



Configuring CWMP Service

This chapter describes how to configure the CWMP service in Cisco Broadband Access Center (BAC).

This chapter includes the following sections:

- [CWMP Service Configuration, page 12-1](#)
- [Connection Request Service, page 12-2](#)
- [Provisioning Group Scalability and Failover, page 12-16](#)

CWMP Service Configuration

CWMP is a specification of a set of remote procedure calls (RPCs), for example, `GetParameterValues`, `SetParameterValues`, and so on. These RPCs define the generic mechanism by which Cisco BAC reads or writes parameters to customer premises equipment (CPE) in order to manage it. These parameters include:

- Device configuration information
- Status information
- Performance statistics

You can enable or disable CWMP features on the DPE by using the DPE CLI.

Among the features that you can configure on the DPE are:

- HTTP-based Basic or Digest authentication
- Certificate-based authentication
- HTTP over SSL/TLS service settings
- Handling of unknown devices
- Debugging settings
- Session management settings
- CWMP service settings
- HTTP file service settings

For information on how to configure these properties, see the [Cisco Broadband Access Center 3.8 DPE CLI Reference](#).

Configuring Service Ports on the DPE

You can configure the ports on which the CWMP services communicate with a device. You can independently configure each instance of the CWMP services—the CWMP RPC service and the HTTP file service—to suit your requirements. [Table 12-1](#) describes how you configure ports for each service.

Table 12-1 Configuring Service Ports

Command	Syntax Description	Default
Configuring the CWMP RPC Service		
<code>service cwmp num port port</code>	<ul style="list-style-type: none"> <code>num</code>—Identifies the CWMP service, which could be 1 or 2. <code>port</code>—Identifies the port number that the service should use. <p>Example:</p> <pre>dpe# service cwmp 1 port 7547 % OK (Requires DPE restart "# dpe reload")</pre>	By default, the CWMP service is configured to listen on: <ul style="list-style-type: none"> Port 7547 for service 1. Port 7548 for service 2.
Configuring the HTTP File Service		
<code>service http num port port</code>	<ul style="list-style-type: none"> <code>num</code>—Identifies the HTTP file service, which could be 1 or 2. <code>port</code>—Identifies the port number that the service should use. <p>Example:</p> <pre>dpe# service http 1 port 7549 % OK (Requires DPE restart "# dpe reload")</pre>	By default, the HTTP file service is configured to listen on: <ul style="list-style-type: none"> Port 7549 for service 1. Port 7550 for service 2.

For more configuration instructions, see the [Cisco Broadband Access Center 3.8 DPE CLI Reference](#).

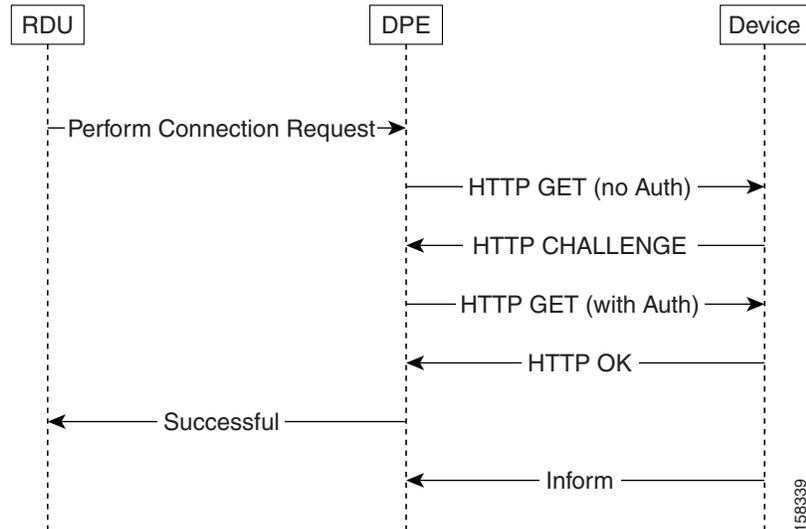
Connection Request Service

Connection requests instruct the device to establish a CWMP session with the DPE. You can use the Cisco BAC connection request service to activate a configuration on a device, execute firmware changes to the device, or execute immediate operations upon the device.

Initiated by the DPE, connection requests are the only method available to the DPE to establish a session with a device. Once a session is established, the device or the DPE may perform any RPCs, including device operations and configuration changes.

[Figure 12-1](#) describes the flow of a connection request in Cisco BAC. The RDU delegates the connection request to the best available DPE in the provisioning group of the device. Once the connection request ends, the DPE notifies the RDU of the result.

Figure 12-1 Connection Request in BAC



Configuring Connection Request Options

You can use Cisco BAC to control the behavior of connection requests by configuring your preferences for:

- [Configuring Authentication](#)
- [Configuring Connection Request Methods](#)
- [Configuring Reachability](#)



Note

You can set your preferences on the device object or in its property hierarchy.

Configuring Authentication

Two properties that you set on the device object in the RDU affect authentication. They are:

- On the API:
 - `IPDeviceKeys.CONNECTION_REQUEST_USERNAME`
 - `IPDeviceKeys.CONNECTION_REQUEST_PASSWORD`
- On the administrator user interface:
 - **Devices > Modify Device > Connection Request User Name** field
 - **Devices > Modify Device > Connection Request Password** field

You can also set the connection request username and password while adding a device on the Add Device page, and change the username and password in the Modify Device page.

Both properties control the connection request username and password that are used in DPE-CPE authentication. This username and password differ from the username and password used to authenticate CWMP sessions between the DPE and a device. These properties are for a single device; thus, they can be set only on the device object.

If you do not specify a Connection Request password, a Connection Request password is automatically generated for the device using the connection request master secret. If you do not specify the Connection Request username, the device ID is used.

It is up to the device to issue an authentication challenge during connection request authentication, as illustrated in [Figure 12-1](#). The DPE expects to be challenged with HTTP Digest authentication. There is no DPE configuration for connection request handling.

The API properties do not automatically update device parameters. You must preconfigure the corresponding values on the device or configure the values using a configuration template which can reference these properties.

Autogenerating Connection Request Passwords

In this release of Cisco BAC, Connection Request passwords can be autogenerated or specified by the Operational Support System (OSS).

Cisco BAC Generated Passwords:

In this approach, Cisco BAC generates a unique Connection Request password for each CWMP device. The password is encrypted using the connection request master secret and forwarded to the DPEs. You specify the connection request master secret in the CWMP Defaults page in the administrator user interface (see [CWMP Defaults, page 17-8](#)).

The DPEs derive the device passwords by using the hash message authentication code. If the DPE fails to authenticate using the current password, Cisco BAC attempts to authenticate by using the old password derived from the earlier master secret. Cisco BAC stores the last 15 passwords, by default, and attempts authentication by using each of these passwords in reverse order, until authentication succeeds.

To use autogenerated passwords, you have to specify the value, `__AUTO_GENERATED__` for Connection Request password in the configuration template.

When the RDU attempts a connection request to a device:

- If `/IPDevice/connectionRequestPassword` property is not specified in the device record, the RDU assumes that an autogenerated password is used for that device.
- If `/IPDevice/connectionRequestUsername` property is not specified in the device record, the RDU uses the device ID as the user name for that device.

OSS Provided Passwords:

In this approach, you can provide passwords that may or may not be unique to each device. The Connection Request password is set on the `/IPDevice/connectionRequestPassword` property in the device record. When the password is set on the property, the following changes are triggered in the system:

- This RDU-provided password is used as the Connection Request password, instead of the autogenerated password at the DPE. As a rule, the password set on the device takes precedence over the autogenerated password.
- The hash key for the device reverts to the legacy format (*DeviceConfHash*).

You can also set the Connection Request username and password in the device configuration templates.

- If the username and password are set on both the device record and the configuration template, the username and password set in the device record are used.

- If the username and password are present only on the configuration template (and not in the device record), this password is used instead of the auto-generated password.

Configuring Connection Request Methods

You can specify the method in which Cisco BAC attempts to perform a connection request by using the provisioning API or the administrator user interface. The selected method dictates how Cisco BAC determines the connection request URL to be used to contact the device.

The API property `IPDeviceKeys.CONNECTION_REQUEST_METHOD` specifies the connection request method (subsequent descriptions provide the details of each method).



Note

You can specify this property anywhere in the device hierarchy.

To configure the default connection request method by using the administrator user interface, choose **Configuration > Default > CWMP Defaults**, and select an option from the drop-down list.

Cisco BAC supports the following methods to configure a connection request:

- Discovered
- Using FQDN
- Using IP
- LeaseQuery
- Annex G
- Use CMHS

When selecting a connection request method, it is important to consider performance and manageability, as each method provides different levels of both.

Discovered Connection Request

The Discovered method, during CPE interactions with the DPE, modifies the data synchronization instruction to discover the device's connection request URL, which corresponds to the parameter `InternetGatewayDevice.ManagementServer.ConnectionRequestURL`, whenever the DPE interacts with the device. The RDU records any updates to this parameter value and uses it when making connection requests.



Note

This value must be discovered before connection requests are attempted.

Because this parameter value changes each time the device's WAN IP address changes and every update has to be stored at the RDU, it is not the optimal method for connection requests.

Use FQDN Connection Request

The Use FQDN method uses the fully qualified domain name (FQDN) specified for the device at the RDU to construct a connection request URL for the device. It uses the FQDN along with the values specified in the following properties on the API:

- `IPDeviceKeys.CONNECTION_REQUEST_PORT`
- `IPDeviceKeys.CONNECTION_REQUEST_PATH`

You can also specify these properties on the administrator user interface:

Step 1 Choose **Devices > Manage Devices**.

Step 2 Use one of these methods:

- Add a device record. To do this, click the **Add** button. The Add Device page appears.
- Search for a device record. To do this,
 - a. Specify a Search Type and enter values for the screen components that are unique to the search type that you select.

A list of devices appears.

- b. Click the Identifier link corresponding to the desired device.

The Modify Device page appears.

Step 3 From the Property Name drop-down list, select the `/IPDevice/connectionRequestPort` and `/IPDevice/connectionRequestPath` properties, and enter appropriate values in the Property Value field.



Note The API constants for `/IPDevice/connectionRequestPort` and `/IPDevice/connectionRequestPath` are `IPDeviceKeys.CONNECTION_REQUEST_PORT` and `IPDeviceKeys.CONNECTION_REQUEST_PATH`, respectively.

Step 4 Click **Add**.

You can specify the port and path properties anywhere in the property hierarchy.



Note Because the Use FQDN method relies on the DNS being updated with the device's correct IP address, you do not need to update Cisco BAC whenever the device IP address changes. Subsequently, this option is the most scalable one for connection requests.

Use IP Connection Request

The Use IP method discovers the device's WAN IP address by using the same mechanism as the Discovered method. Then, it constructs a connection request URL for the device by using the values in the following API properties:

- `IPDeviceKeys.CONNECTION_REQUEST_PORT`
- `IPDeviceKeys.CONNECTION_REQUEST_PATH`

You can also specify these properties on the administrator user interface:

Step 1 Choose **Devices > Manage Devices**.

Step 2 Use one of these methods:

- Add a device record. To do this, click the **Add** button. The Add Device page appears.
- Search for a device record. To do this,
 - a. Specify a Search Type and enter values for the screen components that are unique to the search type that you select.

A list of devices appears.

- b. Click the Identifier link corresponding to the desired device.

The Modify Device page appears.

- Step 3** From the Property Name drop-down list, select the `/IPDevice/connectionRequestPort` and `/IPDevice/connectionRequestPath` properties, and enter appropriate values in the Property Value field.



Note The API constants for `/IPDevice/connectionRequestPort` and `/IPDevice/connectionRequestPath` are `IPDeviceKeys.CONNECTION_REQUEST_PORT` and `IPDeviceKeys.CONNECTION_REQUEST_PATH`, respectively.

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

Because the Use IP method relies on the RDU having the device's WAN IP address, it requires the WAN IP address to be discovered before connection requests are attempted. Also, because the device's WAN IP address changes, the RDU must be updated with the new IP address. Subsequently, this option is not the optimal method for connection requests.

Using LeaseQuery

The LeaseQuery method discovers the latest IPsec address of the HNB from Cisco Network Registrar DHCP server. It first forms the Connection request URL by using the following API properties and then sends it to HNB through SeGW (Security Gateway).

- `IPDeviceKeys.CONNECTION_REQUEST_PORT`
- `IPDeviceKeys.CONNECTION_REQUEST_PATH`

You can also specify these properties on the administrator user interface:

- Step 1** Choose **Devices > Manage Devices**.

- Step 2** Use one of these methods:

- Add a device record. To do this, click **Add**. The Add Device page appears.
- Search for a device record. To do this,
 - a. Specify a Search Type and enter values for the screen components that are unique to the search type that you select. A list of devices appears.
 - b. Click the Identifier link corresponding to the desired device. The Modify Device page appears.

- Step 3** From the Property Name drop-down list, select the `/IPDevice/connectionRequestMethod`, `/IPDevice/connectionRequestPort` and `/IPDevice/connectionRequestPath` properties, and enter the appropriate values in the Property Value field.



Note The API constants for `/IPDevice/connectionRequestMethod`, `/IPDevice/connectionRequestPort` and `/IPDevice/connectionRequestPath` are `IPDeviceKeys.CONNECTION_REQUEST_METHOD`, `IPDeviceKeys.CONNECTION_REQUEST_PORT`, and `IPDeviceKeys.CONNECTION_REQUEST_PATH`, respectively.

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

Configuring STUN Parameters (Annex G)

Cisco BAC now includes the TR-069 Annex G support for remotely managing CPE devices which are connected through a NAT gateway. Annex G uses a STUN server to identify a CPEs public IP address and port details.

The STUN server also tracks all the changes being made to the CPEs. Annex G uses a UDP based connection request mechanism defined in TR069 Annex G to allow Cisco BAC to initiate a session with the CPEs.

Step 1 Choose **Devices > Manage Devices**.

Step 2 Use one of these methods:

- Add a device record. To do this:
Click the Add button. The Add Device page appears.
- Search for a device record. To do this,
 - a. Specify a Search Type and enter values for the screen components that are unique to the search type that you select.
A list of devices appears.
 - b. Click the Identifier link corresponding to the desired device.
The Modify Device page appears.

Step 3 From the Property Name drop-down list, select the */IPDevice/connectionRequestPort* and */IPDevice/connectionRequestPath* properties, and enter appropriate values in the Property Value field.

Configure the following properties at the property hierarchy level:

- */IPDevice/cr/stunEnable*
- */IPDevice/cr/stunServerAddress*
- */IPDevice/cr/stunServerPort*
- */IPDevice/cr/stunServerHttpPort*
- */IPDevice/cr/stunServerHttpUsername*
- */IPDevice/cr/stunServerHttpPassword*

The API constants for */IPDevice/cr/stunEnable*, */IPDevice/cr/stunServerAddress*, */IPDevice/cr/stunServerHttpPort*, */IPDevice/cr/stunServerPort*, */IPDevice/cr/stunServerHttpUsername*, and */IPDevice/cr/stunServerHttpPassword* are:

`IPDeviceKeys.STUN_ENABLED`, `IPDeviceKeys.STUN_SERVER_ADDRESS`,
`IPDeviceKeys.STUN_SERVER_PORT`, `IPDeviceKeys.STUN_SERVER_HTTP_PORT`,
`IPDeviceKeys.STUN_HTTP_USER_NAME`, and `IPDeviceKeys.STUN_HTTP_PASSWORD`, respectively.

Step 4 Click **Add**.

You can specify the port and path properties anywhere in the property hierarchy.

Configuring CMHS Connection Request (Use CMHS)

This method allows Cisco BAC to send connection request to CPE devices using the CMHS server, using BAC NB API Interfaces or BAC Admin UI.

When RDU receives the connection request message, it sends the message to the CMHS server that is configured in the BAC properties hierarchy. The CMHS NB API client contacts the CMHS server that the DLC is most likely be connected to.

-
- Step 1** Choose **Devices > Manage Devices**.
- Step 2** Use one of these methods:
- Add a device record. To do this:
Click Add. The Add Device page appears.
 - Search for a device record. To do this:
 - a. Specify a Search Type and enter values for the screen components that are unique to the search type that you select.
A list of devices appears.
 - b. Click the Identifier link corresponding to the desired device.
The Modify Device page appears.
- Step 3** From the Property Name drop-down list, select the `/IPDevice/connectionRequestMethod` property, and select **Use CMHS** from the Property Value drop-down list.
- Step 4** Click **Add**.
- Step 5** Create a group type called CMHS. See [Adding a Group Type, page 16-19](#) for details on how to add a new group type.
- Step 6** Add a group with type CMHS for each CMHS server which needs to be considered to send CMHS connection request. See [Adding a New Group, page 16-20](#) for details on how to add a new group.
- Step 7** Configure the following properties with CMHS server details, host, http port, user name and password, in the BAC properties hierarchy at group level.
- `/IPDevice/cr/cmhsNbApiAddress`
 - `/IPDevice/cr/cmhsNbApiPort`
 - `/IPDevice/cr/cmhsNbApiUsername`
 - `/IPDevice/cr/cmhsNbApiPassword`



Note CMHS groups must not be used for any other purpose other than configuring CMHS server details for CMHS connection requests.

If you do not specify any values for the properties `/IPDevice/cr/cmhsNbApiUsername` and `/IPDevice/cr/cmhsNbApiPassword`, the default values are taken from `cmhs.properties` file available in the `BAC_HOME/rdu/conf/` folder. For the property `/IPDevice/cr/cmhsNbApiPort`, the default value is 80.



Note You can use the API `IPDevice.changeNodeProperties()`, if want to change any of the BAC properties for the CMHS group.

- Step 8** Configure CMHS Server List Property, `/IPDevice/cr/cmhsServerListPropertyName` with a deployment specific Custom Property name. It needs to be set on BAC property hierarchy using BAC NB API or UI.

This Custom Property is not pre-defined by BAC. It needs to be created and populated with comma separated values of CMHS groups names. These values need to be considered while sending CMHS connection request.



Note The API constants for /IPDevice/connectionRequestMethod, is `IPDeviceKeys.CONNECTION_REQUEST_METHOD`.

Configuring External URLs on DPE

The ACS URL is the configured URL of the Cisco BAC server associated with each provisioning group. The devices use this URL to contact the DPEs in a given provisioning group. The RDU uses this URL to perform various operations, such as redirecting the devices to the home provisioning group.

The ACS URL that the DPE issues, depends on the URL that is configured for each of the CWMP services. You can use the following commands to configure the external URL for each service running on the DPE:

Command	Description
<code>service cwmp 1 external-url url</code>	Configures the DPE to represent externally the specified external URL as the URL of the CWMP service 1.
<code>service cwmp 2 external-url url</code>	Configures the DPE to represent externally the specified external URL as the URL of the CWMP service 2.
<code>service http 1 external-url url</code>	Configures the DPE to represent externally the specified external URL as the URL of the HTTP service 1.
<code>service http 2 external-url url</code>	Configures the DPE to represent externally the specified external URL as the URL of the HTTP service 2.

For specific information about using these commands, see the [Cisco Broadband Access Center 3.8 DPE CLI Reference](#).

For CWMP services, the URL of the CWMP service that is first enabled on the DPE is sent to the RDU as the discovered ACS URL.

Disabling Connection Requests

You can choose to disable the connection request service. This option is useful if there is no STUN server and a device is behind NAT setup. Connection requests are not possible in such scenarios.



Note You can use ConnectionRequest via API device operations even if you disable connection requests.

Configuring Reachability

Reachability plays an important role in configuring connection requests. In earlier versions of Cisco BAC, connection requests were refused if a device's reported IP address and its source IP address did not match. This was because this implies that the device used the NAT standard, and therefore, configuration requests did not normally succeed. You could change this behavior from the administrator user interface or the API.

Cisco BAC supports a new standalone STUN server to handle the UDP connection requests feature. A UDP based connection request mechanism defined in TR069 Annex G, allows Cisco BAC to initiate a session with a CPE that is operating behind a NAT gateway. STUN Service can be run on Solaris or Linux.

By using the API, set the property `IPDeviceKeys.FORCE_ROUTABLE_IP_ADDRESS` to *true* to allow TCP based connection requests regardless of a mismatch in the device source IP address.

From the administrator user interface:

Step 1 Choose **Devices > Manage Devices**.

Step 2 Use one of these methods:

- Add a device record. To do this:
Click the Add button. The Add Device page appears.
- Search for a device record. To do this,
 - a. Specify a Search Type and enter values for the screen components that are unique to the search type that you select.
A list of devices appears.
 - b. Click the Identifier link corresponding to the desired device.
The Modify Device page appears.

Step 3 From the Property Value drop-down list, select `/IPDevice/forceRouteIPAddress`, and set a property value.

The API constant for `/IPDevice/forceRouteIPAddress` is `IPDeviceKeys.FORCE_ROUTABLE_IP_ADDRESS`.

Step 4 Click **Add**.



Note You can specify this property anywhere on the device's hierarchy.

Discovering Data from Devices

This section describes the Discovery of Data feature, which you use to retrieve a predefined set of parameters from the device, and store these parameters in the RDU for future use.

You can use this discovered data to manage device firmware and device configurations by providing some key attributes of the device and its current configuration. Discovered parameters are updated at the RDU any time their values change on the device.

You can configure data discovery at the RDU. The RDU forwards discovery policy instructions and the values of parameters, discovered for each device during the previous discovery process, to the appropriate DPEs. During interactions with the device, the DPE consults the discovery policy instructions specific to the device to determine what parameters need to be discovered.

After parameter values are discovered, existing device parameters are compared with those already stored for the device. If the values have changed or are being obtained for the first time, this data is updated at the RDU. If the RDU is not available to receive the update, the newly discovered data is dropped, and the entire process of data discovery is initiated the next time the device connects with the DPE.

Discovered parameters are stored on device records and you can view them by using the administrator user interface or obtain them through the API `IPDevice.getDetails()` call. `IPDevice-getDetails()` API also retrieves the device data along with the device property hierarchy. To view discovered parameters using the user interface, access the Manage Devices page under the **Devices** tab on the primary navigation bar.

Locate the device by using the search options, and click the **View Details** icon () corresponding to the device. The Device Details page appears, displaying details of the discovered parameters for the device.

Multi-instance object support is available in data discovery for preregistered and unknown devices. For more information on multi-instance object support, see [Multi-Instance Object Support, page 4-5](#).

The following sections describe:

- [Configuring Data Discovery, page 12-12](#)
- [Troubleshooting Data Discovery, page 12-13](#)
- [Automatic IMEI Update, page 12-15](#)

Configuring Data Discovery

Data discovery policy is configured using the RDU and includes parameters checked:

- On every device contact.
- Only upon a firmware upgrade.

While the discovery of data process executes every time a device establishes contact with the DPE, you can configure validation of certain parameters to occur only if the device has reported a new firmware version. This check eliminates the need to validate parameters whose values change only with a firmware upgrade.

For example, the device model name does not change without a firmware upgrade; thus, you do not need to check this parameter on the device unless the firmware has been upgraded.

Cisco BAC ships with a default configuration for data discovery. You can augment this default configuration in two ways:

- Add parameters to a custom list.
- Change the default list. This option, however, is not recommended.

Configuring Parameters Checked on Every Contact

You can configure the default list of parameters discovered everytime the device connects with the DPE by using the `ServerDefaultsKeys.CWMP_DISCOVER_PARAMETERS` property. You can add more parameters to the default list by providing a comma-separated list of parameters as a value to the `IPDeviceKeys.CWMP_CUSTOM_DISCOVER_PARAMETERS` property.

The default list includes:

- Inform.DeviceId.Manufacturer
- Inform.DeviceId.ManufacturerOUI
- Inform.DeviceId.ProductClass
- InternetGatewayDevice.DeviceInfo.HardwareVersion
- InternetGatewayDevice.DeviceInfo.SoftwareVersion
- InternetGatewayDevice.ManagementServer.ParameterKey

For example:

```
IPDeviceKeys.CWMP_CUSTOM_DISCOVER_PARAMETERS= InternetGatewayDevice.ManagementServer.URL,
InternetGatewayDevice.ManagementServer.PeriodicInformEnable
```

Configuring Parameters Checked on Firmware Upgrade

You can configure the default list of parameters discovered after every firmware upgrade by using the `ServerDefaultsKeys.CWMP_FIRMWARE_CHANGED_CPE_PARAMETERS` property. This default list includes the `InternetGatewayDevice.DeviceInfo.ModelName` parameter.

To customize this list to include more parameters, use the `IPDeviceKeys.CWMP_CUSTOM_FIRMWARE_CHANGED_PARAMETERS` property.

Troubleshooting Data Discovery

You can also troubleshoot data discovery from the DPE CLI by using any of the tasks described in [Table 12-2](#).

Table 12-2 *Troubleshooting Discovery of Data from DPE*

Task	Use ...	Description
Enable info-level logging	dpe# log level 6-info	Displays information messages

Table 12-2 Troubleshooting Discovery of Data from DPE (continued)

Task	Use ...	Description
View device log	<pre>dpe# show log</pre> <pre>dpe# show log last 100</pre> <pre>dpe# show log run</pre> <p>Note You can use any of the three listed commands.</p>	<p>Displays recent DPE log entries.</p> <p>Displays the last 100 lines of the DPE log.</p> <p>Displays the running DPE log.</p>
<p>Example</p> <p>The output of the show log run command has been shortened for demonstration purposes.</p> <pre>dpe# show log run</pre> <p>% Press <enter> to stop.</p> <pre>2006 08 04 00:47:01 EDT: %BAC-DPE-6-0104: Obtained configuration for device [0014BF-CJJ005B00009] from RDU.</pre> <pre>2006 08 04 00:47:21 EDT: %BAC-CWMP-6-5129: Device [0014BF-CJJ005B00009]. Source IP [10.86.147.149]. Retrieving [1] discovered CPE parameters.</pre> <pre>2006 08 04 00:47:21 EDT: %BAC-CWMP-6-5107: Device [0014BF-CJJ005B00009]. Source IP [10.86.147.149]. Sent [GetParameterValues] message.</pre> <pre>2006 08 04 00:47:21 EDT: %BAC-CWMP-6-5106: Device [0014BF-CJJ005B00009]. Source IP [10.86.147.149]. Received [GetParameterValuesResponse] message.</pre> <pre>2006 08 04 00:47:21 EDT: %BAC-CWMP-6-5120: Device [0014BF-CJJ005B00009]. Source IP [10.86.147.149]. New data discovered from CPE. Queued update of [7] parameters to RDU.</pre> <p>Note Discovery of data is successful if the output of the show log commands is similar to the sample output featured here.</p>		

Table 12-2 Troubleshooting Discovery of Data from DPE (continued)

Task	Use ...	Description
Enable debugging	<pre>dpe# debug on</pre> <pre>dpe# debug service cwmp num data-sync</pre> <p><i>num</i>—Specifies the instance of the CWMP service, which could be 1 or 2.</p>	Enables debug logging of the data synchronization process for the CWMP service
View data discovery configuration for a specific device	<pre>dpe# show device-config device-id</pre> <p><i>device-id</i>—Specifies the ID of the device.</p>	Displays the device configuration cached at the DPE.

For specific information on using these commands, see the [Cisco Broadband Access Center 3.8 DPE CLI Reference](#).

**Note**

You can use the provisioning API to execute many of the operations that are possible from the administrator user interface.

Automatic IMEI Update

Cisco BAC now provides the capability to discover the International Mobile Equipment Identity (IMEI) number that is unique to every device.

After the IMEI is discovered, it is stored on Hostname field in the device record in the RDU and is used as the secondary Device ID. In the Admin GUI, the Hostname field is available in the Device Details page. The auto-discover IMEI feature functions in the same way for both known and unknown devices.

To enable IMEI update:

-
- Step 1** Configure BAC property `/IPDevice/custom/discover/parameters` with CPE IMEI parameter name.
 - Step 2** Configure BAC IMEI update property `/IPDevice/secondary-deviceID/cpe-param`, also called as Secondary Device ID CPE Parameter field with CPE IMEI parameter name.
 You can set the IMEI update property `/IPDevice/secondary-deviceID/cpe-param` property in Cisco BAC property hierarchy through Class of Service page. This property is listed under the **Property Name** drop-down list.
 - Step 3** Configure BAC property `/IPDevice/properties/available/pg` with BAC IMEI update property name `/IPDevice/secondary-deviceID/cpe-param`.
-

To disable IMEI update, Configure BAC IMEI update the `/IPDevice/secondary-deviceID/cpe-param` property with value set to empty string or **NOT_SUPPORTED**. This disables IMEI update on hostname field.

Provisioning Group Scalability and Failover

This section describes the scalability and failover features provided in this release of Cisco BAC.

Cisco BAC's scalability and failover provide a high degree of availability to suit networks of virtually any size, even those with millions of devices deployed. The product also provides such critical features as DPE redundancy and failover protection.

Scalability in a Cisco BAC deployment is accomplished through provisioning groups, each of which is a cluster of redundant DPE servers that communicate with a set of CPE. Provisioning groups enhance the scalability of the Cisco BAC network by making each provisioning group responsible only for a subset of devices.

This partitioning of devices can be along regional groupings or any other policy defined by the service provider.

To scale a deployment, the administrator can:

- Upgrade existing DPE server hardware.
- Add DPE servers to a provisioning group.
- Add provisioning groups and redistribute devices among them.

Cisco BAC supports explicit assignment and automatic membership of devices to provisioning groups. For more information, see [CPE Management Overview, page 4-1](#).

Redundancy in Cisco BAC

Redundancy helps to ensure:

- High availability for your network applications.
- Users do not experience long network delays or black holes due to a single point of failure.

Cisco BAC supports local as well as regional server redundancy.

Local Redundancy

The Cisco BAC provisioning group is a cluster of redundant DPE servers that communicate with a set of devices. A single URL identifies each provisioning group.

You can configure local redundancy by using any software or hardware that provides load-balancing features, such as the Cisco Application Control Engine (ACE) 4710.

Regional Redundancy

Regional redundancy ensures that DPEs in a different location temporarily process CPE requests in case of regional failure. The simplest way of facilitating this deployment is to set up DPEs within a provisioning group in different geographic locations. In such a setup, CPE requests are serviced by DPEs located in different regions, providing regional failover within the confines of the provisioning group.

In some deployments, however, you may require that CPE requests be processed by servers in one location and fail over to DPEs in another location only in case of failure. In such a scenario, you can also locate DPEs for a provisioning group in different regions yet configure the network such that, under normal conditions, CPE requests are directed only to a subset of the DPEs in a particular region.

Typically, you can configure regional redundancy by using DNS techniques. To configure the network for regional redundancy, ensure that the Cisco BAC hostname of a given provisioning name normally resolves to IP address(es) representing one set of DPEs.

However, when none of the DPE servers in that set responds to keepalives, such as ICMP, HTTP GET, or a TCP handshake, the DNS server should resolve the Cisco BAC hostname to the IP address(es) of the DPEs from a second set. CPE requests are then directed to a different set of DPEs.

Since DPEs in both sets belong to the same provisioning group, the new DPEs are ready to respond to requests. Subsequent DNS lookup requests fall back to the primary set of DPEs as soon as they become available.

DPE Load-Balancing

Cisco BAC supports the following mechanisms for DPE redundancy and load-balancing:

- [Using DNS Round Robin, page 12-17](#)
- [Using a Hardware Load Balancer, page 12-17](#)

Using DNS Round Robin

In using the DNS round-robin mechanism, the DNS server shuffles the list of DPE IPs when resolving the autoconfiguration server (ACS) hostname for the device. The device then takes the first IP address in the list as the ACS hostname.

This option is not recommended if the service provider does not control DNS caching servers. Also, DNS round-robinning performs less than ideally in a power outage scenario, when a large number of devices may receive the same order IP address cached by the DNS server, thus impacting performance. To work around this issue, Cisco recommends configuring a very short TTL of 1 second on the DNS server.

Using a Hardware Load Balancer

When a hardware load balancer is used, the ACS URL contains the IP address or is resolved by the DNS server to a single IP address. All DPEs for a provisioning group are hidden behind a single virtual IP address of the hardware load balancer, such as Cisco ACE 4710.

The ACE provides load balancing and SSL/TLS acceleration for TR-069 traffic between the CWMP devices and the DPEs. You can configure the ACE to translate its virtual IP address to a specific DPE IP address, based on a load-balancing algorithm. ACE performs a series of checks and calculations to determine the DPE that can best service each device request according to the load-balancing algorithm.

ACE also provides stickiness that allows the same device to maintain multiple simultaneous or subsequent connections with the same DPE for the duration of a session.

A redundant pair of load balancers can improve redundancy and may service more than one provisioning group.

Adding DPE to a Provisioning Group

This section describes how to add a DPE to a new provisioning group. In adding a DPE to a provisioning group, there are three possible options:

- Adding a DPE to a provisioning group
- Add a DPE to a provisioning group in a deployment that uses a hardware load balancer, such as Cisco ACE 4710, between the device and the DPE. In this case, you must update the load balancer.
- Adding a DPE to a provisioning group in a deployment in which a DNS server, using round robin, resolves to multiple DPEs for a provisioning group.

To add a DPE to a provisioning group:

Step 1 Configure the DPE from the DPE CLI. Among the configurations you must perform are:

- Specifying the provisioning group to which the DPE must belong. Enter:

```
dpe# dpe provisioning-group primary name
```

- *name*—Identifies the assigned primary provisioning group.

- Specifying the RDU to which the DPE connects. Enter:

```
dpe# dpe rdu-server {host | ip} port
```

- *host*—Identifies the FQDN of the host on which the RDU is running.
- *ip*—Identifies the IP address of the RDU.
- *port*—Identifies the port number on which RDU is listening for DPE connections. By default, this port number is 49187.

- Configuring the FQDN for a specific interface. The provisioning FQDN is the FQDN that is given to a device to contact the specific DPE interface. Enter:

```
dpe# interface ethernet {intf0 | intf1} provisioning fqdn fqdn
```

- *intf0 | intf1*—Identifies the Ethernet interface.
- *fqdn*—Identifies the FQDN that is set on the specified interface.

You must use the same FQDN for all DPEs in a given provisioning group. If DPEs are located behind the load balancer, use the FQDN of the load balancer as the interface FQDN, and ensure that it is the same for all DPEs which are part of the same load-balancing group.

You must also configure the CWMP service and the HTTP file service on one DPE to match the configurations on other DPEs.

For more information on configuration options, see the [Cisco Broadband Access Center 3.8 DPE CLI Reference](#). Also, see [Configuration Workflows and Checklists, page 3-1](#).

Step 2 Start the DPE by using the **dpe start** command, and allow the DPE to synchronize and populate device configuration instructions from the RDU.

Step 3 Optionally, if you are using a load balancer, add the DPE address to the load balancer.

Step 4 Optionally, if you are using the DNS round robin technique, add the DPE address to the DNS server.
