



CHAPTER 2

Broadband Access Center Architecture

This chapter describes the system architecture implemented in this Broadband Access Center (BAC) release.

- Regional Distribution Unit (RDU) that provides:
 - The authoritative data store of the BAC system.
 - Support for processing application programming interface (API) requests.
 - Monitoring of the system's overall status and health.

See [Regional Distribution Unit, page 2-3](#), for additional information.

- Device Provisioning Engines (DPEs) that provide:
 - Interface with customer premises equipment (CPE).
 - Configuration cache.
 - Autonomous operation from the RDU and other DPEs.
 - PacketCable provisioning services.
 - IOS-like command-line interface (CLI) for configuration.

See [Device Provisioning Engines, page 2-4](#), for additional information.

- Client API that provides total client control over system capabilities.
- Cisco Network Registrar servers that provide:
 - Dynamic Host Configuration Protocol (DHCP).
 - Domain Name System (DNS).

See [Network Registrar, page 2-10](#), for additional information.

- Provisioning Groups that provide:
 - Logical grouping of Network Registrar servers and DPEs in a redundant cluster.
 - Redundancy and scalability.

See [Provisioning Groups, page 2-10](#), for additional information.

- A Kerberos server that authenticates PacketCable Media Terminal Adapters (MTAs). See [Key Distribution Center, page 2-11](#), for additional information.

- The BAC process watchdog that provides:
 - Administrative monitoring of all critical BAC processes.
 - Automated process-restart capability.
 - Ability to start and stop BAC component processes.See [BAC Process Watchdog, page 2-14](#), for additional information.
- An SNMP agent that provides:
 - Third-party management systems.
 - SNMP version v2.
 - SNMP Notification.See [SNMP Agent, page 2-13](#), for additional information.
- An administrator user interface that supports:
 - Adding, deleting, modifying and searching for devices.
 - Configuring of global defaults and defining of custom properties.See [Administrator User Interface, page 2-20](#), for additional information.

CPE Registration Modes

Registration modes allow the service provider 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 it. For this reason, the service provider must carefully choose among the standard, promiscuous, roaming, and mixed registration modes.

Standard Mode

When operating in the standard mode (sometimes called the fixed mode), a computer is registered and, when it is behind the correct cable modem, it receives registered access. When it is moved behind a different cable modem, however, it receives unprovisioned access.

Promiscuous Mode

When operating in the promiscuous mode, only DOCSIS modems are 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

When operating in the roaming mode, a registered device receives its assigned service behind any other registered device. For example, this mode permits the use of a laptop moving from location to location and obtaining service from multiple cable modems.

Mixed Mode

When operating in the mixed mode, any mode is used at any time in a single deployment (with different devices).

Regional Distribution Unit

The RDU is the primary server in the BAC provisioning system. You must install the RDU on a server running the Solaris 8 or 9 operating system.

The functions of the RDU include:

- Managing device configuration generation
- Generating configurations for DPEs and distributing them to DPE for caching
- Cooperating with DPEs to keep them up to date
- Processing API requests for all BAC functions
- Managing the BAC system

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

Currently, BAC supports one RDU per installation. To provide failover support, we recommend using clustering software from Veritas or Sun. We also recommend using RAID (Redundant Array of Independent Disks) shared storage in such a setup.

The following sections describe these RDU concepts:

- [Generating Device Configurations, page 2-3](#)
- [Service-Level Selection, page 2-4](#)

Generating Device Configurations

When a device boots, it requests a configuration from BAC and it is this configuration that 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
- Level of service

A configuration can contain DHCP configuration and TFTP files for any device. When you install and boot an unprovisioned device, it is assigned a default technology-specific configuration. You can change the default configuration for each supported technology.

Service-Level Selection

The service-level selection extension point determines the DHCP criteria and the Class of Service that the RDU is to use when generating a configuration for a device. The RDU stores this information for each device in its database. Although a device may have been registered as having to receive one set of DHCP criteria and Class of Service, a second set may actually be selected. The configuration generation extensions look for the selected criteria and Class of Service and use them. Consequently, since the RDU automatic regeneration now knows that a second set of criteria and Class of Service is being used, the device configuration is regenerated if any changes occur to any of the DHCP criteria and the Class of Service.

You can enter service-level selection extension points on the default pages for the specific technologies. For additional information, see [Configuring Defaults, page 11-6](#). By default, these properties are populated with zero or with one of the built-in extensions. Do not modify these extensions unless you are installing your own custom extensions.

Device Provisioning Engines

The Device Provisioning Engine (DPE) communicates with CPE to perform provisioning and management functions.

The RDU generates instructions for the CPE that dictate the actions that the DPE must carry out on the device. These configuration instructions are distributed to the relevant DPE servers, in which they are cached. The configurations are then used during interactions with the CPE to accomplish various tasks.

BAC supports multiple DPEs. You can use multiple DPEs to ensure redundancy and scalability.

The DPE handles all configuration requests, including providing configuration files for devices. It is integrated with the 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:

- Synchronizes with the RDU to retrieve the latest configurations for caching.
- Generates last-step device configuration (for instance, DOCSIS timestamps).
- Provides the DHCP server with instructions controlling the DHCP message exchange.
- Delivers configuration files via TFTP.
- Integrates with Network Registrar.
- Provisions voice technology services.

You must install the DPE on a server that runs the Solaris 8 or 9 operating system. You can configure and manage the DPE using a CLI, which you can access locally or remotely via Telnet. For specific information on the CLI commands that a DPE supports, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

During installation, you must configure for each DPE the:

- Name of the provisioning group to which the DPE belongs. This name determines the logical group of devices that the DPE services.
- IP address and port number of the RDU.

Types of DPEs

This BAC release supports two types of DPEs:

- The traditional hardware device (the DPE-2115 appliance). See [Hardware DPEs, page 2-5](#).
- The software-only Solaris DPE. See [Solaris DPEs, page 2-5](#).

With few exceptions, the commands used on the DPE CLI are identical on hardware and Solaris DPEs. For information on the CLI commands that each DPE supports, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

Hardware DPEs

This BAC release supports the DPE-2115 appliance.

For information on the DPE-2115 device, ports, connectors, and rear panel components, refer to the *Installation and Setup Guide for the Cisco 1102 VLAN Policy Server* 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 DPEs

The Solaris DPE functions in the same way as the hardware DPE, with the exception that it is installed on a computer running the Solaris 8 or 9 operating system.

See these sections for other important information:

- [DPE Licensing, page 2-5](#)
- [TACACS+ and DPE Authentication, page 2-6](#)
- [DPE-RDU Synchronization, page 2-7](#)
- [TFTP Server, page 2-8](#)
- [Provisioning Groups, page 2-10](#)

DPE Licensing

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



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 the release number or type, including those used as part of a BAC lab installation. For additional information, see [Managing License Keys, page 11-30](#).

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

When you delete a registered DPE from the RDU database, a license is freed. Since the DPEs automatically register with the RDU, you must take the DPE offline if the intention is to free up the license. Then, delete the DPE from the RDU database from the administrator user interface.

Deleted DPEs are removed from all the provisioning groups that they belong to 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 remains 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 user interface. You can determine the license state only by examining the DPE and RDU log files (*dpe.log* and *rdulog*).

**Note**

The functions enabled via a specific license continue to operate even when the corresponding license is deleted from the system.

TACACS+ and DPE Authentication

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 TACACS+, a DPE can support multiple users, with each username and login and 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 in to 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 2-1 TACACS+ Service Levels

Mode	Description
Login	User-level commands at <i>router></i> prompt.
Enable	Enable-level commands at <i>router#</i> prompt.

TACACS+ Client Settings

TACACS+ uses a number of properties that are configured from the CLI. For information on these TACACS+-related commands, refer to the *Cisco Broadband Access Center DPE CLI Reference*, 2.7.1.

When TACACS+ is enabled, you must specify either the IP addresses of all TACACS+ servers or their FQDNs with nondefault values.

You 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. This value is the maximum length of time that the TACACS+ client will wait for a TACACS+ server to reply to protocol requests.
- The TACACS+ server number of retries. This value identifies the number of times that the TACACS+ client attempts a valid protocol exchange with a TACACS+ server.

**Note**

These commands are used on both hardware and Solaris DPEs. On the hardware DPE, you can use these commands only in the console mode.

DPE-RDU Synchronization

The DPE-RDU synchronization is a process of automatically updating the DPE cache to be consistent with the RDU. The DPE cache comprises the instruction cache, with instructions for devices, and the file cache, with files required for devices.

Under normal conditions, the RDU generates events containing configuration updates and sends them to all relevant DPEs to keep them up to date. Synchronization is needed if the DPE is missing some events due to connection loss. Such loss could be because of a network issue, the DPE server going down for administrative purposes, or a failure.

Synchronization also covers the special case when the RDU database is restored from backup. In this case, the DPE cache database must be returned to an older state to be consistent with the RDU.

The RDU and DPE synchronization process is automatic and requires no administrative intervention. Throughout the synchronization process, the DPE is still fully capable of performing provisioning and management operations on the CPE.

Synchronization Process

The DPE triggers the synchronization process every time it establishes a connection with the RDU.

When the DPE first starts up, it establishes the connection to the RDU and registers with the RDU to receive updates of configuration changes. The DPE and RDU then monitor the connection using heartbeat message exchanges. When the DPE determines that it has lost its connection to the RDU, it automatically attempts to re-establish it. It continues its attempts with a backoff-retry interval until it is successful.

The RDU also detects the lost connection and stops sending events to the DPE. Since the DPE may miss the update events from the RDU when the connection is down, the DPE performs synchronization every time it establishes a connection with the RDU.

General DPE States

During the process of synchronization, the DPE is in the following states:

1. **Registering**—During the process of connection establishment and registration with the RDU, the DPE is in the *Registering* state.
2. **Synchronizing**—The DPE requests a list of all the configurations it should have from the RDU. This list contains the identifiers for instructions and revision numbers, but not the actual instruction content. By using this list, the DPE determines which configurations in its store are inconsistent (wrong revision number), which ones are missing, and which ones to delete. Throughout the process of obtaining the synchronization list and comparing it to its store, the DPE is in the *Synchronizing* state.
3. **Populating**—Once the DPE determines what to obtain from the RDU, it starts obtaining configurations from the RDU. The DPE only obtains missing or out-of-date configurations. During this process, the DPE is in the *Populating* state.
4. **Ready**—The DPE populates at a fixed rate to ensure that the RDU is not overloaded with its requests. If multiple DPEs in the provisioning group are populating, the population time may be decreased as the requested configurations are sent to all DPEs in the provisioning group. After the DPE finishes populating, it is in the *Ready* state and fully synchronized with the RDU.

You can view the DPE state:

- From the administrator user interface. See [Viewing Device Provisioning Engines, page 10-19](#).
- From the DPE CLI by using the **show dpe** command. Refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

TFTP Server

The integrated TFTP server receives requests for files, including DOCSIS configuration files, both from device and nondevice 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 *BPR_DATA/dpe/tftp* directory.

By default the TFTP server only looks in its cache for a TFTP read. However, if you run the **tftp allow-read-access** command, the TFTP server looks in 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 exists in the cache, the server uses it; otherwise, it sends a request for the file to the RDU.

When you can enable read access from the local file system, directory structure read requests are allowed only from the local file system.



Note

Ensure that you give unique names to all TFTP files instead of differentiating the files by using upper or lowercase. The filename casing is important because the DPE, while looking for a file in its local directory or cache, converts all filenames to lowercase.

DOCSIS Shared Secret

BAC lets you define multiple DOCSIS shared secrets (DSS) for dynamic DOCSIS configuration files only on devices belonging to different cable modem termination systems (CMTS). In this way, a compromised shared secret compromises only a limited number of CMTS instead of every CMTS in the deployment.

Although the DSS can be set for each DPE, you should set it on a provisioning-group basis. Also, ensure that it matches what has been configured for the CMTS in that provisioning group.

**Caution**

Configuring multiple DSS within one provisioning group could, under some conditions, could result in degraded CMTS performance. However, this factor has virtually no effect on BAC.

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 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 DSS takes precedence over the one set from the RDU.

Resetting the DOCSIS Shared Secret

You can reset the DSS if the security of the DSS is compromised or to simply change the shared secret for administrative purposes. To reset the DSS, run the **show running-config** command from the CLI, then copy and paste the DOCSIS shared secret from the configuration that appears back into the DPE configuration. In this way, you can copy what you enter in a Cisco CMTS into the DPE CLI. For additional information, refer to the *Cisco Broadband Access Center DPE CLI Reference*, 2.7.1.

**Note**

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

To change the DSS:

- Step 1** Identify the provisioning group on which you need to reset the DOCSIS shared secret.
- Step 2** Examine the list of DPEs and CMTS 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 change is required to allow cable modems to continue to register until all the DOCSIS configuration files are successfully changed to use the new DSS.
- Step 5** Determine which DPEs were affected and change the DSS on each accordingly.
- Step 6** Confirm that the DOCSIS configuration files are using the new DSS and then remove the compromised secondary shared secret from the CMTS configuration.

Provisioning Groups

A provisioning group is designed to be a logical (typically geographic) grouping of servers that usually consists of one or more DPEs and a failover pair of DHCP servers that can handle the provisioning needs of up to one million devices. Each DPE in a given provisioning group caches identical sets of configurations from the RDU, thus enabling redundancy and load balancing. 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 network operations center.

Provisioning groups enhance the scalability of the BAC deployment by making each provisioning group responsible for only a subset of devices. This partitioning of devices can be along regional groupings or any other policy that the service provider defines.

To scale a deployment, the service provider can:

- Upgrade existing DPE server hardware
- Add DPE servers to a provisioning group
- Add provisioning groups

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.

Network Registrar

Network Registrar provides the DHCP and DNS functionality in BAC.

For additional information on Network Registrar, refer to the *Cisco Network Registrar User's Guide*, 6.2.1; *Cisco Network Registrar CLI Reference*, 6.2.1; and *Cisco Network Registrar Installation Guide*, 6.2.

DHCP

The DHCP server automates the process of configuring IP addresses on IP networks. The 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 feature 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 main and backup server. Under normal circumstances, the main server performs all DHCP functions. If the main server becomes unavailable, the backup server takes over. In this way, DHCP failover prevents loss of access to the DHCP service if the main server fails.

DNS

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

Lease Reservation

BAC lease reservation works with Network Registrar's Central Configuration Management (CCM) to assign a device with a static IP address during provisioning.

**Note**

This feature is only supported when Network Registrar, version 6.1.2.3 or later, is in use with the Regional CCM feature that is deployed. The Lease Reservation feature in BAC 2.7.1 works only in scenarios involving a single Network Registrar DHCP server with no failover configured. This feature is not supported in cases involving failover DHCP servers. Cisco plans to add more functional use of this feature in a later version of BAC.

When you provision a new device, BAC determines whether the IP address is specified and then determines which Network Registrar server identifies it as 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 that BAC supports, and:

- Lets you add and remove IP address reservations from the BAC administrator user interface. See [Managing Devices, page 10-13](#).
- 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.

You must configure the CCM address, port, username, and password before BAC can implement lease reservation. These parameters are set from the RDU Defaults page. Changes are dynamic and take effect immediately. See [RDU Defaults, page 11-19](#), for information on these configuration parameters.

**Note**

The lease reservation function is disabled by default and times out if the CCM server cannot be reached for a specified duration.

Key Distribution Center

The Key Distribution Center (KDC) authenticates PacketCable MTAs and also grants service tickets to MTAs. As such, it must check the MTA certificate, and provide its own certificates so that the MTA can authenticate the KDC. It also communicates with the DPE (the provisioning server) to validate that the MTA is provisioned on the network.

**Note**

The KDC is supported on multiprocessor computers.

The certificates used to authenticate the KDC are not shipped with BAC. 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. For additional information, see [Using the PKCert.sh Tool, page 13-5](#).

**Caution**

The KDC does not function if the certificates are not installed.

The KDC also requires a license to function. Obtain a KDC license from your Cisco representative and install it in the correct directory. For details on how to install the license, see [KDC Licenses, page 5-9](#).

The KDC has several default properties that are populated during a BAC installation into the *BAC_home/kdc/solaris/kdc.ini* properties file. You can edit this file to change values as operational requirements dictate. For detailed information, see [Default KDC Properties, page 5-7](#).

The KDC also supports the management of multiple realms. For details on configuring additional realms, see [Multiple Realm Support, page 5-10](#).

BAC MIBs

BAC supports several different MIBs. These include:

- CISCO-BACC-RDU-MIB
- CISCO-BACC-DPE-MIB
- CISCO-APPLIANCE-MIB
- CISCO-BACC-SERVER-MIB

For details on each MIB, see [MIB Support, page 2-13](#).

[Table 2-2](#) summarizes MIB support for each BAC component.

Table 2-2 *BAC-Supported MIBs*

Component	MIBs Supported
Solaris DPE	CISCO-BACC-SERVER-MIB
	CISCO-BACC-DPE-MIB
Hardware DPE	RFC1213 - MIB II
	CISCO-APPLIANCE-MIB
	CISCO-BACC-SERVER-MIB
	CISCO-BACC-DPE-MIB
RDU	CISCO-BACC-SERVER-MIB
	CISCO-BACC-RDU-MIB

BAC Agents

This section describes BAC 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:

- [SNMP Agent, page 2-13](#)
- [BAC Process Watchdog, page 2-14](#)

SNMP Agent

BAC provides basic SNMP v2-based monitoring of the RDU and DPE servers. The BAC SNMP agents support SNMP informs and traps, collectively called notifications. You can configure the SNMP agent on the DPE using `snmp-server` CLI commands, and on the RDU using the SNMP configuration command-line tool.

For additional information on the SNMP configuration command-line tool, see [Using the `snmpAgentCfgUtil.sh` Tool, page 13-15](#). For additional information on the DPE CLI, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

MIB Support

The SNMP agent supports the CISCO-BACC-SERVER-MIB. This MIB defines the managed objects that are common to all servers on BAC. This MIB supports the monitoring of multiple BAC servers when they are installed on the same device. The `ciscoBaccServerStateChanged` notification is generated every time a server state change occurs.

The RDU SNMP agent supports the CISCO-BACC-RDU-MIB, which defines managed objects for the RDU. This MIB defines statistics related to the state of the RDU and the statistics on the communication interface between the RDU and DPE and between the RDU and Network Registrar.

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

The DPE SNMP agent supports the CISCO-BACC-DPE-MIB, which defines 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.

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 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 is MIB-II.

**Note**

For a description of all objects, refer to the corresponding MIBs files in the *BAC_home/rdu/mibs* directory.

BAC Process Watchdog

The BAC process watchdog is an administrative agent that monitors the runtime health of all BAC processes. This watchdog process ensures that if a process stops unexpectedly, it is automatically restarted. One instance of the BAC process watchdog runs on every system which runs BAC components.

You can use the BAC process watchdog as a command-line tool to start, stop, restart, and determine the status of any monitored processes.

If a monitored application fails, it is restarted automatically. If, for any reason, the restart process also fails, the BAC process watchdog server waits a prescribed length of time before attempting to restart again.

**Note**

You do not have to use the BAC process watchdog and the SNMP agent to monitor Network Registrar extensions.

The period between restart attempts starts at 1 second and increases exponentially with every subsequent attempt 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:

1. Process A fails.
2. The BAC process watchdog server attempts to restart it and the first restart fails.
3. The BAC process watchdog server waits 2 seconds and attempts to restart the process and the second restart fails.
4. The BAC process watchdog server waits 4 seconds and attempts to restart the process and the third restart fails.
5. The BAC process watchdog server waits 16 seconds and attempts to restart the process.

Using the BAC Process Watchdog from the Command Line

The BAC process watchdog automatically starts whenever the system boots up. Consequently, this watchdog also starts those BAC system components installed on the same system. You can control the BAC watchdog through a simple command-line utility by running the `/etc/init.d/bprAgent` command.

Table 2-3 describes the command-line interface commands available for use with the BAC watchdog process.

Table 2-3 BAC CLI Commands

Command	Description
bprAgent start	Starts the BAC watchdog agent, including all monitored processes.
bprAgent stop	Stops the BAC watchdog agent, including all monitored processes.
bprAgent restart	Restarts the BAC watchdog agent, including all monitored processes.
bprAgent status	Gets the status of the BAC watchdog agent, including all monitored processes.
bprAgent start <i>process-name</i>	Starts one particular monitored process. The value <i>process-name</i> identifies that process.

Table 2-3 BAC CLI Commands (continued)

Command	Description
bprAgent stop <i>process-name</i>	Stops one particular monitored process. The value <i>process-name</i> identifies that process.
bprAgent restart <i>process-name</i>	Restarts one particular monitored process. The value <i>process-name</i> identifies that process.
bprAgent status <i>process-name</i>	Gets the status of one particular monitored process. The value <i>process-name</i> identifies that process.

The *process-name* mentioned in this table can be:

- **rdu**—Specifies the RDU server.
- **dpe**—Specifies the DPE server.
- **kdc**—Specifies the KDC server.
- **snmpAgent**—Specifies the SNMP agent.
- **tomcat**—Specifies the administrator and sample user interfaces.
- **cli**—Specifies the DPE command-line interface.

**Note**

When the Solaris operating system is rebooted, the BAC process watchdog is first stopped, allowing BAC servers to shut down properly. To shut down or reboot the operating system, use the Solaris **shutdown** command. Remember, the Solaris **reboot** command does not execute application shutdown hooks and kills BAC processes rather than shuts them down. While this action is not harmful to BAC, it may delay server startup and skew certain statistics and performance counters.

The events that trigger an action in the BAC watchdog daemon, including process crashes and restarts, are logged in a log file, *BPR_HOME/agent/logs/agent.log*. The watchdog daemon also logs important events to syslog under the standard `local6` facility.

Logging

Logging of events is performed at the RDU and the DPE, and in some unique situations, DPE events are additionally logged at the RDU to give them higher visibility. Log files are stored in their own log directories and can be examined by using any text processor. You can compress the files for easier e-mailing to the Cisco Technical Assistance Center (TAC) or system integrators for troubleshooting and fault resolution. You can also access the RDU and the DPE logs from the administrator user interface.

Log Levels and Structures

The log file structure, illustrated in [Example 2-1](#), 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 BAC.

- Sub-Facility—This identifies the BAC subsystem or component.
- Severity Level—The logging system defines seven levels of severity (as described in [Table 2-4](#)) that are used to identify the urgency with which you might want to address log issues. The process of configuring these severity levels is described in [Configuring Severity Levels, page 2-17](#).

Table 2-4 **Severity Levels**

Log Level	Description
0-Emergency	System unstable. Sets the logging function to save all emergency messages.
1-Alert	Immediate action needed. Sets the logging function to save all activities that need immediate action and those of a more severe nature.
2-Critical	Critical conditions exist. Sets the logging function to save all error messages and those of a more severe nature.
3-Error	Error conditions exist. Sets the logging function to save all error messages and those of a more severe nature.
4-Warning	Warning conditions exist. Sets the logging function to save all warning messages and those of a more severe nature.
5-Notification	A normal, but significant, condition exists. Sets the logging function to save all notification messages and those of a more severe nature.
6-Information	Informational messages. Sets the logging function to save all logging messages available.
Note Another level known as 7-Debug is used exclusively by Cisco for debugging purposes. Do not use this level except at the direction of the Cisco TAC.	

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

Example 2-1 **Sample Log File**

Domain Name	Data and Time	Facility	Sub-facility	Severity Level	Msg ID	Message
bacc.cisco.com:	2007 3 16 03:06:11 EST:	BPR-	RDU-	5	0236:	BPR Regional Distribution Unit starting up
bacc.cisco.com:	2007 3 16 03:06:15 EST:	BPR-	RDU-	5	0566:	Initialized API defaults
bacc.cisco.com:	2007 3 16 03:06:15 EST:	BPR-	RDU-	5	0567:	Initialized CNR defaults
bacc.cisco.com:	2007 3 16 03:06:15 EST:	BPR-	RDU-	5	0568:	Initialized server defaults
bacc.cisco.com:	2007 3 16 03:06:18 EST:	BPR-	RDU-	5	0570:	Initialized DOCSIS defaults
bacc.cisco.com:	2007 3 16 03:06:18 EST:	BPR-	RDU-	5	0571:	Initialized computer defaults
bacc.cisco.com:	2007 3 16 03:06:19 EST:	BPR-	RDU-	5	0573:	Initialized CableHome defaults
bacc.cisco.com:	2007 3 16 03:06:19 EST:	BPR-	RDU-	5	0572:	Initialized PacketCable defaults
bacc.cisco.com:	2007 3 16 03:06:19 EST:	BPR-	RDU-	5	0569:	Created default admin user
bacc.cisco.com:	2007 3 16 03:06:19 EST:	BPR-	RDU-	5	0574:	Loaded 6 license keys
bacc.cisco.com:	2007 3 16 03:06:20 EST:	BPR-	RDU-	5	0575:	Database initialization completed in 471 msec

Example 2-1 Sample Log File (continued)

Domain Name	Data and Time	Facility	Sub-facility	Severity Level	Msg ID	Message
bacc.cisco.com:	2007 3 16 03:06:25 EST:	BPR-	RDU-	3	0015:	Unable to locate manifest file
bacc.cisco.com:	2007 3 16 03:06:28 EST:	BPR-	RDU-	3	0280:	Command error

Configuring Severity Levels

You can configure the severity levels of logging for both the RDU and the DPE to suit your specific requirements. For example, the severity 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 severity 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.

**Note**

The KDC is not considered in this log file.

To configure the severity level on the DPE, use the **log level** command from the DPE command line. For detailed information, refer to the *Cisco Broadband Access Center DPE CLI Reference*, 2.7.1.

To configure the log level tool on the RDU, see [Using the RDU Log Level Tool, page 13-2](#).

Rotating Log Files

All log files are numbered and rolled over based on a configured maximum file size. The default maximum file size is 10 MB. (To configure the maximum file size from the API, use the `ServerDefaultsKeys.SERVER_LOG_MAXSIZE` property.) Once a log file touches the configured limit, the data is rolled over to another file. This file is renamed in the `XXX.N.log` format, where:

- `XXX`—Specifies the name of the log file.
- `N`—Specifies any value between 1 and 100.

**Note**

The RDU and DPE servers store up to 100 log files at a given time. For a list of log files in these servers, see subsequent sections.

For example, once `rdu.log` reaches the 10-MB limit, it is renamed as `rdu.1.log`. With every 10-MB increase in file size, the latest file is renamed as `rdu.2.log`, `rdu.3.log`, and so on. So, the `rdu.4.log` file will contain data more recent than `rdu.7.log`. The latest log information, however, is always stored in `rdu.log`.

RDU Logs

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

- **rdu.log**—Records RDU processing according to the configured default severity level. (For instructions on setting the default log levels, see [Setting the RDU Log Level, page 13-3](#).)
- **audit.log**—Records high-level changes to the BAC configuration or functionality including the user who made the change.

When you enable logging of informational messages (6-Information), the RDU logs additional messages which expose batch-processing operations. These messages also contain information on elapsed time and rate.

Viewing the *rdu.log* File

You can use any text processor to view the *rdu.log* file. In addition, you can view the log file from the administrator user interface. To view the file:

-
- | | |
|---------------|--|
| Step 1 | Choose the RDU tab under Servers . |
| Step 2 | The View Regional Distribution Unit Details page appears. Click the View Details icon (🔍) corresponding to RDU Log File. |
| Step 3 | The View Log File Contents page appears, displaying data from <i>rdu.log</i> . |
-

Viewing the *audit.log* File

You can use any text processor to view the *audit.log* file. In addition, you can view the log file from the administrator user interface. To view the file:

-
- | | |
|---------------|--|
| Step 1 | Choose the RDU tab under Servers . |
| Step 2 | The View Regional Distribution Unit Details page appears. Click the View Details icon corresponding to Audit Log File. |
| Step 3 | The View Log File Contents page appears, displaying data from <i>audit.log</i> . |
-

DPE Log

The DPE maintains a *dpe.log* file in the *BAC_data/dpe/logs* directory. The file contains records of 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 resides in the *BAC_data/dpe/logs* directory. For PacketCable command usage, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

**Note**

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

You can use any text viewer to view the dpe.log file. In addition, you can use the **show log** command from the DPE CLI. For additional information, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

You can also view the DPE log file using the BAC administrator user interface. To view the file:

-
- Step 1** Choose **Servers > DPEs**.
 - Step 2** Click the link of the DPE whose log file you want to view.
 - Step 3** The View Device Provisioning Engines Details page appears. To view the contents of the dpe.log file, click the View Details icon against the DPE Log File in the Log Files area.
-

Network Registrar Logs

BAC generates log messages from Network Registrar's DHCP server extensions. The DHCP server log resides in the *NetworkRegistrar_home/name_dhcp_1_log* directory; *NetworkRegistrar_home* is a variable and is specific to the value that you enter. The default location for the DHCP server log file is */var/nwreg2/local/logs/name_dhcp_1_log*.

The log messages emitted via the DHCP server extensions are based on the extension trace level setting. You can set values (described in [Table 2-5](#)) at the trace level; the number you set makes that number the current setting of the **extension-trace-level** attribute for all extensions.

Table 2-5 DHCP Server Extension Trace Levels

Level	Description
0	Logs error and warning conditions. Sets the extensions to emit all error and warning messages and those of a more severe nature.
1	Logs server interactions, which include configuration instructions obtained from the DPE and instruction generation requests that are forwarded to the RDU.
2	Logs processing details, which include individual configuration commands and attribute values forwarded in instruction generation requests.
3	Logs internal processing for extensions debugging, which includes hexadecimal dumps of messages.
4	Logs debugging of extension background operations, which include polling of DPE status.

You can change the extension trace level by using the Network Registrar Web UI. To change the level:

-
- Step 1** Open the Network Registrar local Web UI.
 - Step 2** From the menu, click **DHCP**, then **DHCP Server**.
 - Step 3** Click the Local DHCP Server link.
 - Step 4** On the Edit DHCP Server page, expand the Extensions attribute category.

Step 5 Set the **extension-trace-level** value, then click **Modify Server**.

Step 6 Reload the DHCP server.

**Note**

For detailed information on logging performed by the DHCP server, refer to the *Cisco Network Registrar User's Guide*, 6.2.1.

Administrator User Interface

The BAC administrator user interface is a web-based application for central management of the BAC system. You can use this system to:

- Configure global defaults
- Define custom properties
- Set up Class of Service
- Add and edit device information
- Group devices
- View server status and server logs
- Manage users

Refer to these chapters for specific instructions on how to use this interface:

- [Understanding the Administrator User Interface, page 9-1](#), describes how to access and configure the BAC administrator user interface.
- [Using the Administrator User Interface, page 10-1](#), provides instructions for performing administrative activities involving the monitoring of various BAC components.
- [Configuring Broadband Access Center, page 11-1](#), describes tasks that you perform to configure BAC.

Sample User Interface

BAC comes with a web-based Sample User Interface (SUI), which is explained in [Configuring and Using the Sample User Interface, page 12-1](#). This interface demonstrates how you can use BAC to perform self-provisioning and preprovisioning, and other basic BAC functions in lab scenarios. In full BAC deployments, the SUI functionality is expected to be provided by billing, OSS, workflow applications, or a combination of all three.

**Caution**

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