



# CWMP Device Operations

This chapter describes the device operations mechanism in Cisco Broadband Access Center (BAC). By using this mechanism, you can run CPE WAN Management Protocol (CWMP) remote procedure calls for a device and perform maintenance tasks, such as changing a device’s provisioning group.

This chapter includes the following sections:

- [Overview, page 14-1](#)
- [Connection Modes for Device Operations, page 14-2](#)
- [Managing a Device’s Provisioning Group, page 14-6](#)

## Overview

You use device operations to run individual CWMP remote procedure calls (RPCs) against a device immediately, or the next time the device connects to Cisco BAC. By using this feature from the provisioning API, you can troubleshoot, collect data, or customize the interaction between a device and Cisco BAC.

Device operations are submitted in a batch by using the `IPDevice.performOperation()` API call. For details, refer to the API Javadoc which is available at `BPR_HOME/docs/nb-api/javadoc`.

You can perform some operations from the administrator user interface. This section is focused on describing the concepts behind device operations, regardless of whether they are initiated using the API or the administrator user interface.

During a Cisco BAC installation, several sample files are copied to the `BPR_HOME/rdu/samples` directory, which contains:

- The `cwmp` directory, which features `*parameter-list.xml` files that describe parameter lists used to retrieve live device data by using the administrator user interface.
- The `nb-api` directory, which features `CwmpDiagnosticImmediateExample.java` and `CwmpDiagnosticOnConnectExample.java` files that describe how you can perform device operations by using the API.

The device operations supported in this Cisco BAC release are listed in [Table 14-1](#).

**Table 14-1** Device Operations Supported in Cisco BAC

Operation Name	Description
CreateObjectInstance	Creates object instances within a device’s data model
DeleteObjectInstance	Deletes objects from a device’s data model

**Table 14-1** Device Operations Supported in Cisco BAC (continued)

Operation Name	Description
Download	Instructs the device to perform a file download
FactoryReset	Resets a device's settings to its factory default state
Reboot	Instructs the device to reboot
GetParamAttributes	Retrieves attributes for a device's parameters
GetParamNames	Retrieves names of parameters exposed by a device
GetParamValues	Retrieves parameter values from a device's parameter space
GetRPCMethods	Retrieves a device's list of supported RPC methods
SetParamAttributes	Sets the value of device parameter attributes, such as Notification and Access List
SetParamValues	Sets the value of device parameters
ChangeProvGroup	Redirects a device to a new provisioning group
GenerateConfig	Regenerates the instruction set for a device
PassThrough	Sends a generic SOAP message to the device
RemoveOperation	Removes a pending on-connect device operation
RequestConnection	Initiates a connection request to a device

All device operations are executed before other configuration services, such as data synchronization, configuration synchronization, and firmware rules download.

**Caution**

You should not perform device configuration updates using device operations. The configuration process can overwrite any settings that you configured by using device operations.

## Connection Modes for Device Operations

Cisco BAC enables the performing of device operations in three modes:

- Immediate
- On-connect
- Asynchronous

### Immediate Mode

An immediate execution assumes that the device is reachable and is performed by the DPE first issuing a connection request to the device. If the device is not reachable, the operations automatically fail. You can run multiple operations in the same batch to improve performance, creating a single connection request and, thus, a single management session with the CWMP device.

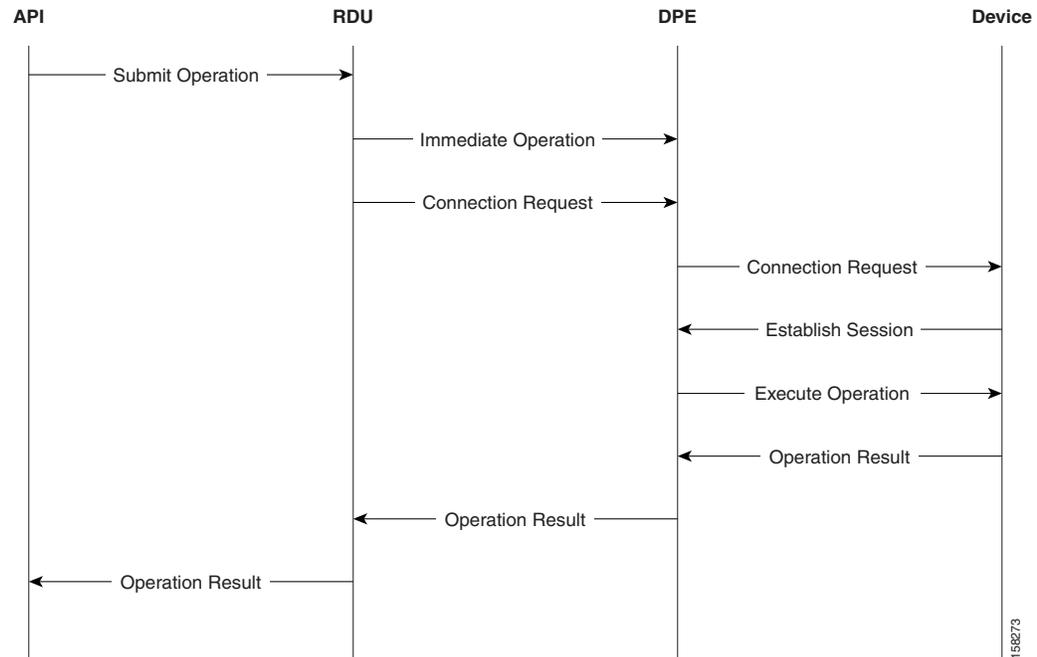
A client uses a timeout period to post the API batch in immediate mode. If no timeout is set, the duration for immediate operations to time out is, by default, 60 seconds. If a device operation times out or its batch times out, an error is returned.

**Note**

Before attempting to run immediate operations against a device, ensure that the device is reachable and that Cisco BAC is configured to perform connection requests to the device. See [Connection Request Service](#), page 12-2.

Figure 14-1 illustrates the high-level flow for immediate operations.

**Figure 14-1 Immediate Operation Mode Workflow**



When the `performOperation()` API method is invoked specifying an immediate execution, the RDU constructs and sends corresponding operation instructions to all DPEs in the device's provisioning group. Then, the RDU sends a connection request instruction to a specific DPE within the provisioning group; the DPE, in turn, sends a connection request notification to the device.

When the device establishes a connection with the DPE, the DPE creates a new session and performs the operation upon the device. When the session is closed, the DPE returns the result of the operation to the RDU, which forwards the result to the API client in the Batch's command status.

**Note**

You must use the batch mode flag, `AUTOMATIC` during the immediate mode.

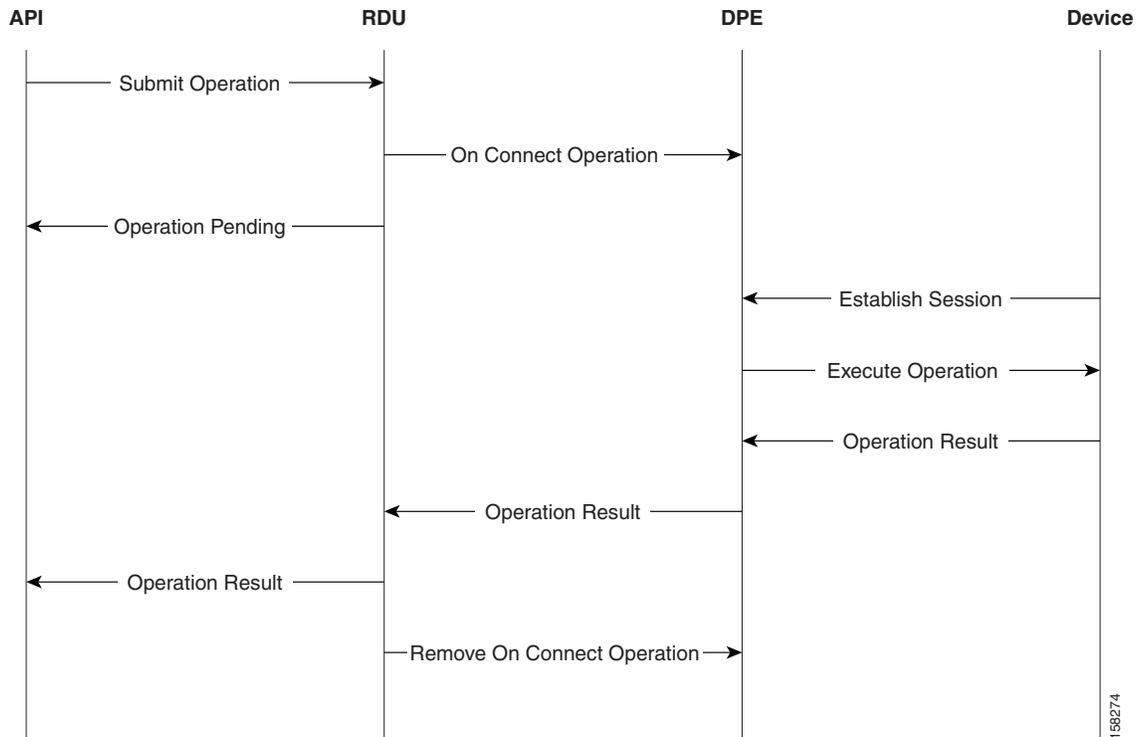
## On-Connect Mode

You can use an on-connect execution to support devices that are not reachable. It is best used in conjunction with periodic Inform RPCs from the CWMP device.

On-connect operations persist at each DPE in the device's provisioning group. They are not removed from the DPEs until the RDU instructs the DPEs to do so (after notifying the client of the execution results using an `AsyncOperationEvent`).

Figure 14-2 illustrates the high-level flow for on-connect operations.

**Figure 14-2 On-connect Operation Mode Workflow**



The API client constructs an `IPDeviceOperation` specifying an on-connect execution. It registers the appropriate `AsyncOperationEvent` listeners, then invokes the `performOperation()` API method. The RDU then constructs and sends corresponding operation instructions to all DPEs in the device's provisioning group. The DPEs save the operation in their internal database. The RDU batch completes, and the `BatchStatus` is returned to the API client.

The DPE does not initiate a connection request notification to the device, as it does while running an immediate operation.

When the device connects to the DPE the next time, the DPE creates a new session with the device and runs the operation upon the device. When the session is closed, the operation result is sent to the RDU, and the RDU sends an `AsyncOperationEvent` to any registered API client listeners.

Immediate operations are always run before on-connect operations. While you can use Cisco BAC to mix modes of operations along with the built-in instructions, you must ensure that the instructions do not override one other.

For example, if the device's configuration template specifies `False` for `InternetGatewayDevice.ManagementServer.PeriodicInformEnable` and a device operation changes that to `True`, the configuration template, which runs only after the device operation, overrides it to `False`.

## Asynchronous Mode

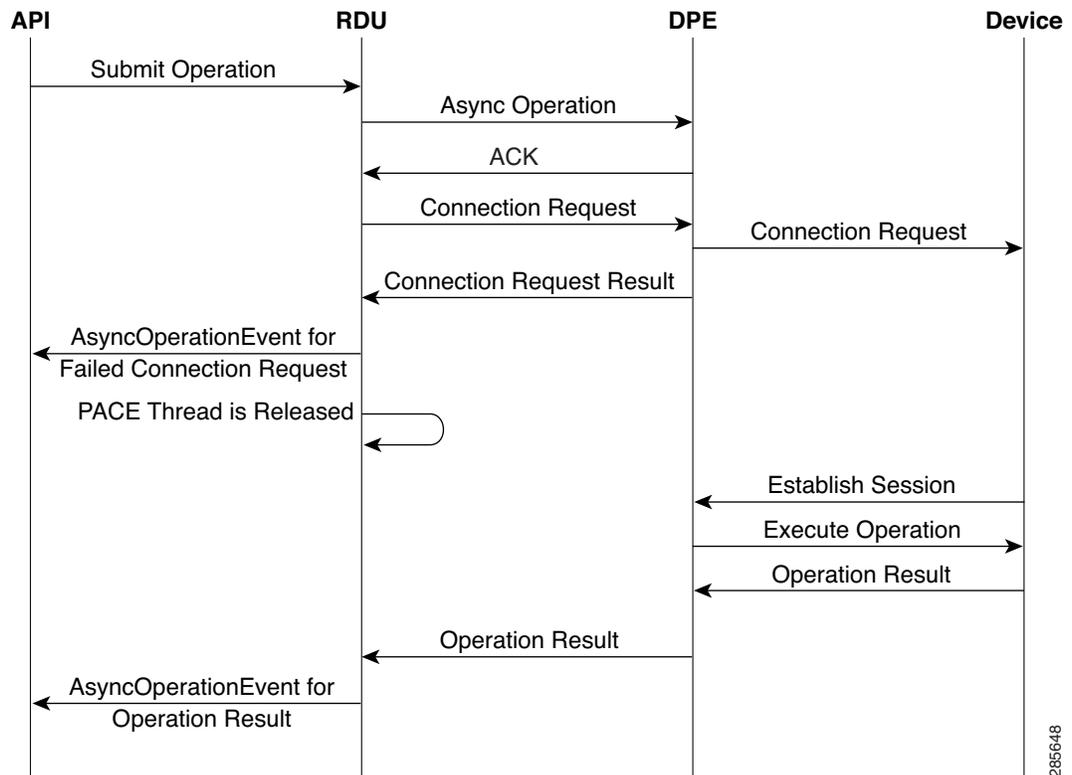
Asynchronous mode facilitates the RDU to execute the asynchronous immediate operation without reserving a PACE thread for the complete operation. The TR-069 IPDevice operations like GetParameterValues and GetParamValues RPC can be performed with this mode. This feature helps in increasing the RDU efficiency in serving more number of API operation requests.

The asynchronous operation is performed as follows:

1. The API client submits asynchronous operation batches to the RDU.
2. The RDU sends the asynchronous operation request to the DPE.
3. After receiving an acknowledgment from the DPE, RDU sends a connection request to DPE. The DPE then sends a connection request to the CPE.
4. The DPE sends the connection request result to the RDU.
5. If connection request fails, DPE sends an AsyncOperationEvent event to the RDU with the connection request status and the RDU releases the PACE thread.
6. When the connection request to CPE is successful, the CPE establishes a session in a separate thread, and the DPE executes the asynchronous operation on the CPE.
7. The CPE sends the operation results to DPE, and DPE further sends the operation result to RDU.
8. RDU processes the operation result and further sends an AsyncOperationEvent event to API client with the operation result.

Figure 14-3 illustrates the high-level flow for asynchronous operations.

**Figure 14-3 Asynchronous Operation Mode Workflow**



285648

**Note**


---

You must use the batch mode flag, `ASYNC_AUTOMATIC` during the asynchronous operation mode.

---

The API command `Configuration.getRDUDetails()` is enhanced to report statistics of the following asynchronous operations:

- Total number of asynchronous operation requests received by the RDU.
- Total number of asynchronous operation succeeded.
- Total number of asynchronous operation failed.
- Minimum time taken for asynchronous operation.
- Maximum time taken for asynchronous operation.
- Average time taken for asynchronous operation.
- Total time taken for all asynchronous operations.
- Total number of asynchronous immediate operations failed due to acknowledgment failed from DPE.
- Total number of asynchronous operations failed due to DPE server not found in Provisioning Group.
- Total number of asynchronous operations failed due to connection request error.
- Total number of asynchronous operations failed due to connection request failed.
- Total number of connection request succeeded for asynchronous operations requests.

## Conditional Execution

Additionally, immediate and on-connect operations are conditionally executed based on the TR-069 Event Codes in the Inform message from the device. If no intersection occurs between the reported Event Codes in the device's Inform and the list of Event Codes specified in the operation, the operation is not executed.

For example, an operation might be conditional upon the Inform Event Code list including `6 CONNECTION REQUEST`. Therefore, the operation only executes if the device reports it is connecting because of an autoconfiguration server (ACS) connection request. If the device reports it is connecting for any other reason, for instance a periodic Inform, the operation does not execute.

## Managing a Device's Provisioning Group

The provisioning group for a CWMP device is specified on the device object in the RDU. On the:

- Provisioning API, through the `IPDeviceKeys.HOME_PROV_GROUP` property.
- Administrator user interface, through the Home Provisioning Group drop-down list in the Devices pages.

In managing a device's provisioning group, occasionally, you might need to redirect or correct the provisioning group of a device:

- [Redirecting CPE to Home Provisioning Group, page 14-7.](#)

In this scenario, the device communicates with a specific provisioning group, for example PG1, and you want to redirect it to communicate with its home provisioning group, PG2.

- [Correcting a Device's Provisioning Group](#), page 14-8.

In this scenario, the device's provisioning group in Cisco BAC is incorrect and must be changed only in Cisco BAC; for example, the device attempts to contact PG1 but is provisioned for PG2 in Cisco BAC.

Both use cases are described in detail in the following sections.

**Note**

---

You cannot change a device's provisioning group if the DPE cannot connect to the RDU.

---

## Redirecting CPE to Home Provisioning Group

Cisco BAC redirects a device to its home provisioning group by having the provisioning groups communicate among themselves to find the device's correct home provisioning group.

When a device attempts to establish a session with a DPE, the DPE searches its cache for an entry related to that device. If it cannot find an entry for that device, it sends home provisioning group queries to the DPEs in other provisioning groups.

**Note**

---

If the home provisioning group is down or is not responding, the DPE that receives the Inform message returns the *503 Service Unavailable* error along with the *Retry-After* header. Thereafter, the session with the device is dropped.

---

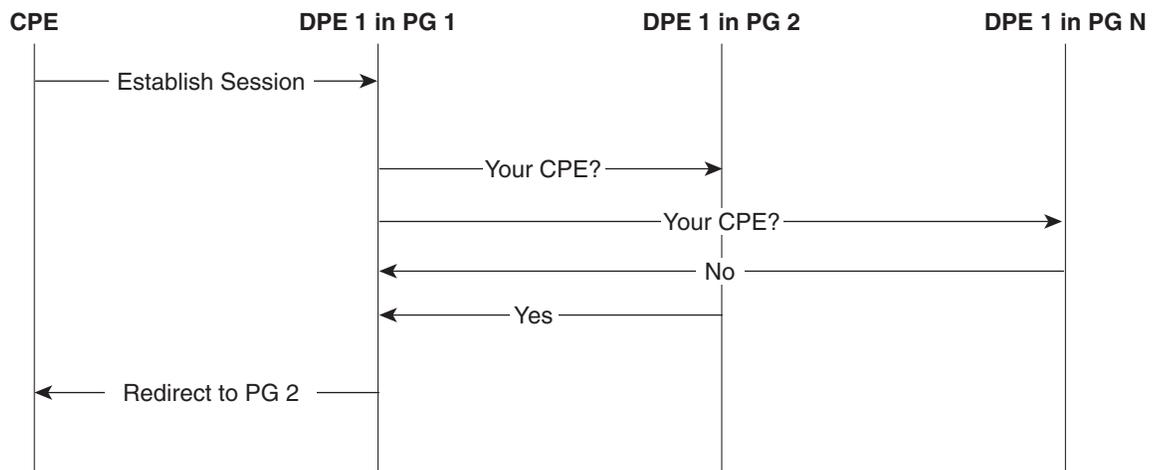
To select the best available DPE in each provisioning group, the DPE maintains the status data of all other DPEs in the deployment. The DPE sends a status request at configured time intervals to update its knowledge of the state of other provisioning groups.

Each state has a corresponding integer value. The higher the value, the better the state of the provisioning group. The DPE, using its knowledge of the current state of other DPEs, selects a DPE in the highest state in each provisioning group.

After sending the home provisioning group query, the DPE waits for a response from the other provisioning groups. If a provisioning group responds that the device belongs to its group, then the DPE redirects the device to its home provisioning group.

[Figure 14-4](#) highlights the basic flow when a device attempts to establish a session with the wrong provisioning group.

Figure 14-4 Redirecting CPE to Home Provisioning Group



1. The CPE attempts to establish a session with DPE 1 in provisioning group 1 (PG 1).
2. DPE 1 in PG 1 searches its cache for the device.  
If it cannot find an entry for that device, the DPE sends a request to the best available DPE in each provisioning group to find out whether the device belongs to that group.  
Only one DPE per provisioning group is queried.
3. The DPEs that receive the request in each provisioning group, send their reply to DPE 1.  
If a provisioning group responds that the device belongs to its group, then DPE 1 in PG 1 redirects the CPE to its home provisioning group.
4. The CPE attempts to establish a session with its home provisioning group.

To enable the home provisioning group redirection feature, you must configure the home provisioning group redirection service on the DPE. See [Configuring Home Provisioning Group Redirection Service on the DPE](#), page 3-9.

## Correcting a Device's Provisioning Group

You can also just reset an incorrect provisioning group specification on the device. In this scenario, you do not need to change the actual device—you only need to update Cisco BAC.

To change an incorrect provisioning group from the API, invoke the `IPDevice.changeProperties()` API and include the `IPDeviceKeys.HOME_PROV_GROUP` property with the corrected provisioning group. For details, refer to the API Javadoc which is available at `BPR_HOME/docs/nb-api/javadoc`.

To change an incorrect provisioning group specification from the administrator user interface:



### Note

If you change the home provisioning group from the administrator user interface, you only correct the provisioning group in Cisco BAC; the change does not affect the device.

- Step 1** Click the **Devices** tab on the primary navigation bar.

The Manage Devices page appears.

- Step 2** Click the Identifier link corresponding to the correct device.
- Step 3** Select your options from the Home Provisioning Group drop-down list.
- Step 4** Click **Submit** to save the changes to the device.

The device is now assigned to a specific provisioning group.

---

