the various objects in the Cisco BAC device data model are:

- Name. For example, Gold Class of Service
- Attributes. For example, Device ID and a fully qualified domain name (FQDN)
- Relationships. For example, the relationship of a device to a Class of Service
- Properties. For example, a property which specifies that a device must be in a provisioning group.

See Figure 4-1 for a description of the interaction between the various objects in the device data model.

Figure 4-1 Device Object Model



In the Cisco BAC device object model, the IPDevice is related to the Class of Service, the Provisioning Group, and the Device Type. The Class of Service is then related to the Configuration Template and the Firmware Rules Template. Files can be related to each other, such as the firmware rules template being related to the firmware image.

Table 4-1 describes the attributes and relationships unique to each object in the data model.

Table 4-1 Device Object Relationships

Object	Related to ...
IP Device • Could be preregistered or unregistered (See Device Deployment in Cisco BAC, page 4-8). • Attributes include Device ID (OUI-Serial) and FQDN	• Owner ID • Provisioning Group • Class of Service • Device Type
Owner ID • Is associated with devices and, therefore, cannot exist without a device related to it. • Enables grouping; for example, you can group all devices belonging to *Joe*.	IP Device
Device Type • Stores defaults common to all devices of a technology, specifically CWMP. • Enables grouping; for example, you can group all CWMP devices.	IP Device
File • Stores files used in provisioning; for example, configuration template and firmware rules template	Class of Service
Class of Service • Attributes include Type, Name, and Properties. (For details, see Class of Service, page 4-3.)	• IP Device • File • Configuration Template (optional) • Firmware Rules Template (optional)

Class of Service

Class of Service is an RDU abstraction that represents the configuration to be handed to the device in the form of templates. It enables you to group devices into configuration sets, which are service levels or different packages that are to be provided to the CPE.

The different Classes of Service are:

- Registered—Specified by the user when the device is registered. This Class of Service is explicitly added to the device record via the application programming interface (API).
- Selected—Selected by the RDU for a device that, for one reason or another, cannot retain its registered Class of Service.
- Related—Related to the device by being registered, selected, or both.

If the selected Class of Service for a device is changed, the Instruction Generation Service regenerates instructions for the device configuration. If the registered Class of Service for a device is changed, it regenerates instructions for the generated device configuration even if it is not the selected Class of Service, since it could impose a policy that would change the selected Class of Service.

Other concepts related to the device data model are:

- Property Hierarchy, page 4-4
- Custom Properties, page 4-4

Property Hierarchy

While processing configuration templates for substitutable parameters, Cisco BAC searches objects for this property using a certain order called Property Hierarchy.

Cisco BAC properties allow you to access and store data in Cisco BAC using the API. Preprovisioned, discovered, and status data can be retrieved through the properties of corresponding objects, using the API. Properties also enable you to configure Cisco BAC at the appropriate level of granularity (from system level to device groups and to individual devices).

The Cisco BAC property hierarchy gives you the flexibility to define system-wide or service class defaults that can be overridden by individual devices.

Cisco BAC allows you to store any number of properties on objects in its data model. You can reference these properties in configuration templates or firmware rules. You can use properties in this property hierarchy:

1. Device
2. Group (priority is set through the Group Type, see Managing Group Types, page 16-19)
3. Provisioning Group
4. Class of Service
5. Device Type
6. System Defaults

Custom Properties

Custom properties allow for the definition of new properties, which can then be stored on any object via the API.

Custom properties are variable names defined in the RDU, and must not contain any spaces. The template parser works bottom up when locating properties in the hierarchy and converts the template option syntax. For detailed information, see Custom Properties, page 5-13.

Discovering CPE Parameters

Cisco BAC is enabled to read CPE parameters using Remote Procedure Calls (RPCs) as defined in CWMP. This is made possible through the Data Synchronization Instruction. This instruction discovers data from the CPE device, reports it to the RDU, and keeps the RDU up-to-date when CPE device data changes. This instruction can be used to keep the RDU up-to-date on such key parameters as the software version and the model name. These parameters may, in turn, be used in generating other instructions, such as configuration instructions specific to a given device type.

Table 4-2 lists the default parameters that Cisco BAC discovers.

Table 4-2 Default Discovered Parameters

Parameter	Description
Inform.DeviceId.Manufacturer	Identifies the manufacturer of the CPE.
Inform.DeviceId.ManufacturerOUI	Identifies the unique identifier of the CPE manufacturer.
Inform.DeviceId.ProductClass	Identifies the product or class of product over which the manufacturer's *SerialNumber* parameter is unique.
InternetGatewayDevice.DeviceInfo. HardwareVersion	Identifies the hardware version of the CPE.
InternetGatewayDevice.DeviceInfo. SoftwareVersion	Identifies the software version currently installed on the CPE.
InternetGatewayDevice.DeviceInfo. ModelName	Identifies the model name of the CPE.
InternetGatewayDevice. ManagementServer.ParameterKey	Specifies the value of the *ParameterKey* from the most recent SetParameterValues, AddObject, or DeleteObject call from the server.

These parameters can be updated via the API (using the /server/acs/discover/parameters property) or via the administrator user interface (see Discovering Data from Devices, page 12-9).

Note Even if the device has a different root object, such as Device, instead of InternetGatewayDevice, the parameters are still discovered.

Instruction Generation and Processing

Instruction generation is the process of generating specific instruction sets for CWMP devices. By using technology extensions, through which device technologies are incorporated into Cisco BAC, device details are combined with provisioning rules to produce instruction sets appropriate to the CPE. These instructions are then forwarded to the DPEs in the device's provisioning group and cached there.

When a device is activated in a Cisco BAC deployment, it initiates contact with the Cisco BAC server. Once contact is established, the device's preconfigured policy, based on configuration templates or firmware rules templates associated with the device, determine the management actions undertaken by the DPE. This preconfigured policy determines the device's level of service, also known as Class of Service. Device configurations can include customer-required provisioning information, such as authentication information, periodic inform rate, and Class of Service. This authoritative provisioning information for the device is forwarded to DPEs from the RDU as device configuration instructions.

Instructions are logical operations which the DPE autoconfiguration server (ACS) executes for a specific device. The instructions may map directly into a CWMP remote procedure call (RPC), for example, GetParameterValues; or, they may combine additional logic with multiple CWMP RPCs, such as firmware rules.

IGS could be controlled using the new API property, *ApiCommandKeys.IGS_ENABLE*. This property determines whether IGS should be triggered or not. The API command uses this property to control the regeneration of device configuration being affected by other API commands. When this property is set to false in the map of associated properties of an API command, the command skips the regeneration of the configuration process and it returns immediately with a warning message. The API commands affected by this property are:

- *replaceFile*
- *changeClassOfServiceProperties*
- *changeProvGroupProperties*
- *changeNodeProperties*
- *changeDefaults*

For the list of other API commands see Non-concurrent Commands, page 18-2.

The RDU requests that instructions be processed, by passing "InstructionRecords" to the DPE. The DPE server then converts these "InstructionRecords" to "Instructions", and returns the results to the RDU as "InstructionResponseRecords".

Cisco BAC generates instructions through:

- The Instruction Generation Extension—Generates "InstructionRecords" for a single device.
- The Instruction Generation Service—Generates "InstructionRecords" for more than one device.

  You can access statistics from the Instruction Generation Service from the administrator user interface at **Servers > RDU > View Regional Distribution Unit Details**.

Among the various instructions that the RDU generates are:

- Data Synchronization Instruction (DataSyncRecord)—Keeps the RDU up to date on various CPE parameters, such as the software version and the model name. These parameters may, in turn, be used in generating other instructions, such as configuration instructions that are specific to a given device type. Some parameters are checked on every connection, while others are checked only when a change in the firmware version occurs. For details, see Discovering Data from Devices, page 12-9.
- Routable IP Address Instruction (RoutableIPAddressRecord)—Discovers if a particular device is reachable, enabling the DPE to create a TCP connection with a device in order to service a connection request. The instruction retrieves the WAN IP address for the device, PPP or DHCP, and compares it with the source IP address. If the IP addresses are different, the instruction updates the RDU.
- Firmware Rules Instruction (FirmwareRulesRecord)—Determines the firmware image to be downloaded to a device. The firmware image files are associated to groups of devices by a firmware rules template. Cisco BAC uses the rules in the associated template to evaluate the firmware to be downloaded to the CPE. This instruction takes effect only if the device has been associated with a firmware rules template.
- Configuration Synchronization Instruction (ConfigSyncRecord)—Triggers a synchronization of the CPE configuration that is stored in the DPE cache. This instruction comes into effect only if the device has been associated with a configuration template. The process of configuration synchronization is explained in the subsequent section.

When the Signed Configuration feature is enabled, the RDU parses the ToBeSigned tag specified in the configuration template and sets the corresponding value on the CWMP parameter object.

By default, the ToBeSigned flag on the CWMP parameter object is set to False. For more information on Signed Configuration, see Signed Configuration for Devices, page 13-19.

Device Configuration Synchronization

During the process of CPE configuration synchronization, a device's configuration is automatically synchronized based on the configuration template associated with the device's Class of Service object. The process of synchronizing the CPE configuration according to the ConfigSync instruction stored in the DPE for this device is called Configuration Synchronization.

As part of this process, the DPE configures all parameters values and attributes found in the configuration template associated with a device via its Class of Service, so that:

- Notifications reflect those configured in the configuration template. For more information on Notifications, see Notification, page 5-7.
- Access control for all parameters reflects that configured in the configuration template. For more information on access control, see Access Control, page 5-8.

CPE configuration is associated with a unique configuration key, as defined in the TR-069 specification. This configuration key is saved in the DPE database, and is used as the *ParameterKey* parameter in RPCs that are forwarded to the CPE.

Every time the CPE establishes a connection with the DPE, the device reports the value for the *ParameterKey*—in the form of a configuration revision number—by using the Inform message that the device forwards to the DPE. The DPE compares this value with the one in its cache for the particular device. A mismatch of the values triggers the DPE and the device synchronization process.

During the process of configuration synchronization:

1. The DPE receives a *ParameterKey* from the device. If the value of this *ParameterKey* matches the one stored in the DPE, no synchronization is initiated. If the *ParameterKey* values differs, the synchronization process continues.
2. If access control is set in its configuration, the DPE sets the *AccessList* parameter to ACS-only. The access control feature is, by default, enabled. For more information on access control, see Access Control, page 5-8.
3. If you enable the notification feature, the DPE sets notification attributes as specified in the device configuration. Notifications are, by default, enabled. For more information on notifications, see Notification, page 5-7.
4. After the DPE configures the parameter values on the device according to the template, it sets a new configuration revision number in the *ParameterKey* argument. This revision number is used to determine if the device configuration is synchronized the next time the device and the DPE establish a connection.

   Note If the device connection with the DPE times out during the synchronization process, the CPE attempts to reconnect to the DPE. In this scenario, the value of the *ParameterKey* in the Inform message remains the same, because only a successful synchronization process changes the *ParameterKey* value. When the CPE reconnects to the DPE, the DPE initiates another round of synchronization with the original *ParameterKey* value.

5. The synchronization process ends with the DPE forwarding the new value for the *ParameterKey* attribute in its last update to the CPE.

Note In some situations, you must update the device even if the *ParameterKey* on the DPE matches the one on the device.

To force a configuration synchronization:

1. From the **Devices** page, locate the device whose configuration you want to synchronize.
2. Click the **Operations** icon (🔍) corresponding to the device.
3. The Device Operations page appears. From the drop-down list under Perform Device Operation, select Force Configuration Synchronization.
4. Click **Submit**.

   The device configuration is synchronized with the DPE.

Device Deployment in Cisco BAC

A Cisco BAC deployment is divided into provisioning groups, with each provisioning group responsible only for a subset of the devices. All services provided by the provisioning group are implemented to provide fault tolerance.

Note A key principle of device management is that the RDU does not directly communicate with devices. All device interactions are delegated to DPEs in the provisioning group to which the device belongs.

Cisco BAC provides two device deployment options, which can also be used in combination:

- Preregistered—The device record is added to the RDU before the device makes initial contact with the DPE, also known as the ACS.
- Unregistered—The device makes first contact with the DPE before the device record is added to the RDU.

Preregistered Devices

In this scenario, device data is preprovisioned into Cisco BAC, and the device is associated with a specific Class of Service. The Class of Service can correspond to a service that the subscriber registered for or a default configuration.

A preregistered device is preconfigured with certain parameters specific to the service provider. These parameters are typically “burnt-in” as factory defaults.

Note If you reset the device to factory defaults, the settings on the device revert to the preburnt configuration, and the device may go through the reconfiguration process.

Device data is preregistered in Cisco BAC. This is typically done through the API; alternatively, it can be done via the administrator user interface.

Preconfiguration involves three important issues:

- The device must be able to establish network connectivity. For DSL devices, this typically involves using auto-detection of ATM PVC and using PPP for authentication. The IP address is obtained via PPP or via DHCP. Other devices typically use an existing internet connection and local DHCP for address assignment.

- CPE must contact the configuration servers of the appropriate service provider; in other words, the CPE must know the ACS URL. The ACS URL can be preburnt into the device (assigned) or discovered using DHCP from the WAN side.
- The service provider must be able to associate the CPE with a specific subscriber. This process is typically accomplished by the Operations Support Systems (OSS) application responsible for subscriber registration. Cisco BAC is updated with appropriate data to provision device configuration.

Unregistered Devices

In this scenario, no device data is prepopulated into Cisco BAC. Device data is added to Cisco BAC only when the device first contacts a Cisco BAC server.

Cisco BAC allows unregistered devices (with no preconfigured parameters) to appear on a network and gain default access. However, the lack of support for preregistering device data into Cisco BAC restricts authentication options for unregistered devices, to using mechanisms based on certificates as opposed to shared secrets. The lack of preregistered data also means that Cisco BAC has to dynamically classify the devices and determine the default configuration of a device.

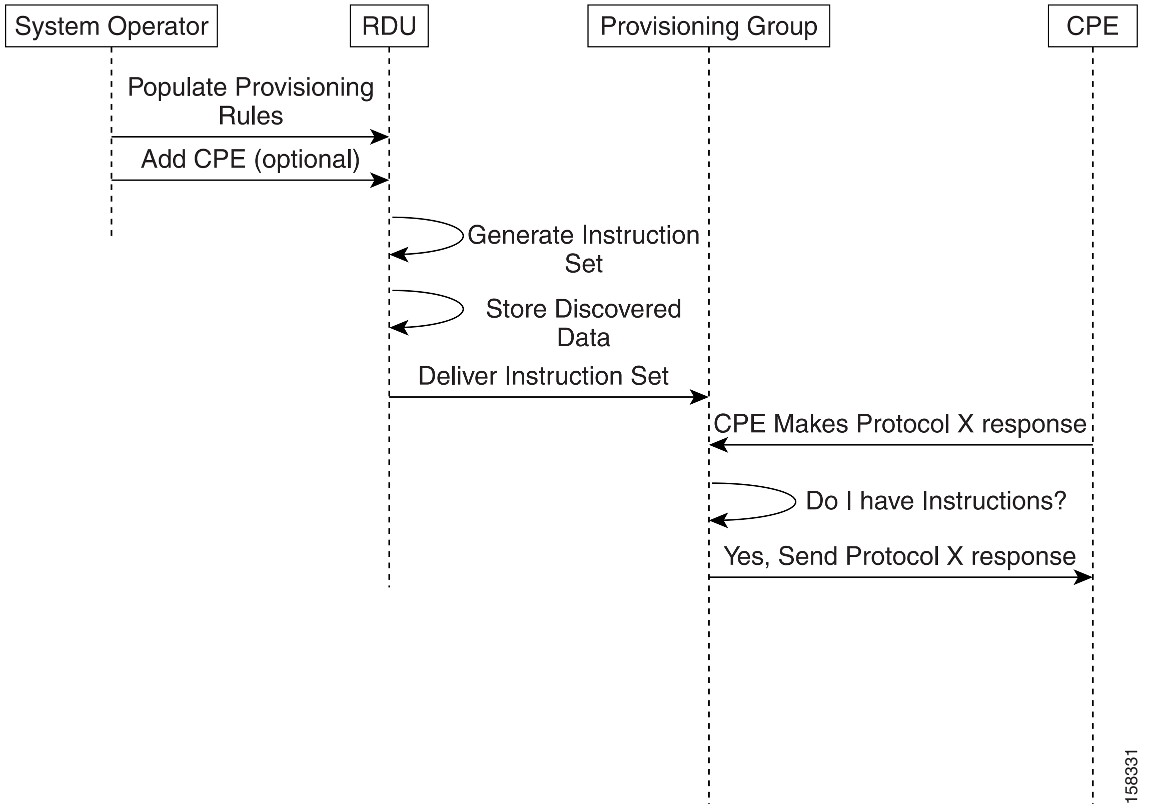
Note With no preregistered device data available, the chances of a Denial of Service attack increase, as unknown devices are not authenticated.

Initial Provisioning Flows

This section describes the configuration workflow for a device, which differs based on whether a device is preregistered or unregistered.

Figure 4-2 shows a common initial configuration flow.

Figure 4-2 Initial CPE Configuration Workflow



For Preregistered Devices

a. From the Cisco BAC API, the RDU is populated with specifically defined configurations and rules for various types of devices. The device is preconfigured and associated with a Class of Service.

b. The preregistered device finds its provisioning group by contacting the Cisco BAC server at a preconfigured URL, and initiates autoprovisioning with provisioning group servers (DPEs).

c. The RDU generates instructions appropriate for the device. The resulting device instructions direct DPE responses to various CPE protocol events, such as a TR-069 Inform and an HTTP file request.

d. The device instruction set is forwarded to the DPE and cached there. Now, the DPE is programmed to handle subsequent CPE protocol interactions for this device autonomously from the RDU. Once the device is added to the network and a configuration is generated for the device, the device boots to allow the DPE to begin its interactions with the preregistered device.

e. During interactions with the device, additional information can be discovered and forwarded to the RDU. In this case, the RDU may decide to generate new instructions and forward them to all DPEs.

For Unregistered Devices

a. From the Cisco BAC API, the RDU is populated with specifically defined configurations and rules for various device types.

b. During bootup, when the DPE receives a device request, it performs a local search for instructions cached for the specific CPE. Because the CPE has never previously contacted the DPE and because device data was not pre-registered into Cisco BAC, no instructions are found. The DPE then packs all relevant CPE information into an “instruction set” generation request, and forwards the request to the RDU. At the same time, the device request is rejected, forcing the device to retry.

c. The RDU generates instructions appropriate for the CPE and distributes it to all DPEs within the device’s provisioning group. The resulting CPE instructions direct DPE responses to various CPE protocol events, such as a TR-069 Inform, and an HTTP file request.

d. The device instruction set is delivered to the DPE and cached there. Now, the DPE is programmed to handle all subsequent CPE protocol interactions for this device autonomously from the RDU.

The following parameters are discovered by Cisco BAC for unknown devices:

- Whether the device IP address is routable or NAT’ed.
- Inform.DeviceId.Manufacturer.
- Inform.DeviceId.ManufacturerOUI.
- Inform.DeviceId.ProductClass.
- InternetGatewayDevice.DeviceInfo.HardwareVersion.
- InternetGatewayDevice.DeviceInfo.SoftwareVersion.
- InternetGatewayDevice.DeviceInfo.ModelName.
- InternetGatewayDevice.ManagementServer.ParameterKey.

Note You can change the default list of discovered parameters. See Discovering Data from Devices, page 12-9.

e. The device then reconnects, and receives the configuration instructions generated for it from the RDU and cached at the DPE.

Assigning Devices to Provisioning Groups

Devices can be assigned to a provisioning group in three ways: explicitly, automatically, or using a combination of both.

Explicit Assignment

You can explicitly assign a device to a provisioning group. Once devices show up in the default provisioning group, the provisioning system may, via the API, assign the device to a new provisioning group. On next contact with the device, Cisco BAC redirects the device.

To set the device to contact the assigned provisioning group, change the URL of the Cisco BAC server to the URL of the provisioning group URL. The Cisco BAC server URL is stored, and from then on, the device contacts Cisco BAC at the new address.

To move the device from one provisioning group to another, change the home provisioning group of the device via the API or the administrator user interface. Each provisioning group has a URL associated with it. To facilitate the move, on next contact, the ACS URL on the device is changed to the new provisioning group URL.

Automatic Membership

If a device has not been explicitly assigned to a provisioning group, the device stays in the provisioning group in which it is brought up. This allows a network-directed assignment of CPE to the provisioning groups. You can use the automatic membership feature for roaming devices, enabling the local provisioning group to service these devices when they are moved.

When a device appears in a new provisioning group, it is automatically assigned to the new provisioning group, and the device data is purged from the old provisioning group. This process involves communicating with the RDU, which, in turn, updates the DPEs in the old and new provisioning groups. This is an expensive process; therefore, take care to prevent a large number of device migrations.

Automatic assignment of a device to a provisioning group works only if the DPEs are configured to allow unknown (unregistered) devices that do not show up in any provisioning group.

If the devices do show up in another provisioning group and the provisioning group is configured to allow access for unknown devices, Cisco BAC automatically assigns the device to the provisioning group.

For details on how to configure access for unknown devices, see the *Cisco Broadband Access Center DPE CLI Reference 3.6*.

Combined Approach

You can explicitly assign devices and allow automatic membership of devices to a provisioning group. For example, a generic unregistered device that appears on the network in a provisioning group is automatically assigned to it. Subsequently, the OSS explicitly assigns the device to another provisioning group by using the API.

Device Diagnostics

CWMP supports device troubleshooting and diagnostics features that you use to focus on a single device and collect diagnostics information for further analysis. This feature enables you to query devices for any data, including:

- Configuration
- Live statistics
- Fault indications
- Log file
- Diagnostics results

Device diagnostics is made possible in Cisco BAC through a set of operations that can be executed on the device. These include:

- Reboot—Reboots the device. This reboot is primarily intended for diagnostic purposes.
- Request Connection—Initiates a connection request with Cisco BAC.

- Factory Reset—Resets a preregistered device settings to its original factory settings, to before a subscriber-specific configuration was burnt in.
- Display Live Data—Views device parameters directly from a device. You can define the parameters you want to appear.
- Ping Diagnostic—Enables you to perform an IP ping diagnostics test from the device to any host.
- Force Firmware Upgrade—Forces a CPE to update its firmware.
- Force Configuration Synchronization—Enables you to force an individual CPE to synchronize its configuration.

For details on performing these device operations, see Performing Operations on Device, page 16-15.

Cisco BAC also provides the following features to aid in troubleshooting:

- Device History—Provides a detailed history of significant events that occur in a device provisioning lifecycle. See Device History, page 8-1.
- Device Faults—Detects devices with recurring faults, which can cause bottlenecks and affect network performance. See Device Faults, page 8-6.
- Device Troubleshooting—Provides detailed records of device interactions with Cisco BAC servers for a set of devices that are designated for such troubleshooting. See Device Troubleshooting, page 8-9.
- Performance Statistics—Provides detailed performance statistics that are related to system performance across major components. It also provides an analysis of the statistical data to aid troubleshooting. See Monitoring Performance Statistics, page 11-14.