Broadband Access Center for Cable System Architecture

This chapter describes the system architecture implemented in this Broadband Access Center for Cable (BACC) release. The architecture described in this chapter is for information purpose only and is intended as background for those using the BACC product.

Architecture

This section describes the basic BACC architecture including:

- **Regional distribution unit (RDU) that provides:**
  - The authoritative data store of the BACC system.
  - Support for processing API requests.
  - A device detection/configuration/disruption engine, which invokes the various installed extensions to determine device types, generate configurations, and support disruption for specific device types.

See the “Regional Distribution Unit” section on page 2-3 for additional information.

- **Device provisioning engines (DPEs) that provide:**
  - TFTP protocol server
  - Configuration cache
  - Redundancy
  - Time of day server
  - PacketCable provisioning services

See the “Device Provisioning Engines” section on page 2-4 for additional information.

- **Provisioning Groups**

See the “Provisioning Groups” section on page 2-8 for additional information.

- **Cisco Network Registrar servers that provide:**
  - Dynamic Host Configuration Protocol (DHCP)
  - Domain Name System (DNS)

See the “Cisco Network Registrar” section on page 2-9 for additional information.
Registration Modes

Registration modes give the service provider a way to control the number of interactions with the subscriber. For any registered device, the service provider must be prepared to process any change to the device. So there is a significant difference between registering 100 cable modems with unregistered computers behind them and registering 100 cable modems each of which has a potentially large number of registered computers behind them. For this reason, the service provider must carefully choose among the standard, promiscuous, and roaming registration modes.

Standard Mode

When operating in the standard mode (sometimes called the fixed mode) a computer is given unprovisioned access when it is moved behind a different cable modem.

Promiscuous Mode

Only the device is registered; the DHCP server maintains lease information about a device operating behind another device. All devices behind a registered device receive network access.

Roaming Mode

In the roaming mode, a registered device can receive its class of service behind any other registered device. This permits, for example, the use of a laptop moving from location to location and obtaining service from multiple cable modems.

Mixed Mode

A mix of the promiscuous mode and the roaming mode can also be used as the two separate modes of operation can also coexist simultaneously.
Regional Distribution Unit

The RDU is the primary server in the BACC provisioning system. The RDU also provides:

- Management of device configuration generation.
- Authoritative datastore for the BACC system.
- Device disruption.
- A clearinghouse function, through which all application programming interface (API) requests must pass.
- Management of the BACC system.

The RDU supports the addition of new technologies and services through an extensible architecture. The RDU also supports:

- Data access and manipulation by means of the API.
- Distributing configurations to the DPEs for scalability.
- External clients, operations support systems (OSSs), and other provisioning related functionality by means of the API.

These sections describe these RDU concepts:

- Generating Device Configurations, page 2-3
- Service Level Selection, page 2-3
- Regional Distribution Unit Failover, page 2-4

Note: The RDU determines the names of DPEs and Network Registrar extensions by the interface they connect to the RDU on. That is, the name of the DPE or Network Registrar extensions is what the RDU machine thinks it is. This requires the RDU to be able to resolve the names (both forward and reverse).

Generating Device Configurations

When a device boots, it requests a configuration from BACC and this configuration determines the level of service for the device. Device configurations can include customer required provisioning information, such as: DHCP IP address selection, bandwidth, data rates, flow control, communication speeds, and level of service. A configuration can contain DHCP configuration and TFTP files for any device. When an unprovisioned device is installed, and the boot operation performed, a default configuration for the appropriate technology is obtained from BACC and sent to the device, by means of either DHCP or TFTP. The default configuration can be changed for each supported technology.

Service Level Selection

The service level selection extension point determines the criteria to be used by the RDU and notifies the RDU which criteria to use when generating a device configuration. The RDU stores this information for each device in the RDU database. Although a device may have been registered as having to receive one set of criteria, a second set may actually get selected. The configuration generation extensions look for the selected criteria and use them. Consequently, since the RDU automatic regeneration now knows that a second set of criteria are being used, the device configuration is regenerated if any changes occur to any of the criteria.
Service level selection extension points are entered on various technology defaults pages (“Configuring Defaults” section on page 10-7 for additional information). These are entered as a comma-separated list of extensions to instantiate creation of an object that can be used as the extension(s) for that specific extension point. By default, these properties are populated with zero or with one of the built-in extensions. These extensions should not be modified unless you are installing your own custom extensions.

Regional Distribution Unit Failover

Broadband Access Center for Cable currently supports one RDU per installation. For failover support, use your own hardware redundant system or obtain a similar system with hot-swap capability.

Device Provisioning Engines

The Cisco Device Provisioning Engine (DPE) communicates with the RDU to give devices their configurations. Each DPE caches information for up to one million devices and multiple DPEs can be used to ensure redundancy and scalability.

The DPE handles all configuration requests including providing configuration files for devices. It is integrated with the Cisco Network Registrar DHCP server to control the assignment of IP addresses for each device. Multiple DPEs can communicate with a single DHCP server.

The DPE manages these activities:

- Last step device configuration generation (DOCSIS timestamps for instance).
- Communication of the configuration files through an embedded TFTP server.
- Integration with Cisco Network Registrar.
- Time of day protocol server.
- Voice technology provisioning services.

Types of Device Provisioning Engine

Two types of DPE are supported by this BACC release; the traditional hardware device (either DPE-590 or DPE-2115) and the software only Solaris DPE. See the “Hardware Device Provisioning Engines” section on page 2-5 and the “Solaris Device Provisioning Engines” section on page 2-5 for additional information.

In addition to the different types of DPE, this section describes these major DPE components:

- DPE License Keys, page 2-5
- TACACS+ and DPE Authentication, page 2-6
- DPE Server Assignments, page 2-7
- TFTP Server, page 2-7
- Provisioning Groups, page 2-8
Hardware Device Provisioning Engines

BACC currently supports the DPE-590 and the DPE-2115 hardware device provisioning engines. Refer to these documents for additional information concerning the device itself, ports, connectors, and rear panel components:

- For the DPE-590, refer to the *Cisco Content Engine 500 Series Hardware Installation Guide*. This can be found at: www.cisco.com/en/US/products/hw/contnetw/ps761/products_installation_guide_book09186a00800801e0.html
- For the DPE-2115, refer to the *Installation and Setup Guide for the Cisco 1102 VLAN Policy Server*. This can be found at: www.cisco.com/en/US/products/sw/secursw/ps2136/products_installation_and_configuration_guide_book09186a00801f0d02.html

**Note**
Whenever an interface link between a DPE-2115 and a Catalyst switch is interrupted, a default 30-second delay occurs before data traffic flows.

Solaris Device Provisioning Engines

The Solaris DPE functions in the same way as the hardware DPE with the exception that this part of the BACC product is installed on a computer running either the Solaris 8 or 9 operating system. With few exceptions, the same command line interface CLI is used on both the hardware and Solaris DPEs. See the *Cisco Broadband Access Center for Cable Command Line Interface Reference* for specific information on the CLI commands that each DPE supports.

DPE License Keys

Licensing controls the number of DPEs (nodes) you can use. If you attempt to install more DPEs than you are licensed to have, those new DPEs will not be registered, and will be rejected. Existing valid DPEs remain online and register themselves again with the RDU at will.

**Note**
For licensing purposes, a registered DPE is considered to be one node.

The number of DPE licenses you register with the RDU includes hardware and Solaris DPEs, regardless of release number or type; including those used as part of a BACC lab installation. See the *Broadband Access Center for Cable Installation Guide* for additional information.

Whenever you change licenses, either by adding a license, extending an evaluation license, or through the expiration of an evaluation license, the changes take immediate effect.

When you delete a DPE from the RDU database, a license is freed.

Deleted DPEs are removed from all provisioning groups they serve and all Network Registrar extensions are notified that the DPE is no longer available. Consequently, when a previously deleted DPE is registered again, it is considered to be licensed again and will remain so until it is deleted from the RDU again or its license expires.

DPEs that are not licensed through the RDU do not appear in the administrator’s user interface. You can determine the license state only by examining the DPE and RDU log files (dpe.log and rdu.log).
Any deleted or unlicensed hardware DPE, running either BPR 2.0.x or BACC 2.5, will constantly attempt to register with the RDU. This is normal behavior for devices running these versions of software and all rejected registrations are recorded in the RDU log and in the generation of SYSLOG alerts identifying this situation.

**TACACS+ and DPE Authentication**

Terminal Access Controller Access Control System plus (TACACS+) is a TCP-based protocol that supports centralized access control for large numbers of network devices and user authentication for the DPE CLI. Through the use of TACACS+ a DPE can support multiple users, with each username and login/enable password configured at the TACACS+ server. TACACS+ is used to implement the TACACS+ client/server protocol (ASCII login only).

**TACACS+ Privilege Levels**

The TACACS+ server uses the TACACS+ protocol to authenticate any user logging into a DPE. The TACACS+ client specifies a certain service level that is configured for the user. Table 2-1 identifies the two service levels used to authorize DPE user access:

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login</td>
<td>User-level commands at router&gt; prompt.</td>
</tr>
<tr>
<td>Enable</td>
<td>Enable-level commands at router# prompt.</td>
</tr>
</tbody>
</table>

**TACACS+ Client Settings**

TACACS+ uses a number of properties that are configured using the command line interface (CLI). See the *Cisco Broadband Access Center for Cable Command Line Reference* for information on these TACACS+ related CLI commands.

When TACACS+ is enabled, the administrator must specify either the IP addresses of all TACACS+ servers or their FQDNs with non-default values.

The administrator can also specify these settings, using their default values if applicable:

- The shared secret key for each TACACS+ server. This key is used for data encryption between the DPE and the TACACS+ server. If you choose to omit the shared secret for any specific TACACS+ server, TACACS+ message encryption is not used.
- The TACACS+ server timeout value. This is the maximum time that the TACACS+ client will wait for a TACACS+ server to reply to protocol requests.
- The TACACS+ server number of retries. This identifies the number of times that the TACACS+ client attempts a valid protocol exchange with a TACACS+ server.

**Note**

These commands are used in both the hardware and Solaris DPEs. In the hardware DPE, you can use only console mode.
DPE Server Assignments

BACC supports multiple DPEs. These communicate with devices, a DHCP failover configuration, and the RDU. During installation, you must make these DPE assignments:

- Assign a DPE to be responsible for one or more logical groups of devices or provisioning groups.
- The IP address and port number of the RDU.

The DPE's primary objective is to send configurations to customer devices whenever those devices are powered up or rebooted. To do this quickly, the DPE keeps a copy of the configuration for each device in a local cache database.

Note
A DPE should be assigned to only one provisioning group.

TFTP Server

The integrated TFTP server receives requests for files, including DOCSIS configuration files, both from device and non-device entities. This server then transmits the file to the requesting entity.

The TFTP server is located in a home directory that is used for local file system access. The local files are stored in the BACC_DATA/dpe/tftp directory.

By default, the TFTP server only looks in its cache for a TFTP read. However, if the tftp allow-read-access command has been run, the TFTP server first looks at the local file system before looking in the cache. If the file exists in the local file system, it is read from there. If not, the TFTP server looks in the cache. If the file is there, the server uses it. Otherwise, the server sends a request to the RDU for the file. When you can enable read access from the local file system, directory structure read requests are allowed only from the local file system.

BACC only allows writes to the local file system; never to the DPE cache. If you run the tftp allow-write-access DPE CLI command, BACC will allow a write to the TFTP home directory. By default, the creation of directories or the override files is not permitted. However, to change this, run either the tftp allow-create-dirs or the tftp allow-override commands.

DOCSIS Shared Secret

BACC lets you define multiple different DOCSIS shared secrets (DSS), for dynamic DOCSIS configuration files only, on devices belonging to different CMTSs. In this way, a compromised shared secret results in a limited number of compromised CMTSs and not every CMTS in the entire deployment.

Although the DSS can be set for each DPE, you should set it on a provisioning group basis and it must match what has been configured for the CMTSs in that provisioning group.

Caution
Configuring multiple DSS within one provisioning group could, under some conditions, result in degraded CMTS performance. This should have virtually no effect on BACC.

You can enter the shared secret as clear text or as IOS-encrypted format.
When entered in clear text, the DSS is encrypted to suit IOS version 12.2BC. You can also set the DSS from the RDU using the Administrator’s user interface or the API. In this case, the DSS is entered, stored at the RDU, and passed to all DPEs in clear text. Consequently, before a DSS entered this way is stored on the DPE, it is encrypted.

If you set the DSS directly at the DPE, using the appropriate CLI command, this will take precedence over what is set from RDU.

**Resetting the DOCSIS Shared Secret**

If the security of the DSS becomes compromised, or if you simply want to change the shared secret for administrative purposes, you can run the show running config CLI command and then copy and paste the DOCSIS shared secret line from the displayed configuration back into the DPE configuration. In this way you can copy what you enter in a Cisco CMTS to the DPE CLI. See the *Cisco Broadband Access Center for Cable Command Line Reference* for additional information.

*Note*

To change the shared secret as described above, the CMTS must be running a software version newer than V 12.2BC.

When the DSS must be changed, you must:

**Step 1** Identify the provisioning group with the DOCSIS shared secret to be reset.

**Step 2** Examine the list of DPEs and CMTSs associated with the provisioning group.

**Step 3** Change the primary DSS on the CMTS.

**Step 4** Change the compromised DSS on the CMTS to the secondary DSS. This is required to allow cable modems to continue to register until all of the DOCSIS configuration files have been successfully changed to use the new DSS.

**Step 5** Determine which DPEs were affected and change the DSS accordingly on each.

**Step 6** Confirm that the DOCSIS configuration files are using the new DSS and then remove the CMTS compromised secondary shared secret from the CMTS configuration.

**Provisioning Groups**

A provisioning group is designed to be a regional grouping of servers usually consisting of two or more DPEs and a failover pair of DHCP servers that can handle the provisioning needs of up to one million devices. As the number of devices grows past one million, you can add additional provisioning groups to the deployment.

*Note*

The servers for a provisioning group are not required to reside at a regional location, they can just as easily be deployed in the central NOC.
To support redundancy and load sharing each provisioning group can support any number of DPEs. As the requests come in from the DHCP servers, they are distributed between the DPEs in the provisioning group and an affinity is established between the devices and a specific DPE. This affinity is retained as long as the DPE state within the provisioning group remains stable. Because a specific device will normally be assigned to the same DPE, expensive Kerberos reticketing is avoided.

Cisco Network Registrar

Network Registrar provides Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS) functionality. It has a complete administrative user interface that, when coupled with customized BACC configuration screens, can be used within a larger enterprise management system.

Note
For additional information on Network Registrar, refer to these documents: Network Registrar User's Guide, Network Registrar CLI Reference, and the Network Registrar Installation Guide.

Dynamic Host Configuration Protocol

The Dynamic Host Configuration Protocol (DHCP) server automates the process of configuring IP addresses on TCP/IP networks. This protocol performs many of the functions that a system administrator carries out when connecting a device to a network. DHCP automatically manages network policy decisions and eliminates the need for manual configuration. This adds flexibility, mobility, and control to networked device configurations.

DHCP failover allows pairs of DHCP servers to act in such a way that one can take over if the other stops functioning. The server pairs are known as the primary and backup server. Under normal circumstances the primary server performs all DHCP functions. If the primary server becomes unavailable, the backup server takes over. In this way, DHCP failover prevents loss of access to the DHCP service if the primary fails.

Domain Name System

The Domain Name System (DNS) server contains information on hosts throughout the network, including IP address hostnames and routing information, and DNS uses them primarily to translate between IP addresses and domain names. This conversion of names, such as www.cisco.com, to IP addresses simplifies accessing Internet-based applications.

The DNS directory service consists of:

- DNS data.
- DNS servers.
- Internet protocols for fetching data from the servers.
- Dynamic DNS for voice provisioning.
Lease Reservation

BACC lease reservation works in conjunction with Network Registrar’s Central Configuration Management (CCM) to assign a device with a static IP address during provisioning.

When you provision a new device, BACC determines whether the IP address is specified and then determines which Network Registrar server identifies it is a valid IP address. After validation, the lease reservation function creates a reservation for the device using the Network Registrar CCM.

Lease reservation operates with all technologies supported by BACC and:

- Lets you add and remove IP address reservations using the BACC graphical user interface. See Managing Devices, page 9-12 for additional information.
- Reports all errors resulting from attempts to reserve an IP address that is already in use or if a reservation is removed from the CCM server.

For lease reservation to operate properly, you must have CNS Network Registrar version 6.1.2.3, with a licensed CCM server present, installed in your network.

You must configure the CCM address, port, username, and password before BACC can implement lease reservation. These parameters are set from the RDU Defaults page. Changes are dynamic and take effect immediately after being entered. (See the “RDU Defaults” section on page 10-19 for information on these configuration parameters.)

The lease reservation function is disabled by default and times out whenever the CCM cannot be reached for a specified amount of time.

Key Distribution Center

The key distribution center (KDC) is an authentication server used to authenticate MTAs and grant security tickets to them.

The KDC is supported on multi-processor computers.

Default KDC Properties

The KDC has several default properties that get populated during BACC installation into a <BACC_HOME>/kdc/solaris/kdc.ini properties file. This file can be edited to change values as operational requirements dictate. After changes have been made, you must restart the KDC before any property file, key or certificate changes can take affect. The default properties are:

- interface address—This is the IP address of the local Ethernet interface that you want the KDC to monitor for incoming Kerberos messages. For example:
  
  interface address = 10.10.10.1

- FQDN—Identifies the fully qualified domain name (FQDN) on which the KDC is installed. For example:
  
  FQDN = kdc.cisco.com
The interface address and FQDN are entered through the KDC Realm Name screen during installation. Refer to the Broadband Access Center for Cable Installation Guide for specific information.

- **maximum log file size**—The KDC generates a set of log files. This property specifies the maximum size, in kilobytes, that the log file can reach. Therefore, the KDC will create a new log file only when the current file reaches this maximum size. For example:
  
  \[
  \text{maximum log file size} = 1000
  \]

- **n saved log files**—This property defines the number of old log files that the KDC saves. The default value is 7, and you can specify as many as required. For example:
  
  \[
  \text{n saved log files} = 10
  \]

- **log debug level**—This property specifies the logging level for the log file.
  
  \[
  \text{log debug level} = 5
  \]

- **minimum (maximum) ps backoff**—This property specifies the minimum (or maximum) time, in tenths of a second, that the KDC will wait for BACC to respond to the FQDN-REQUEST. For example:
  
  \[
  \text{minimum ps backoff} = 150
  \]

Using the example values shown above, a sample INI file might contain data similar to that shown in Example 2-1.

**Example 2-1  Sample KDC INI Configuration File**

```ini
interface address = 10.10.10.1
FQDN = kdc.cisco.com
maximum log file size = 1000
n saved log files = 10
log debug level = 5
minimum ps backoff = 150
maximum ps backoff = 300
```

You can set the times for both minimum and maximum ticket duration to effectively smooth out excessive numbers of ticket requests that could occur during deployment. This is beneficial given that most deployments occur during traditional working hours and excessive loading might, from time to time, adversely affect performance.

**Note**

Shortening the ticket duration forces the MTA to authenticate to the KDC much more frequently. While this results in much greater control over the authorization of telephony endpoints, it also causes much heavier message loads on the KDC and increased network traffic. For most circumstances, the default setting is appropriate and should not be changed.

- **maximum ticket duration**—This property defines the maximum duration for tickets generated by the KDC. The default unit is hours; however, by appending an m or d, the units can be changed to minutes or days, respectively.

  The default value is 168, or seven days, and Cisco recommends that you not change this value because this is the duration required to conform to the PacketCable security specification. For example:

  \[
  \text{maximum ticket duration} = 168
  \]
• minimum ticket duration—This property defines the minimum duration for tickets generated by the KDC. The default unit is hours; however, by appending an \texttt{m} or \texttt{d}, the units can be changed to minutes or days, respectively.

The default value is 144, or six days, and Cisco recommends that you not change this value. For example:

\begin{verbatim}
minimum ticket duration = 144
\end{verbatim}

**Euro-PacketCable Support**

The Key Distribution Center (KDC) component supports Euro-PacketCable (tComLabs) certificate chains. Example 2-2 shows a sample Euro-PacketCable enabled KDC configuration file.

**Example 2-2  Example Euro-PacketCable Enabled KDC Configuration File**

\begin{verbatim}
[generic]
interface address = 10.10.10.1
FQDN = servername.cisco.com
maximum log file size = 10000
number of saved log files = 100
log debug level = 5 minimum
ps backoff = 150 maximum
ps backoff = 300
euro-packetcable = true
\end{verbatim}

**KDC Certificates**

The certificates used to authenticate the KDC are not shipped with BACC. You must obtain the required certificates from Cable Television Laboratories Inc. (CableLabs) and the content of these certificates must match those that are installed in the MTA. See the “Using the PKCert.sh Tool to Manage KDC Certificates” section on page 12-41 for additional information.

\begin{itemize}
  \item \textbf{Caution} Without the certificates installed, the KDC will not function.
\end{itemize}

**KDC Licenses**

Obtain a KDC license from your Cisco representative and then install it in the correct directory.

To install a KDC license file:

\begin{itemize}
  \item \textbf{Step 1} Obtain your license file.
  \item \textbf{Step 2} Copy the license file to the \texttt{<BACC\_HOME>/kdc} directory.
  \item \textbf{Caution} Be careful not to copy this as an ASCII file. The file contains binary data susceptible to unwanted modification during an ASCII transfer.
  \item \textbf{Step 3} If the KDC license file is not called \texttt{baccelc.license}, rename the file to \texttt{bacekdc.license}.
\end{itemize}
Step 4 Run the `bprAgent restart kdc` command, from the `/etc/init.d` directory, to restart the KDC server and make the changes take effect.

Note KDC license files should not be copied between operating systems, as the transfer process may damage the file.

Multiple Realm Support

The BACC KDC supports the management of multiple realms.

To configure additional realms:

Step 1 Locate the directory containing your KDC certificates.

Step 2 Create a subdirectory there with a name matching the desired realm name. This subdirectory must be created using upper case characters only.

Step 3 Place the KDC certificate and the private key for the realm in this directory.

Step 4 If the new realm is not chained to the same Service Provider as the KDC certificate, include all additional higher level certificates which differ from those in the certificates directory. However, since all realms must be rooted in the same certificate chain, only one locale (PacketCable, Euro-PacketCable) is supported per KDC installation.

Note Any given DPE is restricted to providing a service to a single realm.

BACC MIBs

Broadband Access Center for Cable supports several different MIBs. These include:

- CISCO-BACC-SERVER-MIB—The CISCO-BACC-SERVER-MIB defines managed objects that are common to all BACC servers. This MIB supports the monitoring of multiple BACC servers when they are installed on the same device. The `ciscoBaccServerStateChanged` notification is generated every time a server state change takes place.
Table 2-2 summarizes BACC MIB support based on installation type. See the *Cisco Broadband Access Center for Cable Installation Guide* for descriptions of each installation type.

**Table 2-2  BACC Supported MIBs**

<table>
<thead>
<tr>
<th>Installation type</th>
<th>MIBs Supported</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solaris DPE</td>
<td>CISCO-BACC-SERVER-MIB</td>
</tr>
<tr>
<td></td>
<td>CISCO-BACC-DPE-MIB</td>
</tr>
<tr>
<td>Hardware DPE</td>
<td>RFC1213 - MIB II</td>
</tr>
<tr>
<td></td>
<td>CISCO-APPLIANCE-MIB</td>
</tr>
<tr>
<td></td>
<td>CISCO-BACC-SERVER-MIB</td>
</tr>
<tr>
<td></td>
<td>CISCO-BACC-DPE-MIB</td>
</tr>
<tr>
<td>RDU</td>
<td>CISCO-BACC-SERVER-MIB</td>
</tr>
<tr>
<td></td>
<td>CISCO-BACC-RDU-MIB</td>
</tr>
</tbody>
</table>

**BACC Agents**

This section describes BACC agents; what they are and why they are important. Subsequent descriptions provide all the details required to use and understand the agents. These agents are described:

- SNMP Agent, page 2-14
- BACC Agent, page 2-15

**SNMP Agent**

BACC provides basic SNMP v2 based monitoring of both the DPE and RDU servers. The BACC SNMP agents support both SNMP informs and traps. You can configure the SNMP agent on the DPE using `snmp-server` CLI commands and on RDU by using SNMP configuration CLI commands.

See the “Using the `snmpAgentCfgUtil.sh` Command” section on page 12-46 for additional information on the SNMP configuration command line tool, and the *Cisco Broadband Access Center for Cable Command Line Reference* for additional information on the DPE CLI.

**MIB Support**

In addition to RFC 1213 (MIB-II), the SNMP agent supports the CISCO-CW-APPLIANCE-MIB. This MIB defines the managed objects for the software components installed on a hardware DPE. It monitors CPU, memory and disk utilization, and generates notifications whenever use exceeds certain thresholds. Notifications can be selectively enabled and disabled. Resource use is polled at regular intervals and notifications are generated when the average of two consecutive data points exceeds the threshold.

The SNMP agent also supports the CISCO-BACC-DPE-MIB. This MIB defines the managed objects for the software components installed on a Solaris DPE. The DPE manages local caching of device configurations and configuration files used by all supported devices. This MIB provides some basic DPE configuration and statistics information, including entries for TFTP and ToD servers.
The SNMP agent supports the CISCO-NMS-APPL-HEALTH-MIB which defines the Cisco NMS application health status notifications and related objects. These notifications are sent to the OSS/NMS to inform them about the NMS application status, including: started, stopped, failed, busy, or any abnormal exit of applications. The default MIB used is MIB-II.

The SNMP agent generates a cnaHealthNotif trap that announces that the RDU server has started, shutdown, failed, or there is a change in the exit status.

The SNMP agent supports the CISCO-BACC-RDU-MIB, which defines managed objects for the RDU. This MIB also contains managed objects defining statistics between the RDU and DPE and between the RDU and Network Registrar.

Note

These MIBs are located in the <BACC_HOME>/rdu/mibs directory.

The BACC agent is an administrative agent that monitors the run time health of all BACC processes. This watchdog process ensures that if a process stops unexpectedly, it is automatically restarted.

The BACC agent can be used as a command line tool to start, stop, restart, and determine the status of any monitored processes.

Monitored Processes

If a monitored application fails, it is restarted automatically. If, for any reason, the restart process also fails, the BACC agent server will wait a prescribed amount of time before attempting to restart again.

Note

You do not have to use the BACC agent and the SNMP agent to monitor Network Registrar extensions.

The period between restart attempts increases exponentially until it reaches a length of 5 minutes. After that, the process restart is attempted at 5 minute intervals until successful. Five minutes after a successful restart, the period is automatically reset to 1 second again.

For example:

- Process A fails.
- The BACC agent server attempts to restart it and the first restart fails.
- The BACC agent server waits 2 seconds and attempts to restart the process and the second restart fails.
- The BACC agent server waits 4 seconds and attempts to restart the process and the third restart fails.
- The BACC agent server waits 16 seconds and attempts to restart the process and the fourth restart fails.
Logging

Logging of events is performed at both the DPE and RDU, and in some unique situations, DPE events are logged at the RDU. Log files are located in their own log directories and can be examined using any text processor. The files can be compressed to allow them to be easily emailed to the TAC or system integrators for troubleshooting and fault resolution.

BACC Agent Command Line

The BACC agent automatically starts whenever the system boots up. Consequently, this agent also starts those BACC system components it is configured to control. The BACC agent can also be controlled through a simple command line interface. This is performed by running the bprAgent command from the /etc/init.d directory.

Table 2-3 describes the CLI commands available for use with the BACC agent. You can run the CLI from the etc/init.d directory.

Table 2-3  BACC Command Line Interface

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bprAgent start</td>
<td>Starts the BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent stop</td>
<td>Stops the BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent restart</td>
<td>Restarts the BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent status</td>
<td>Gets the status of the BACC agent, including all monitored processes.</td>
</tr>
<tr>
<td>bprAgent start &lt;process-name&gt;</td>
<td>Starts one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
<tr>
<td>bprAgent stop &lt;process-name&gt;</td>
<td>Stops one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
<tr>
<td>bprAgent restart &lt;process-name&gt;</td>
<td>Restarts one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
<tr>
<td>bprAgent status &lt;process-name&gt;</td>
<td>Gets the status of one particular monitored process. The value &lt;process-name&gt; identifies that process.</td>
</tr>
</tbody>
</table>

Note: The <process-name> mentioned in Table 2-3 can be one of the rdu, kdc, dpe, SnmpAgent, and jrun (which runs the administrators and sample user interface) processes. The CLI process is also monitored in both Lab and DPE installations.

The RDU operates in a Solaris environment and may not shut down satisfactorily each time the Solaris reboot command is used. The preferred command to bring down the system is shutdown.
Regional Distribution Unit Logs

The RDU has two logs that it maintains in the BACC_DATA/rdu/logs directory:

- `rdu.log`—Records all RDU events with the configured default level. See the “Setting the RDU Log Level” section on page 12-40 for instructions on setting the default log levels.
- `audit.log`—Records all high level changes to the BACC configuration or functionality including the user who made the change.

Device Provisioning Engine Logs

The DPE maintains a `dpe.log` file, in the BACC_DATA/dpe/logs directory, that also records all events having the configured default level. In situations where the DPE undergoes catastrophic failure, such as engaging in a series of system crashes, the catastrophic errors are also logged into the `rdu.log` file.

The SNMPService.logyyy.log log file is used by the DPE, when PacketCable is enabled on the DPE server, to provide detailed debugging information. You use the `show packetcable snmp log` DPE CLI command to view this file, which is also located in the BACC_DATA/dpe/logs directory. See the Cisco Broadband Access Center for Cable Command Line Reference for PacketCable command usage.

Note
PacketCable logging messages are sent to the `dpe.log` file and the detailed SNMP debugging is sent to the SNMPService.logyyy.log file.

Log Levels and Structures

The log file structure is described here, and illustrated in Example 2-3, and includes:

- **Domain Name**—This is the name of the computer generating the log files.
- **Date and Time**—This is the date on which a message is logged. This information also identifies the applicable time zone.
- **Facility**—This identifies the system which, in this case is the BACC.
- **SubFacility**—This identifies the BACC subsystem or component.
- **Severity Number**—The logging system defines seven levels of severity (log levels) that are used to identify the urgency with which you might want to address log issues. The process of configuring log levels is described in the “Configuring Log Levels” section on page 2-18:
  - 0—Emergency: System unstable.
  - 1—Alert: Immediate action is needed.
  - 3—Error: Error conditions exist.
  - 4—Warning: Warning condition exists.
  - 5—Notification: A normal but significant condition exists.
  - 6—Information: Informational messages only.
Note Another level known as DEBUG is used exclusively by Cisco for debugging purposes. Do not use this level except at the direction of the Cisco TAC.

- Message ID—This is a unique identifier for the message text.
- Message—This is the actual log message.

Note See the “The RDU Log Level Tool” section on page 12-38 for additional logging information.

## Configuring Log Levels

You can configure logging levels for both the RDU and the DPE to suit your specific requirements. For example, the logging level for the RDU could be set to Warning and the level for the DPE could be set to Alert.

Log messages are written based on certain events taking place. Whenever an event takes place, the appropriate log message and level are assigned and, if that level is less than or equal to the configured level the message is written to the log. The message is not written to the log if the level is higher than the configured value.

For example, assume that the log level is set to 4-Warning. All events generating messages with a log level of 4 or less are written into the log file. If the log level is set to 6-Information, the log file will receive all messages. Consequently, configuring a higher log level results in a larger log file size.

### Example 2-3 Sample Log File

<table>
<thead>
<tr>
<th>Domain Name</th>
<th>Data and Time</th>
<th>Facility</th>
<th>Sub-facility</th>
<th>Security Level</th>
<th>Msg ID</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:11 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td></td>
<td>0236:</td>
<td>BPR Regional Distribution Unit starting up</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:15 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0566:</td>
<td>Initialized API defaults</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:15 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0567:</td>
<td>Initialized CNR defaults</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:15 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0568:</td>
<td>Initialized server defaults</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:18 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0570:</td>
<td>Initialized DOCSIS defaults</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:18 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0571:</td>
<td>Initialized computer defaults</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:19 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0573:</td>
<td>Initialized CableHome defaults</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:19 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0572:</td>
<td>Initialized PacketCable defaults</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:19 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0569:</td>
<td>Created default admin user</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:19 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>5</td>
<td>0574:</td>
<td>Loaded 6 license keys</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:25 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>3</td>
<td>0015:</td>
<td>Unable to locate manifest file</td>
</tr>
<tr>
<td>BACC1:</td>
<td>2005 3 16 03:06:28 EST:</td>
<td>BPR-</td>
<td>RDU-</td>
<td>3</td>
<td>0280:</td>
<td>Command error</td>
</tr>
</tbody>
</table>

Note The KDC is not considered in this log file.
Viewing the dpe.log File

You use the `show log` command, from the DPE CLI, to view the log file. See the Cisco Broadband Access Center for Cable Command Line Reference for additional information.

BACC generates log messages from the DHCP server extensions based on the extension trace level setting. You can change the extension trace level using the administrator’s user interface.

To do this:

<table>
<thead>
<tr>
<th>Step 1</th>
<th>Select the DHCP Server, and expand the extension settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2</td>
<td>Reload the DHCP server for the change to take effect.</td>
</tr>
</tbody>
</table>

Administrator’s User Interface

The BACC administrator’s user interface is an HTML-based application from which you can configure and manage users and devices registered with the system. Refer to these chapters for specific instructions on the use of this user interface:

- Chapter 8, “Understanding the Administrator’s User Interface” describes how to access the BACC administrative user interface and explains the various interface components.
- Chapter 9, “Using the Broadband Access Center for Cable Administrator User Interface” provides instructions for performing administrative activities involving the monitoring of various BACC components.
- Chapter 10, “Configuring Broadband Access Center for Cable” describes those activities that you perform to configure BACC.

Sample User Interface

The BACC product also comes with a Sample User Interface (SUI) application, which is found in Chapter 11, “Configuring and Using the Sample User Interface.” This is used to demonstrate how BACC can be used to perform both self-provisioning and pre-provisioning, and other basic BACC functions in lab testing scenarios. In full BACC deployments, SUI functionality is expected to be provided by billing, OSS, and/or workflow applications.

Caution

The SUI is not intended for use in any live environment and is for demonstration purposes only.