



Cisco Broadband Access Center Administrator Guide

Release 2.7.1
November 2008

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
<http://www.cisco.com>
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Text Part Number: OL-4409-03

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

CCVP, the Cisco logo, and the Cisco Square Bridge logo are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn is a service mark of Cisco Systems, Inc.; and Access Registrar, Aironet, BPX, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Enterprise/Solver, EtherChannel, EtherFast, EtherSwitch, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, LightStream, Linksys, MeetingPlace, MGX, Networking Academy, Network Registrar, *Packet*, PIX, ProConnect, ScriptShare, SMARTnet, StackWise, The Fastest Way to Increase Your Internet Quotient, and TransPath are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

All other trademarks mentioned in this document or Website are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0705R)

Any Internet Protocol (IP) addresses used in this document are not intended to be actual addresses. Any examples, command display output, and figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses in illustrative content is unintentional and coincidental.

Book Title

© 2002 - 2008 Cisco Systems, Inc. All rights reserved.



CONTENTS

Preface xxi

- Audience i-xxi
- How This Guide Is Organized i-xxii
- Conventions i-xxiii
- Product Documentation i-xxiii
- Related Documentation i-xxiv
- Obtaining Documentation, Obtaining Support, and Security Guidelines i-xxv

CHAPTER 1

Broadband Access Center Overview 1-1

- Features and Benefits 1-1
- Supported Technologies 1-1
 - DOCSIS High-Speed Data 1-2
 - PacketCable Secure Voice Service 1-2
 - PacketCable Basic Voice Service 1-2
 - CableHome 1-2

CHAPTER 2

Broadband Access Center Architecture 2-1

- CPE Registration Modes 2-2
 - Standard Mode 2-2
 - Promiscuous Mode 2-2
 - Roaming Mode 2-2
 - Mixed Mode 2-3
- Regional Distribution Unit 2-3
 - Generating Device Configurations 2-3
 - Service-Level Selection 2-4
- Device Provisioning Engines 2-4
 - Types of DPEs 2-5
 - Hardware DPEs 2-5
 - Solaris DPEs 2-5
 - DPE Licensing 2-5
 - TACACS+ and DPE Authentication 2-6
 - TACACS+ Privilege Levels 2-6
 - TACACS+ Client Settings 2-6
 - DPE-RDU Synchronization 2-7

- Synchronization Process 2-7
- General DPE States 2-8
- TFTP Server 2-8
- DOCSIS Shared Secret 2-9
- Provisioning Groups 2-10
- Network Registrar 2-10
 - DHCP 2-10
 - DNS 2-11
 - Lease Reservation 2-11
- Key Distribution Center 2-11
- BAC MIBs 2-12
- BAC Agents 2-13
 - SNMP Agent 2-13
 - BAC Process Watchdog 2-14
- Logging 2-15
 - Log Levels and Structures 2-15
 - Configuring Severity Levels 2-17
 - Rotating Log Files 2-17
 - RDU Logs 2-18
 - Viewing the *rdi.log* File 2-18
 - Viewing the *audit.log* File 2-18
 - DPE Log 2-18
 - Network Registrar Logs 2-19
- Administrator User Interface 2-20
- Sample User Interface 2-20

CHAPTER 3

Configuration Workflows and Checklists 3-1

- Component Workflows 3-1
 - RDU Checklist 3-1
 - Hardware DPE Checklist 3-2
 - Solaris DPE Checklist 3-3
 - Network Registrar Checklist 3-5
- Technology Workflows 3-5
 - DOCSIS Checklist 3-6
 - PacketCable Checklists 3-6
 - PacketCable Secure 3-6
 - PacketCable Basic 3-10
 - Non-Secure CableHome Provisioning Checklist 3-11

CHAPTER 4**DOCSIS Configuration 4-1**

- DOCSIS Workflow 4-1
- Using MIBs with Dynamic DOCSIS Templates 4-3
- BAC Features for DOCSIS Configurations 4-4
 - Dynamic Configuration TLVs 4-4
 - DPE TFTP IP Validation 4-4
 - DOCSIS 1.0, 1.1, 2.0 Support 4-5
 - Dynamic DOCSIS Version Selection 4-5
- Troubleshooting DOCSIS Networks 4-6

CHAPTER 5**PacketCable Voice Configuration 5-1**

- PacketCable Secure eMTA Provisioning 5-1
 - BAC PacketCable Secure Provisioning Flow 5-1
- KDC in Provisioning PacketCable Secure eMTAs 5-7
 - Default KDC Properties 5-7
 - KDC Certificates 5-9
 - KDC Licenses 5-9
 - Multiple Realm Support 5-10
 - Configuring the KDC for Multiple Realms 5-11
 - Authoring Template for Provisioning Devices in Multiple Realms 5-25
- PacketCable Basic eMTA Provisioning 5-28
 - PacketCable TLV 38 and MIB Support 5-29
 - SNMP v2C Notifications 5-29
- Euro PacketCable 5-29
 - Euro PacketCable MIBs 5-30
 - Configuring Euro PacketCable MIBs 5-30

CHAPTER 6**Troubleshooting PacketCable eMTA Provisioning 6-1**

- Components 6-1
 - eMTA 6-2
 - DHCP Server 6-2
 - DNS Server 6-2
 - KDC 6-2
 - PacketCable Provisioning Server 6-3
 - Call Management Server 6-3
- Key Variables 6-3
 - Certificates 6-3
 - Scope-Selection Tags 6-4

- MTA Configuration File 6-4
- Troubleshooting Tools 6-4
 - Logs 6-5
 - Ethereal, SnifferPro, or Other Packet Capture Tools 6-5
- Troubleshooting Scenarios 6-5
- Certificate Trust Hierarchy 6-9
 - Certificate Validation 6-10
 - MTA Device Certificate Hierarchy 6-10
 - MTA Root Certificate 6-11
 - MTA Manufacturer Certificate 6-11
 - MTA Device Certificate 6-12
 - MTA Manufacturer Code Verification Certificates 6-13
- CableLabs Service Provider Certificate Hierarchy 6-13
 - CableLabs Service Provider Root Certificate 6-14
 - Service Provider CA Certificate 6-14
 - Local System CA Certificates 6-15
 - Operational Ancillary Certificates 6-16
- Certificate Revocation 6-19
- Code Verification Certificate Hierarchy 6-19
 - Common CVC Requirements 6-19
 - CableLabs Code Verification Root CA Certificate 6-20
 - CableLabs Code Verification CA Certificate 6-20
 - Manufacturer Code Verification Certificate 6-21
 - Service Provider Code Verification Certificate 6-21
 - Certificate Revocation Lists for CVCs 6-22

CHAPTER 7

CableHome Configuration 7-1

- Non-Secure CableHome Provisioning Flow 7-1
- Configuring CableHome 7-3
 - Configuring Network Registrar 7-3
 - Configuring the RDU 7-3
 - Configuring CableHome WAN-MAN 7-4
 - Configuring CableHome WAN-Data 7-4
- Configuring the DPE 7-4

CHAPTER 8

Configuration Templates Management 8-1

- Developing Template Files 8-1
- Template Grammar 8-2
 - Comments 8-2

Includes	8-3
Options	8-3
Instance Modifier	8-4
SNMP VarBind	8-5
DOCSIS MIBs	8-5
PacketCable MIBs	8-6
CableHome MIBs	8-6
Macro Variables	8-6
Adding SNMP TLVs	8-8
Adding SNMP TLVs Without a MIB	8-8
Adding SNMP TLVs With Vendor-Specific MIBs	8-9
Encoding Types for Defined Options	8-12
BITS Value Syntax	8-14
OCTETSTRING Syntax	8-14
DOCSIS Option Support	8-15
PacketCable Option Support	8-25
Non-Secure CableHome Option Support	8-26
Using the Configuration File Utility	8-27
Running the Configuration File Utility	8-28
Adding a Template to BAC	8-29
Converting a Binary File Into a Template File	8-30
Testing Template Processing for a Local Template File	8-31
Testing Template Processing for an External Template File	8-32
Testing Template Processing for a Local Template File and Adding Shared Secret	8-33
Specifying Macro Variables at the Command Line	8-34
Specifying a Device for Macro Variables	8-35
Specifying Output to a Binary File	8-36
Viewing a Local Binary File	8-37
Viewing an External Binary File	8-38
Activating PacketCable Basic Flow	8-39

CHAPTER 9**Understanding the Administrator User Interface 9-1**

Configuring the Administrator User Interface	9-1
Accessing the Administrator User Interface	9-2
Logging In	9-2
Logging Out	9-5
Studying the Administrator User Interface	9-5
Understanding the Icons	9-6

CHAPTER 10

Using the Administrator User Interface 10-1

- User Management **10-1**
 - Administrator **10-1**
 - Read/Write User **10-2**
 - Read-Only User **10-2**
 - Adding a New User **10-2**
 - Modifying Users **10-3**
 - Deleting Users **10-3**
- Device Management **10-4**
 - Manage Devices Page **10-4**
 - Searching for Devices **10-5**
 - Device Management Controls **10-7**
 - Viewing Device Details **10-9**
 - Managing Devices **10-13**
 - Adding Device Records **10-14**
 - Modifying Device Records **10-14**
 - Deleting Devices **10-15**
 - Regenerating Device Configurations **10-15**
 - Relating and Unrelating Devices **10-16**
 - Resetting Devices **10-16**
 - Unregistering a Device **10-16**
- Node Management **10-16**
 - Managing Node Types **10-16**
 - Adding a Node Type **10-17**
 - Modifying Node Types **10-17**
 - Deleting Node Types **10-18**
 - Managing Nodes **10-18**
 - Adding a New Node **10-18**
 - Modifying a Node **10-18**
 - Deleting Nodes **10-18**
 - Relating/Unrelating Node Types to Nodes **10-19**
 - Viewing Node Details **10-19**
- Viewing Servers **10-19**
 - Viewing Device Provisioning Engines **10-19**
 - Viewing Network Registrar Extension Points **10-23**
 - Viewing Provisioning Groups **10-25**
 - Viewing Regional Distribution Unit Details **10-25**

CHAPTER 11

Configuring Broadband Access Center	11-1
Configuring Class of Service	11-1
Adding a Class of Service	11-3
Modifying a Class of Service	11-4
Deleting a Class of Service	11-5
Configuring Custom Properties	11-5
Configuring Defaults	11-6
Selecting Configuration Options	11-6
ATA 186 Defaults	11-7
ATA 188 Defaults	11-8
CableHome WAN Defaults	11-8
CableHome WAN-Data Defaults	11-10
CableHome WAN-MAN Defaults	11-11
Computer Defaults	11-12
DOCSIS Defaults	11-13
Network Registrar Defaults	11-15
PacketCable Defaults	11-17
RDU Defaults	11-19
System Defaults	11-21
Gateway (xGCP) Control Protocol Defaults	11-23
Configuring DHCP Criteria	11-24
Adding DHCP Criteria	11-24
Modifying DHCP Criteria	11-25
Deleting DHCP Criteria	11-25
Managing External Files	11-26
Adding External Files	11-27
Viewing External Files	11-27
Replacing External Files	11-29
Exporting External Files	11-29
Deleting External Files	11-30
Managing License Keys	11-30
Adding and Modifying a License	11-31
Deleting a License	11-31
Managing RDU Extensions	11-32
Writing a New Class	11-33
Installing RDU Custom Extension Points	11-34
Viewing RDU Extensions	11-34
Publishing Provisioning Data	11-35
Publishing Datastore Changes	11-35

- Modifying Publishing Plug-In Settings 11-36
- Configuring SRV Records in the Network Registrar DNS Server 11-36
- Configuring SNMPv3 Cloning on the RDU and DPE for Secure Communication with PacketCable MTAs 11-37
 - Creating the Key Material and Generating the Key 11-37
- Automatic FQDN Generation 11-38
 - Automatically Generated FQDN Format 11-38
 - Properties for Automatically Generated FQDNs 11-38
 - FQDN Validation 11-39
 - Sample Automatic FQDN Generation 11-39

CHAPTER 12

Configuring and Using the Sample User Interface 12-1

- What is the Sample User Interface? 12-1
- Accessing the Sample User Interface 12-2
- Sample User Interface Configuration Options 12-2
 - Class of Service 12-2
 - Promiscuous Mode 12-2
 - Selecting an Internet Service Provider 12-3
 - Using the Technician Login 12-3
 - Administrative Access Levels 12-3
- Subscriber Provisioning Examples 12-3
 - Standard Customer Premise Equipment Registration 12-4
 - Provisioning a New Cable Modem and a New Computer 12-4
 - Provisioning a New Computer with an Existing Cable Modem 12-4
 - Altering an Existing Computer ISP 12-5
 - Promiscuous Customer Premises Equipment Registration 12-5
 - Provisioning a New Cable Modem and a New Computer 12-5
 - Provisioning an Existing Cable Modem and a New Computer 12-6
- Administrator Provisioning Examples 12-6
 - Searching for Accounts 12-6
 - Searching by Account Number 12-6
 - Searching by IP Address 12-6
 - Searching by MAC Address 12-6
 - Maintaining Accounts 12-7
 - Registering a New Account 12-7
 - Managing Class of Service 12-7
 - Managing Cable Modems 12-8
 - Managing Computers 12-8
 - Deleting an Account 12-8

Sample sampleui.properties File 12-9

CHAPTER 13
Support Tools and Advanced Concepts 13-1

BAC Tools 13-1

Using the RDU Log Level Tool 13-2

 Setting the RDU Log Level 13-3

 Viewing the Current Log Level of RDU 13-4

Using the PKCert.sh Tool 13-5

 Running the PKCert Tool 13-5

 Creating a KDC Certificate 13-6

 Validating the KDC Certificates 13-7

 Setting the Log Level for Debug Output 13-8

Using the KeyGen Tool 13-11

Using the changeNRProperties.sh Tool 13-13

Using the snmpAgentCfgUtil.sh Tool 13-15

 Adding a Host 13-16

 Deleting a Host 13-16

 Adding an SNMP Agent Community 13-17

 Deleting an SNMP Agent Community 13-17

 Starting the SNMP Agent 13-18

 Stopping the SNMP Agent 13-18

 Configuring an SNMP Agent Listening Port 13-18

 Changing the SNMP Agent Location 13-19

 Setting Up SNMP Contacts 13-19

 Displaying SNMP Agent Settings 13-20

 Specifying SNMP Notification Types 13-20

Using the disk_monitor.sh Tool 13-21

Troubleshooting Devices by MAC Address 13-21

 Relating a Device to a Node 13-22

 Viewing a List of the Devices in Troubleshooting Mode 13-23

CHAPTER 14
Database Management 14-1

Understanding Failure Resiliency 14-1

Database Files 14-2

 Database Storage File 14-2

 Database Transaction Log Files 14-2

 Automatic Log Management 14-2

 Miscellaneous Database Files 14-3

- Disk Space Requirements 14-3
 - Handling Out of Disk Space Conditions 14-3
- Backup and Recovery 14-4
 - Database Backup 14-4
 - Database Recovery 14-5
 - Database Restore 14-6
- Changing Database Location 14-7
- RDU Database Migration 14-8

APPENDIX A

Alert and Error Messages A-1

- Syslog Alert Messages A-1
 - Message Format A-1
 - RDU Alerts A-2
 - Solaris DPE Alerts A-3
 - Watchdog Alerts A-5
 - Network Registrar Extension Point Alerts A-6
- RDU Error Messages with CCM A-7
 - [OBJECT_EXISTS] A-8
 - [RemoveReservation: NOT_FOUND] A-8
 - [AddReservation: INVALID_PARENT] A-8
 - [AddReservation: INVALID_SECOND_PARENT] A-8
 - [AddReservation: FORWARD_FAILED] A-9
 - Example Case #1 A-9
 - Example Case #2 A-9
 - [AddReservation: AX_ETIME] A-9
 - [AddReservation: INVALID_OBJECT] A-10
 - Selection-criteria exclusion tags will be ignored A-10
 - [AX_EIO] A-10
 - [AX_EPIPE] A-10

APPENDIX B

PacketCable DHCP Options to BAC Properties Mapping B-1

- Option 122 and BAC Property Comparison B-1
- Option 177 and BAC Property Comparison B-2

APPENDIX C

API Use Cases C-1

- Use Cases C-1
- Self-Provisioned Modem and Computer in Fixed Standard Mode C-2
- Adding a New Computer in Fixed Standard Mode C-5

Disabling a Subscriber	C-7
Preprovisioning Modems/Self-Provisioned Computers	C-9
Modifying an Existing Modem	C-11
Unregistering and Deleting a Subscriber's Devices	C-12
Self-Provisioning First-Time Activation in Promiscuous Mode	C-14
Bulk Provisioning 100 Modems in Promiscuous Mode	C-17
Preprovisioning First-Time Activation in Promiscuous Mode	C-19
Replacing an Existing Modem	C-20
Adding a Second Computer in Promiscuous Mode	C-21
Self-Provisioning First-Time Activation with NAT	C-21
Adding a New Computer Behind a Modem with NAT	C-23
Move Device to Another DHCP Scope	C-24
Log Device Deletions Using Events	C-25
Monitoring an RDU Connection Using Events	C-26
Logging Batch Completions Using Events	C-27
Getting Detailed Device Information	C-27
Searching Using the Default Class of Service	C-28
Retrieving Devices Matching a Vendor Prefix	C-30
Preprovisioning PacketCable eMTA	C-32
SNMP Cloning on PacketCable eMTA	C-34
Incremental Provisioning of PacketCable eMTA	C-35
Preprovisioning DOCSIS Modems with Dynamic Configuration Files	C-37
Optimistic Locking	C-39
Temporarily Throttling a Subscriber's Bandwidth	C-40
Preprovisioning CableHome WAN-MAN	C-41
CableHome with Firewall Configuration	C-43
Retrieving Device Capabilities for CableHome WAN-MAN	C-45
Self-Provisioning CableHome WAN-MAN	C-47
Lease Reservation Use Cases	C-49
API Calls Affected by the Lease Reservation Feature	C-50
Bringing a Device Online Using IP Address Provided by Service Provider	C-50
Removing and Re-Creating a Reservation	C-51
Assigning a New Device with an Old Device's IP Address	C-52
Removing a Reservation and Assigning a New IP Address	C-53
Rebooting a Device with the Same IP Address	C-54
Removing a Device from BAC	C-55

[A Submitted Batch Fails when BAC Uses CCM](#) C-56
[A Submitted Batch Fails when BAC Does Not Use CCM](#) C-56

GLOSSARY

INDEX



FIGURES

<i>Figure 4-1</i>	DOCSIS Provisioning Flow	4-1
<i>Figure 5-1</i>	Embedded-MTA Secure Power-On Provisioning Flow	5-2
<i>Figure 5-2</i>	Provisioning Motorola MTA—Device Details	5-21
<i>Figure 5-3</i>	Provisioning Linksys MTA—Device Details	5-22
<i>Figure 5-4</i>	Provisioning SA MTA—Device Details	5-23
<i>Figure 6-1</i>	PacketCable Certificate Hierarchy	6-9
<i>Figure 7-1</i>	Non-Secure CableHome Flow	7-1
<i>Figure 9-1</i>	Login Page	9-3
<i>Figure 9-2</i>	Main Menu Page	9-4
<i>Figure 9-3</i>	Administrator User Interface	9-5
<i>Figure 10-1</i>	Manage Users Page	10-2
<i>Figure 10-2</i>	Manage Devices Page	10-4
<i>Figure 10-3</i>	View Device Details Page	10-10
<i>Figure 10-4</i>	Manage Nodes Page	10-17
<i>Figure 10-5</i>	View Device Provisioning Engines Details Page	10-20
<i>Figure 10-6</i>	View Network Registrar Extension Point Details Page	10-23
<i>Figure 10-7</i>	View Provisioning Group Details Page	10-25
<i>Figure 10-8</i>	View Regional Distribution Unit Details Page	10-26
<i>Figure 11-1</i>	Manage Class of Service Page	11-2
<i>Figure 11-2</i>	Configure Defaults—ATA 186 Defaults Page	11-7
<i>Figure 11-3</i>	Configure Defaults—CableHome WAN-Data Defaults Page	11-10
<i>Figure 11-4</i>	Configure Defaults—CableHome WAN-MAN Defaults Page	11-11
<i>Figure 11-5</i>	Configure Defaults—Computer Defaults Page	11-12
<i>Figure 11-6</i>	Configure Defaults—DOCSIS Defaults Page	11-13
<i>Figure 11-7</i>	Configure Defaults—NR Defaults Page	11-15
<i>Figure 11-8</i>	Configure Defaults—PacketCable Defaults Page	11-17
<i>Figure 11-9</i>	Configure Defaults—RDU Defaults Page	11-19
<i>Figure 11-10</i>	Configure Defaults—System Defaults Page	11-21
<i>Figure 11-11</i>	Configure Defaults—XGCP Defaults Page	11-23
<i>Figure 11-12</i>	View External Files Page	11-26

<i>Figure 11-13</i>	Sample Binary File Content	11-28
<i>Figure 11-14</i>	Sample Jar File Content	11-28
<i>Figure 11-15</i>	Manage License Keys Page	11-31
<i>Figure 11-16</i>	Manage Publishing Page	11-35



T A B L E S

<i>Table 1</i>	Product Documentation	i-xxiii
<i>Table 2</i>	Related Documentation	i-xxiv
<i>Table 2-1</i>	TACACS+ Service Levels	2-6
<i>Table 2-2</i>	BAC-Supported MIBs	2-12
<i>Table 2-3</i>	BAC CLI Commands	2-14
<i>Table 2-4</i>	Severity Levels	2-16
<i>Table 2-5</i>	DHCP Server Extension Trace Levels	2-19
<i>Table 3-1</i>	RDU Workflow Checklist	3-1
<i>Table 3-2</i>	Hardware DPE Configuration Checklist	3-2
<i>Table 3-3</i>	Solaris DPE Configuration Checklist	3-4
<i>Table 3-4</i>	Network Registrar Workflow Checklist	3-5
<i>Table 3-5</i>	DOCSIS Checklist	3-6
<i>Table 3-6</i>	PacketCable Secure Checklist	3-7
<i>Table 3-7</i>	PacketCable Basic Checklist	3-10
<i>Table 3-8</i>	Non-Secure CableHome Provisioning Checklist	3-11
<i>Table 4-1</i>	DOCSIS Workflow Description	4-2
<i>Table 5-1</i>	PacketCable Secure eMTA Provisioning	5-3
<i>Table 5-2</i>	KDC Logging Levels	5-8
<i>Table 5-3</i>	PacketCable Certificates	5-10
<i>Table 5-4</i>	Directory Structure for Multiple Realms	5-11
<i>Table 5-5</i>	PacketCable Basic eMTA Provisioning	5-28
<i>Table 6-1</i>	Troubleshooting Scenarios	6-5
<i>Table 6-2</i>	MTA Root Certificate	6-11
<i>Table 6-3</i>	MTA Manufacturer Certificates	6-12
<i>Table 6-4</i>	MTA Device Certificates	6-13
<i>Table 6-5</i>	CableLabs Service Provider Root Certificates	6-14
<i>Table 6-6</i>	CableLabs Service Provider CA Certificates	6-15
<i>Table 6-7</i>	Local System CA Certificates	6-15
<i>Table 6-8</i>	KDC Certificates	6-16
<i>Table 6-9</i>	DF Certificates	6-17

Table 6-10	PacketCable Server Certificates	6-18
Table 6-11	CableLabs Code Verification Root CA Certificates	6-20
Table 6-12	CableLabs Code Verification CA Certificates	6-20
Table 6-13	Manufacturer Code Verification Certificates	6-21
Table 6-14	Service Provider Code Verification Certificates	6-22
Table 7-1	Non-Secure CableHome Provisioning Workflow	7-2
Table 8-1	Template Grammar	8-2
Table 8-2	Defined Option Encoding Types	8-12
Table 8-3	DOCSIS Options and Version Support	8-15
Table 8-4	PacketCable MTA 1.0 Options	8-25
Table 8-5	Non-Secure CableHome Options and Version Support	8-26
Table 9-1	Browser Platform Support	9-2
Table 9-2	Administrator User Interface Icons	9-6
Table 10-1	Searches Supported for Device Management	10-6
Table 10-2	View Device Details Page	10-11
Table 10-3	View Device Provisioning Engines Details Page	10-20
Table 10-4	View Network Registrar Extension Point Details Page	10-24
Table 10-5	View Provisioning Groups Details Page	10-25
Table 10-6	View Regional Distribution Unit Details Page	10-26
Table 11-1	Manage Class of Service Page	11-2
Table 11-2	Configure Defaults—ATA 186 Defaults Page	11-7
Table 11-3	Configure Defaults—CH WAN-Data/CH WAN-MAN Defaults Page	11-8
Table 11-4	Configure Defaults—DOCSIS Defaults Page	11-14
Table 11-5	Configure Defaults—Network Registrar Defaults Page	11-15
Table 11-6	Configure Defaults—PacketCable Defaults Page	11-17
Table 11-7	Configure Defaults—RDU Defaults Page	11-20
Table 11-8	Configure System Defaults Page	11-21
Table 11-9	Configure XGCP Defaults Page	11-23
Table 11-10	View External Files Page	11-27
Table 11-11	Required RDU Extension Points	11-32
Table 11-12	Modify Publishing Plug-Ins Page	11-36
Table 13-1	BAC Tools	13-1
Table 13-2	Logging Levels	13-2
Table 13-3	Sample Relate/Unrelate Process	13-22
Table A-1	Severity Levels for Alert Messages	A-2

<i>Table A-2</i>	RDU Alerts	A-2
<i>Table A-3</i>	DPE Alerts	A-4
<i>Table A-4</i>	Watchdog Agent Alerts	A-5
<i>Table A-5</i>	Network Registrar Extension Alerts	A-6
<i>Table B-1</i>	DHCP Option 122 to BAC Property Comparison	B-1
<i>Table B-2</i>	DHCP Option 177 to BAC Property Comparison	B-2



Preface

Welcome to the *Cisco Broadband Access Center Administrator Guide, 2.7.1.* This guide describes concepts and configurations related to Cisco Broadband Access Center, referred to as BAC throughout this guide.

The preface provides an outline of other chapters in this guide, details information about related documents that support this BAC release, and demonstrates the styles and conventions used in the guide.



Note

Use this guide along with the documentation listed in [Product Documentation, page xxiii](#), and [Related Documentation, page xxiv](#).

This Preface describes:

- [Audience, page xxi](#)
- [How This Guide Is Organized, page xxii](#)
- [Conventions, page xxiii](#)
- [Product Documentation, page xxiii](#)
- [Related Documentation, page xxiv](#)
- [Obtaining Documentation, Obtaining Support, and Security Guidelines, page xxv](#)

Audience

System administrators use this guide to configure BAC for automating large-scale provisioning for broadband access. The administrator should be familiar with:

- Basic networking concepts and terminology
- Network administration
- Cable networks

How This Guide Is Organized

The major sections of this guide are:

Broadband Access Center Overview	Describes BAC, its features, and benefits.
Broadband Access Center Architecture	Describes the system architecture implemented in this BAC release.
Configuration Workflows and Checklists	Provides checklists to follow when configuring BAC for use.
DOCSIS Configuration	Describes how to bring a BAC DOCSIS deployment into service.
PacketCable Voice Configuration	Describes how to bring a PacketCable voice deployment into service.
Troubleshooting PacketCable eMTA Provisioning	Describes how to troubleshoot the provisioning process for PacketCable embedded Media Terminal Adapters (eMTAs).
CableHome Configuration	Describes how to bring a CableHome deployment, using the non-secure (DHCP) version, into service.
Configuration Templates Management	Describes the configuration templates that BAC supports and how to develop template files.
Understanding the Administrator User Interface	Describes how to access BAC from the administrator user interface.
Using the Administrator User Interface	Describes how to perform administration activities, including searching for and viewing device information, from the administrator user interface.
Configuring Broadband Access Center	Describes how to perform configuration activities from the administrator user interface.
Configuring and Using the Sample User Interface	Describes concepts, uses, and application of the sample workflow user interface.
Support Tools and Advanced Concepts	Describes BAC tools intended to help configure, maintain speed, and improve the installation, deployment, and use of BAC.
Database Management	Describes how to manage and maintain the RDU database.
Alert and Error Messages	Lists and describes BAC alert messages.
PacketCable DHCP Options to BAC Properties Mapping	Identifies the mapping of BAC properties to the PacketCable DHCP options used for PacketCable provisioning.
API Use Cases	Presents a series of the most common provisioning API use cases, including pseudo code segments that can be used to model typical service-provider workflows.
Glossary	Defines terminology used in this guide and generally applicable to the technologies being discussed.

Conventions

This document uses the following conventions:

Item	Convention
Commands and keywords	boldface font
Variables for which you supply values	<i>italic</i> font
Displayed session and system information	screen font
Information you enter	boldface screen font
Variables you enter	<i>italic screen</i> font
Menu items and button names	boldface font
Selecting a menu item in paragraphs	Option > Network Preferences
Selecting a menu item in tables	Option > Network Preferences



Note

Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the publication.



Caution

Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Product Documentation



Note


We sometimes update the printed and electronic documentation after original publication. Therefore, you should also review the documentation on Cisco.com for any updates.

[Table 1](#) describes the documentation that is available for this BAC release.

Table 1 **Product Documentation**

Document Title	Available Format
<i>Release Notes for Cisco Broadband Access Center, Release 2.7.1</i>	<ul style="list-style-type: none"> PDF on the product CD-ROM On Cisco.com: http://cisco.com/en/US/products/sw/netmgsw/ps529/prod_release_notes_list.html
<i>Installation and Setup Guide for Cisco Broadband Access Center, Release 2.7.1.</i>	<ul style="list-style-type: none"> PDF on the product CD-ROM On Cisco.com: http://cisco.com/en/US/products/sw/netmgsw/ps529/prod_installation_guides_list.html

Table 1 *Product Documentation (continued)*

Document Title	Available Format
<i>Cisco Broadband Access Center Administrator Guide, Release 2.7.1</i>	<ul style="list-style-type: none"> PDF on the product CD-ROM On Cisco.com: http://cisco.com/en/US/products/sw/netmgtsw/ps529/prod_maintenance_guides_list.html
<i>Cisco Broadband Access Center DPE CLI Reference, Release 2.7.1</i>	<ul style="list-style-type: none"> PDF on the product CD-ROM On Cisco.com: http://cisco.com/en/US/products/sw/netmgtsw/ps529/prod_command_reference_list.html
<p>To support the DPE-2115:</p> <ul style="list-style-type: none"> <i>DPE-2115 Recovery CD-ROM Release Notes</i> <i>Installation and Setup Guide for the Cisco 1102 VLAN Policy Server</i> <p> Caution Refer to the Installation and Setup guide only for port and connector identification and to perform hardware installation. Do not attempt to perform any of the configuration instructions found in the guide.</p>	<p>On Cisco.com:</p> <ul style="list-style-type: none"> http://cisco.com/en/US/products/sw/netmgtsw/ps529/prod_release_notes_list.html http://cisco.com/en/US/products/sw/secursw/ps2136/prod_installation_guides_list.html

Related Documentation



Note

We sometimes update the printed and electronic documentation after original publication. Therefore, you should also review the documentation on [Cisco.com](http://cisco.com) for any updates.

[Table 2](#) describes the related documentation that is available for this BAC release.

Table 2 *Related Documentation*

Document Title	Available Format
<i>Release Notes for Cisco Network Registrar 6.2.3</i>	<p>On Cisco.com:</p> <p>http://cisco.com/en/US/products/sw/netmgtsw/ps1982/prod_release_notes_list.html</p>
<i>Cisco Network Registrar User's Guide, Release 6.2.1</i>	<p>On Cisco.com:</p> <p>http://cisco.com/en/US/products/sw/netmgtsw/ps1982/products_user_guide_list.html</p>

Table 2 *Related Documentation (continued)*

Document Title	Available Format
<i>Release Notes for Cisco Network Registrar 6.2.3</i>	On Cisco.com: http://cisco.com/en/US/products/sw/netmgtsw/ps1982/prod_release_notes_list.html
<i>Cisco Network Registrar CLI Reference, Release 6.2.1</i>	On Cisco.com: http://cisco.com/en/US/products/sw/netmgtsw/ps1982/prod_command_reference_list.html

Obtaining Documentation, Obtaining Support, and Security Guidelines

For information on obtaining documentation, obtaining support, providing documentation feedback, security guidelines, and also recommended aliases and general Cisco documents, see the monthly *What's New* in Cisco Product Documentation, which also lists all new and revised Cisco technical documentation, at:

<http://www.cisco.com/en/US/docs/general/whatsnew/whatsnew.html>



CHAPTER 1

Broadband Access Center Overview

Cisco Broadband Access Center (BAC) automates the tasks of provisioning and managing customer premises equipment (CPE) in a broadband service-provider network.

With the high-performance capabilities of BAC, you can scale the product to suit networks of virtually any size, even those with millions of CPE. It also offers high availability, made possible by the product's distributed architecture and centralized management.

BAC is designed to handle the rapid growth of service providers. It targets broadband service providers (including multiple service operators), internet, and voice service providers who want to deploy IP data, voice, and video on hybrid fiber and coaxial cable networks.

BAC provides such critical features as redundancy and failover. It can be integrated into new or existing environments through a provisioning application programming interface (API) that lets you control how BAC operates. You can use the provisioning API to register devices in BAC, assign device configurations, and configure the entire BAC provisioning system.

Features and Benefits

BAC lets multiple service operators (MSOs) meet the rapidly changing demands for data over cable services. Using BAC, you can realize these benefits of its architecture:

- Increased scalability
- Distributed architecture
- Redundancy
- Extensibility

Supported Technologies

This BAC release supports these technologies:

- DOCSIS high-speed data
- PacketCable voice services, Secure and Basic
- Non-secure CableHome provisioning

DOCSIS High-Speed Data

The Data Over Cable Service Interface Specification (DOCSIS) defines functionality in cable modems that are involved in high-speed data distribution over cable television system networks. This functionality allows MSOs to provide a variety of services through an “always-on” Internet connection. These services include broadband Internet connectivity, telephony, real-time interactive gaming, and video conferencing.

**Note**

This BAC release supports DOCSIS 1.0, 1.1, and 2.0 devices.

PacketCable Secure Voice Service

PacketCable voice technology enables the delivery of advanced, real-time multimedia services over a two-way cable network. PacketCable is built on top of the infrastructure supported by cable modems to enable a wide range of multimedia services such as IP telephony, multimedia conferencing, interactive gaming, and general multimedia applications.

Using PacketCable voice technology, you can provide additional services, such as basic and extended telephony services, in a broadband network. For this purpose, PacketCable is an efficient and cost-effective option.

**Note**

BAC currently supports versions 1.0, 1.1, and 1.5 of the PacketCable specifications.

Euro-PacketCable services are the European equivalent of the North American PacketCable standard. The only significant difference between the two is that Euro PacketCable uses different MIBs.

PacketCable Basic Voice Service

Non-secure PacketCable voice services are the same as the standard PacketCable voice services except for the lack of security found in the non-secure variant.

CableHome

Non-secure CableHome 1.0 provisioning (hereafter referred to as home networking technology) is built on top of the existing DOCSIS standard and supports a ‘plug and play’ environment for residential broadband connectivity. This form of home networking technology encompasses a DOCSIS home access device with support for CableHome functionality. This device is known as Portal Services and is considered to be the home’s entry point.



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, 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
<code>bprAgent start</code>	Starts the BAC watchdog agent, including all monitored processes.
<code>bprAgent stop</code>	Stops the BAC watchdog agent, including all monitored processes.
<code>bprAgent restart</code>	Restarts the BAC watchdog agent, including all monitored processes.
<code>bprAgent status</code>	Gets the status of the BAC watchdog agent, including all monitored processes.
<code>bprAgent start process-name</code>	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 *rdu.log*.
-

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 *audit.log*.
-

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.



CHAPTER 3

Configuration Workflows and Checklists

This chapter is divided into two major sections that define the processes to follow when configuring BAC components to support various technologies. These sections are:

- [Component Workflows, page 3-1](#)
- [Technology Workflows, page 3-5](#)

Component Workflows

This section describes the workflows you must follow to configure each BAC component for the technologies that BAC supports. You must perform these configuration tasks before configuring BAC to support specific technologies.

In some instances, certain procedures may only be applicable to a lab or component installation. In these cases that appropriate indication is made.

The component workflows described in this section are arranged in a checklist format and include:

- [RDU Checklist, page 3-1](#)
- DPE Checklists, including:
 - [Hardware DPE Checklist, page 3-2](#)
 - [Solaris DPE Checklist, page 3-3](#)
- [Network Registrar Checklist, page 3-5](#)



Note

Tasks marked with an asterisk (*) are mandatory.

RDU Checklist

[Table 3-1](#) identifies the workflow to follow when configuring the RDU.

Table 3-1 RDU Workflow Checklist

Task	Refer to...	Installation Type
1. Configure the system syslog service for use with BAC.	<i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i>	Both

Table 3-1 RDU Workflow Checklist (continued)

Task	Refer to...	Installation Type
2. Access the BAC administrator user interface.	Accessing the Administrator User Interface, page 9-2	Both
3. Change the admin password.	Accessing the Administrator User Interface, page 9-2	Both
4. Add the appropriate license keys.	Managing License Keys, page 11-30	Both
5. Configure the RDU database backup procedure.	Backup and Recovery, page 14-4	Component Only
6. Configure the RDU SNMP agent.	Using the snmpAgentCfgUtil.sh Tool, page 13-15	Component Only

Hardware DPE Checklist

You must perform the activities described in [Table 3-2](#) after those described in [Table 3-1](#).



Note Tasks marked with an asterisk (*) are mandatory.

[Table 3-2](#) identifies the workflow to follow when configuring the hardware DPE.

Table 3-2 Hardware DPE Configuration Checklist

Task	Refer to ...	Installation Type
1. Change the passwords.	The password command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> .	Component Only
2. Configure the system syslog service for use with BAC.	<i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i> .	Both
3. Configure your IP address.*	The interface ethernet ip address command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> .	Component Only
4. Configure the provisioning interface.*	The interface ethernet provisioning enabled command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> .	Component Only
5. Configure the default hardware gateway.*	The ip default-gateway command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> .	Component Only
6. Configure the provisioning FQDN.	The interface ethernet provisioning fqdn command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> .	Component Only

Table 3-2 Hardware DPE Configuration Checklist (continued)

Task	Refer to ...	Installation Type
7. Configure the BAC shared secret.*	The dpe shared-secret command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
8. Configure the DPE to connect to the desired RDU.*	The dpe rdu-server command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
9. Configure the Network Time Protocol (NTP).	The ntp server command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
10. Configure the primary provisioning group.*	The dpe provisioning-group primary command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
11. Configure a hostname.*	The hostname command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
12. Configure a domain name.*	The ip domain-name command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
13. Configure a minimum of one name server.*	The ip name-server command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
14. Configure the required routes to the other BAC components as well as to the devices in the network.	The ip route command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
15. Configure the DPE SNMP agent.	The SNMP agent commands in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
16. Verify that you are connected to RDU.	Viewing Servers, page 10-19	Component Only

Solaris DPE Checklist

You must perform the activities described in [Table 3-3](#) after those described in [Table 3-1](#).



Note

This checklist applies to component installation of the Solaris DPE. A lab installation prompts for the required parameters, and automatically configures the selected technologies. Lab installations also use a single SNMP agent to monitor the DPE and the RDU. You can configure this agent from the DPE CLI or the `snmpAgentCfgUtil.sh` tool. See [Using the snmpAgentCfgUtil.sh Tool, page 13-15](#).



Note Tasks marked with an asterisk (*) are mandatory.

Table 3-3 identifies the workflow to follow when configuring the Solaris DPE.

Table 3-3 Solaris DPE Configuration Checklist

Task	Refer to ...	Installation Type
1. Configure the system syslog service for use with BAC.	<i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1.</i>	Both
2. Change the passwords.	The password command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Both
3. Configure the provisioning interface.*	The interface ethernet provisioning enabled command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
4. Configure the provisioning FQDN.	The interface ethernet provisioning fqdn command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
5. Configure the BAC shared secret.*	The dpe shared-secret command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
6. Configure the DPE to connect to the desired RDU.*	The dpe rdu-server command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
7. Configure the Network Time Protocol (NTP).	Solaris documentation for configuration information.	Component Only
8. Configure the primary provisioning group.*	The dpe provisioning-group primary command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
9. Configure the required routes to the other BAC components as well as to the devices in the network.	The ip route command described in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
10. Configure the DPE SNMP agent.	The SNMP agent commands in the <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1.</i>	Component Only
11. Verify that you are connected to RDU.	Viewing Servers, page 10-19	Both

Network Registrar Checklist

You must perform the activities described in [Table 3-4](#) after those described in [Table 3-2](#) or [Table 3-3](#).



Caution

The BAC DHCP option settings always replace any DHCP option values set within Network Registrar.



Note

Tasks marked with an asterisk (*) are mandatory.

[Table 3-4](#) identifies the workflow to follow when configuring Network Registrar.

Table 3-4 Network Registrar Workflow Checklist

Task	Refer to...	Installation Type
1. Validate the Network Registrar extensions.	<i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i> , for information on configuring valid extensions.	Both
2. Configure the system syslog service for use with BAC.	<i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i> , for information on configuring the system BAC syslog service.	Both
3. Configure client classes/scope-selection tags that match those defined in the RDU.*	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring client classes and scope-selection tags.	Both
4. Configure scopes.*	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring scopes.	Both
5. Configure policies.*	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring policies.	Both
6. Configure the backup procedure for the Network Registrar database.	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on backing up the Network Registrar database.	Component Only
7. Verify that you are connected to the correct RDU.	Viewing Servers, page 10-19	Both

Technology Workflows

This section describes the activities that you must perform when configuring BAC to support specific technologies.

The technology workflows described in this section are arranged in a checklist format and include:

- [DOCSIS Checklist, page 3-6](#)

- PacketCable Checklists including:
 - [PacketCable Secure, page 3-6](#)
 - [PacketCable Basic, page 3-10](#)
- [Non-Secure CableHome Provisioning Checklist, page 3-11](#)

**Note**

Tasks marked with an asterisk (*) are mandatory.

DOCSIS Checklist

You must perform the activities described in [Component Workflows, page 3-1](#), in addition to those described in [Table 3-5](#) to successfully configure BAC for DOCSIS operations.

Table 3-5 **DOCSIS Checklist**

Task	Refer to...
1. Configure the RDU	
a. Configure all provisioned DHCP criteria.	Configuring DHCP Criteria, page 11-24
b. Configure provisioned Class of Service. Add the Class of Service that may be used by any provisioned DOCSIS modem.	Configuring Class of Service, page 11-1
c. Configure the promiscuous mode of operation.	System Defaults, page 11-21
2. Configure Network Registrar	
Configure client classes/scope-selection tags to match those added for the provisioned DOCSIS modem DHCP criteria.	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring client classes and scope-selection tags.

PacketCable Checklists

BAC supports two variations of PacketCable. This section identifies the tasks that must be performed for each, including:

- [PacketCable Secure, page 3-6](#)
- [PacketCable Basic, page 3-10](#)

**Note**

The checklists in this section assume that an appropriate PacketCable configuration file and the correct MIBs are loaded.

PacketCable Secure

BAC supports two variants of PacketCable Secure:

- North American PacketCable
- Euro PacketCable

You must perform the PacketCable-related tasks described in [Table 3-6](#) after those described in [Component Workflows, page 3-1](#).

The Secure PacketCable checklists involve working with every BAC component.

**Note**

For PacketCable-compliant operations, the maximum allowable clock skew between the MTA and KDC is 300 seconds (5 minutes). This value is the default setting.

[Table 3-6](#) identifies the workflow to follow when configuring PacketCable Secure on BAC.

**Note**

Tasks marked with an asterisk (*) are mandatory.

Table 3-6 PacketCable Secure Checklist

Task	PacketCable Variant		Refer to...
	North American	Euro	
1. Configure the RDU			
a. Enable the autogeneration of Media Terminal Adapter (MTA) FQDNs.	✓	✓	Automatic FQDN Generation, page 11-38 , for information on enabling and configuring autogeneration of FQDNs.
b. Configure all provisioned DHCP criteria.	✓	✓	Configuring DHCP Criteria, page 11-24
c. Configure all provisioned Class of Service.	✓	✓	Configuring Class of Service, page 11-1
d. Configure an SNMPv3 cloning key.*	✓	✓	Configuring SNMPv3 Cloning on the RDU and DPE for Secure Communication with PacketCable MTAs, page 11-37
e. Configure the RDU to use Euro PacketCable MIBs.		✓	Configuring Euro PacketCable MIBs, page 5-30
2. Configure the DPE			
a. Configure a KDC service key.*	✓	✓	The packetcable registration kdc-service-key command described in the: <ul style="list-style-type: none"> • <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> • <i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i>

Table 3-6 PacketCable Secure Checklist (continued)

Task	PacketCable Variant		Refer to...
	North American	Euro	
b. Configure a privacy policy.*	✓	✓	The packetcable registration policy-privacy command described in the: <ul style="list-style-type: none"> • <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> • <i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i>
c. Configure an SNMPv3 cloning key.*	✓	✓	The packetcable snmp key-material command described in the: <ul style="list-style-type: none"> • <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> • <i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i> <p>Note On a hardware DPE, you must run this command from the console mode.</p>
d. Enable PacketCable.*	✓	✓	The packetcable enable command described in the: <ul style="list-style-type: none"> • <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> • <i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i>
e. Configure the optional MTA file encryption.	✓	✓	The packetcable registration encryption command described in the: <ul style="list-style-type: none"> • <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> • <i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i>
3. Configure the KDC			
a. Obtain a KDC license from your Cisco representative and copy that file to the <i>BPR_HOME/kdc</i> directory.	✓	✓	KDC Licenses, page 5-9
b. Configure a certificate chain using the PKCert.sh tool. For Euro PacketCable, use the -e option.	✓	✓	Using the PKCert.sh Tool, page 13-5

Table 3-6 PacketCable Secure Checklist (continued)

Task	PacketCable Variant		Refer to...
	North American	Euro	
c. Configure a service key pair for each DPE's provisioning FQDN.	✓	✓	Using the KeyGen Tool, page 13-11
d. Configure service keys for the ticket-granting-ticket (TGT).	✓	✓	Using the KeyGen Tool, page 13-11
e. Configure service keys for the Call Management Server.	✓	✓	Using the KeyGen Tool, page 13-11
f. Configure Network Time Protocol (NTP).	✓	✓	Solaris documentation for information on configuring NTP for Solaris.
4. Configure DHCP			
a. Configure all necessary PacketCable voice technology properties.	✓	✓	Using the KeyGen Tool, page 13-11
b. Configure dynamic DNS for the MTA scopes.	✓	✓	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring dynamic DNS.
c. Configure client classes/scope-selection tags that match those defined in the RDU.*	✓	✓	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring client classes and scope-selection tags.
5. Configure DNS			
a. Configure dynamic DNS for each DHCP server.	✓	✓	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring dynamic DNS.
b. Configure a zone for the KDC realm.	✓	✓	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring zones.
c. Configure an SRV record for the KDC.	✓	✓	Configuring SRV Records in the Network Registrar DNS Server, page 11-36 , and the <i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring SRV records.
d. Configure records for the KDC and DPE provisioning interface names.	✓	✓	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring records.
Note	We recommend that you use the DNS procedure to configure a reverse zone for the DNS server IP address. Some DNS clients, including nslookup, attempt to resolve the DNS server IP address to an FQDN. This attempt may fail to retrieve any records from the DNS unless the reverse zone is present and properly configured.		

PacketCable Basic

You must perform the PacketCable-related tasks described in [Table 3-7](#) after those described in [Component Workflows, page 3-1](#). The PacketCable Basic checklist involves working with almost every BAC component.

[Table 3-6](#) identifies the workflow to follow when configuring PacketCable Basic on BAC.



Note Tasks marked with an asterisk (*) are mandatory.

Table 3-7 PacketCable Basic Checklist

Task	Refer to...
1. Configure the DPE	
Enable PacketCable.*	The packetcable enable command described in the: <ul style="list-style-type: none"> • <i>Cisco Broadband Access Center DPE CLI Reference, 2.7.1</i> • <i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1.</i>
2. Configure DHCP	
a. Configure dynamic DNS for the MTA scopes.	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring dynamic DNS.
b. Configure client classes/scope-selection tags that match those defined in the RDU.*	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring client classes and scope-selection tags.
3. Configure DNS	
Configure dynamic DNS for each DHCP server.	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring dynamic DNS.
Note	We recommend that you use the DNS procedure to configure a reverse zone for the DNS server IP address. Some DNS clients, including nslookup, attempt to resolve the DNS server IP address to an FQDN. This attempt may fail to retrieve any records from the DNS unless the reverse zone is present and properly configured.
4. Configure a Class of Service, which must contain the following properties:	
Note	You can configure these properties anywhere on the device property hierarchy.

Table 3-7 PacketCable Basic Checklist (continued)

Task	Refer to...
<p>a. /pktcbl/prov/flow/mode</p> <p>This property commands the specific flow that an MTA uses. Set this property to either:</p> <ul style="list-style-type: none"> – BASIC.1—Executes the BASIC.1 flow. – BASIC.2—Executes the BASIC.2 flow. 	Configuring Class of Service, page 11-1
<p>b. /cos/packetCableMTA/file:</p> <p>This property contains the name of the configuration file that is to be presented to the MTA. The configuration file is stored as an external file in BAC.</p> <p>The configuration file presented to a Basic MTA must contain the Basic integrity hash. If you are using a dynamic configuration template, the hash is inserted transparently during template processing. You can use the dynamic template for provisioning in both Secure and Basic modes.</p> <p>However, if the file is a Secure static configuration file, you must convert this file to a Basic static configuration file because Secure and Basic static configuration files are not interoperable. For details on how to perform this conversion, see Activating PacketCable Basic Flow, page 8-39.</p>	Configuring Class of Service, page 11-1

Non-Secure CableHome Provisioning Checklist

You must perform the tasks described in [Component Workflows, page 3-1](#), in addition to those described in [Table 3-8](#) to successfully configure BAC for non-secure CableHome provisioning.

Table 3-8 Non-Secure CableHome Provisioning Checklist

Task	Refer to...
1. Configure the RDU	
<p>a. Configure provisioned DHCP criteria.</p> <p>Add all the DHCP criteria that will be used by the non-secure CableHome devices that you will provision.</p>	Configuring DHCP Criteria, page 11-24
<p>b. Configure provisioned Class of Service.</p> <p>Add the Class of Service that may be used by any provisioned non-secure CableHome device.</p>	Configuring Class of Service, page 11-1
c. Configure the promiscuous mode of operation.	System Defaults, page 11-21

Table 3-8 Non-Secure CableHome Provisioning Checklist (continued)

Task	Refer to...
2. Configure Network Registrar Configure the client classes/scope-selection tags to match those added for the provisioned non-secure CableHome DHCP criteria.	<i>Cisco Network Registrar User's Guide, 6.2.1</i> , for information on configuring client classes and scope-selection tags.



CHAPTER 4

DOCSIS Configuration

This chapter describes the provisioning flow in a Broadband Access Center (BAC) DOCSIS deployment. It also provides information required before configuration and describes the available tools.

- [DOCSIS Workflow, page 4-1](#)
- [Using MIBs with Dynamic DOCSIS Templates, page 4-3](#)
- [BAC Features for DOCSIS Configurations, page 4-4](#)
- [Troubleshooting DOCSIS Networks, page 4-6](#)

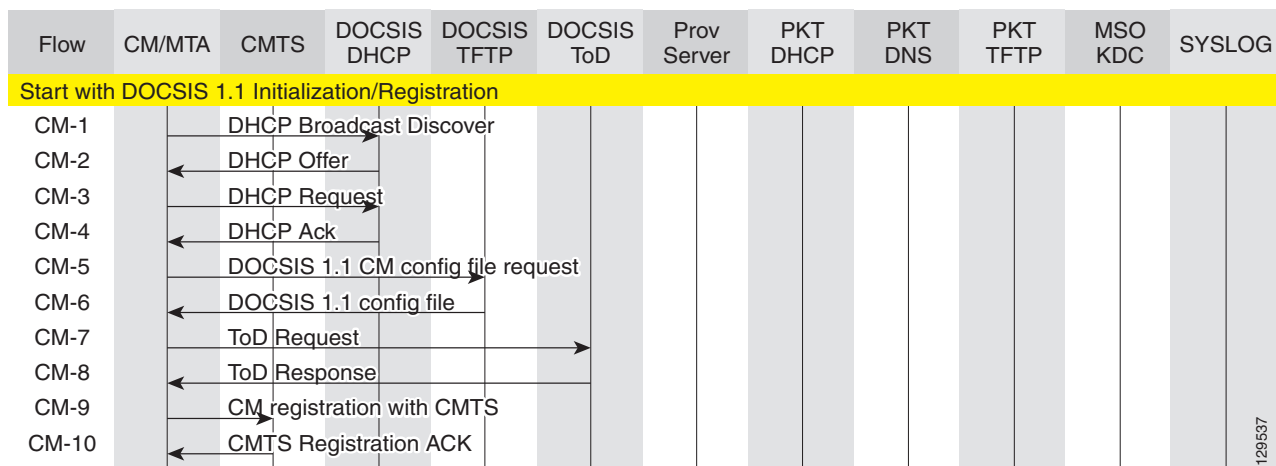


Note See [DOCSIS Option Support, page 8-15](#), for information on DOCSIS options supported by this BAC release.

DOCSIS Workflow

[Figure 4-1](#) shows the provisioning workflow contained in the DOCSIS Provisioning Specification. Each step is described subsequently.

Figure 4-1 DOCSIS Provisioning Flow



129537

Table 4-1 describes the potential problems that can exist at various DOCSIS provisioning steps illustrated in Figure 4-1.

Table 4-1 DOCSIS Workflow Description

Step	DOCSIS Workflow	Potential Problems
CM ¹ -1	DHCP Discover	<ul style="list-style-type: none"> • The init(d) state • No addresses available • Incorrect BAC shared secret • Incorrectly configured Class of Service • DOCSIS template parsing errors (invalid option, include file - not found, and so on) <p>Network Registrar DHCP</p> <ul style="list-style-type: none"> • Incorrect DHCP configuration • DHCP server not there in provisioning group <p>BAC Network Registrar Extension</p> <ul style="list-style-type: none"> • Network Registrar extension cannot contact DPEs • Network Registrar extension fails to find any DPEs in provisioning group • Verify extensions are connected to the RDU • Network Registrar extension gets DPE cache miss, sends request to RDU <p>RDU</p> <ul style="list-style-type: none"> • No appropriate scopes defined (or do not match BAC RDU configuration) • Incorrect RDU IP address • Incorrect RDU port (default 49187) • RDU cannot be pinged from DPE • Configuration generation is failing at the RDU • RDU licenses exceeded, not configured • Device detection is failing at the RDU <p>DPE</p> <ul style="list-style-type: none"> • DPEs not assigned to provisioning group • DPEs cannot be pinged from the DHCP server • DPE interfaces not enabled for provisioning

Table 4-1 DOCSIS Workflow Description (continued)

Step	DOCSIS Workflow	Potential Problems
CM-2	DHCP Offer	Routing issues between DHCP and CMTS ²
CM-3	DHCP Request	init(i) state DHCP server did not provide all the parameters required
CM-4	DHCP Ack	
CM-5	TFTP Request	<ul style="list-style-type: none"> • Init(o) state • Routing issues between CMTS and DPE • No route from TFTP server (DPE) to modem • DPE cache miss (static file, and RDU down or does not have file) • File not found at TFTP server (DPE) • DPE cache miss (dynamic file) • DPE IP validation failure (for example, the IP address of the device is not what was expected, the Dynamic Shared Secret is enabled on CMTS, or a hacker is spoofing as a DOCSIS modem)
CM-6	TFTP Response	Routing issues between DPE and CMTS
CM-7	ToD Request	init(t) state - No route from time server (DPE) to modem
CM-8	ToD Response	
CM-9	CM registration with CMTS	<ul style="list-style-type: none"> • reject(m) - * CMTS shared secret mismatch with BAC or DPE DOCSIS shared secret • reject(c) - * delivered incorrect DOCSIS configuration file (1.1 file to 1.0 CM)
CM-10	CMTS registration Ack	Acceptable states are: <ul style="list-style-type: none"> • online • online(d) • online(pk) • online(pt)

1. CM = cable modem
2. CMTS = cable modem termination system

Using MIBs with Dynamic DOCSIS Templates

For a full list of MIBs that BAC ships with, see [SNMP VarBind, page 8-5](#).

Two versions of the DOCSIS MIB are loaded into the RDU:

- DOCS-CABLE-DEVICE-MIB-OBSOLETE (experimental branch)
- DOCS-CABLE-DEVICE-MIB (mib2 branch)

For information on how to use them, see [DOCSIS MIBs, page 8-5](#).

You can add MIBs using an API call or by modifying *rdu.properties*. For more details, see [Configuring Euro PacketCable MIBs, page 5-30](#).

You can add SNMP TLVs to a template:

- When no MIB is available. See [Adding SNMP TLVs Without a MIB, page 8-8](#).
- With vendor-specific MIBs. See [Adding SNMP TLVs With Vendor-Specific MIBs, page 8-9](#).

BAC Features for DOCSIS Configurations

This section describes BAC value-added features as they relate to the DOCSIS technology.

Dynamic Configuration TLVs

The DPE adds the following TLVs when it receives a TFTP request for dynamic DOCSIS configuration:

- TLV 19: TFTP Server Timestamp (optional)—Displays in the Configure DOCSIS Defaults page as the TFTP Time Stamp Option. See [Table 11-4](#) for more information. This TLV requires NTP synchronization on CMTS and DPE.
- TLV 20: TFTP Server Provisioned Modem Address (optional)—Displays in the Configure DOCSIS Defaults page as the TFTP Modem Address Option. See [Table 11-4](#) for more information.



Note This feature is incompatible with the Cisco CMTS DSS feature. If DSS is set on the Cisco CMTS, you must ensure that the TFTP Server Provisioned Modem Address is disabled.

- TLV 6: CM MIC Configuration Setting (required)
- TLV 7: CMTS MIC Configuration Setting (required)—Displays in the Configure DOCSIS Defaults page as the CMTS Shared Secret. See [Table 11-4](#) for more information.



Note

When configuring CMTS MIC, note the following CMTS IOS release dependencies:

- DOCSIS 2.0 CMTS MIC requires CMTS IOS 12.3BC when including TLV 39 or TLV 40.
- Certain CMTS IOS commands are assumed to be configured by BAC:

- **ip dhcp relay information option**
- **no ip dhcp relay information check**
- **cable helper-address x.x.x.x**

where *x.x.x.x* is the IP address of the Network Registrar DHCP server.

- **cable dhcp-giaddr primary**
-

DPE TFTP IP Validation

For dynamic configuration files, the DPE TFTP server verifies if the IP address of the TFTP client matches the expected DOCSIS cable modem IP address. If it does not match, the request is dropped. This feature is incompatible with the Cisco CMTS DMIC feature.

Use the **no tftp verify-ip** command to disable the verification of requestor IP addresses on dynamic configuration TFTP requests. For detailed information, refer to the *Cisco Broadband Access Center DPE CLI Reference*, 2.7.1.

DOCSIS 1.0, 1.1, 2.0 Support

BAC 2.7.1 supports DOCSIS 1.0, 1.1, and 2.0. See [Support Tools and Advanced Concepts](#), page 13-1, for information describing the TLVs and options that this BAC release supports in each DOCSIS version.

Dynamic DOCSIS Version Selection

BAC can detect a cable modem's DOCSIS version from an incoming DHCP request. It can also detect the CMTS DOCSIS version from a customer-supplied source that provides a mapping of GIADDR to the CMTS DOCSIS version.

Using this information, BAC, if so configured, determines the optimum DOCSIS configuration file for the device. This is the highest common DOCSIS version between the device and the CMTS. For example, if the device supports DOCSIS 2.0 and the CMTS supports DOCSIS 1.1, the DOCSIS 1.1 file is used.



Note

You can specify the DOCSIS version by using the Configuration File Utility. For more information, see [Using the Configuration File Utility](#), page 8-27. This function that the file utility performs is different from RDU verification, during which the RDU DOCSIS Version Selector feature determines the latest DOCSIS version supported by a CMTS.

DOCSIS Configuration File Based on DOCSIS Version

The following Class of Service properties are supported by the BAC administrator user interface and the API:

```
/cos/docsis/file/1.0  
/cos/docsis/file/1.1  
/cos/docsis/file/2.0
```

Optionally, you can add these properties to a DOCSIS Class of Service to associate a DOCSIS configuration filename with a particular DOCSIS version. Each of these properties (if set) causes the RDU to establish a database relationship between the Class of Service and the file named by the property value, as is done for the existing DOCSIS configuration filename property.

GIADDR-Based Dynamic DOCSIS Version Selection

During configuration generation, the service-level selection extension for DOCSIS modems looks for the `/docsis/cmts/version/giaddrToVersionMap` property. The value of this property is the name of an external file containing a mapping of the GIADDR to the DOCSIS version supported by the cable modem. This mapping file must be named `giaddr-docsis-map.txt`.

**Note**

You can add the `giaddr-docsis-map.txt` to the RDU:

- From the API via the `Configuration.addExternalFile()` call.
 - From the administrator user interface via **Configuration > External Files**. See [Adding External Files, page 11-27](#).
-

The `giaddr-docsis-map.txt` file must include the necessary information in the following format:

IPv4_address DOCSIS_version

- *IPv4_address*—IPv4 address of the CMTS interface
- *docsis_version*—DOCSIS version that the cable modem supports

For example, if the CMTS interface had IP address 10.30.0.1 with DOCSIS version 1.0, the file would include:

```
10.30.0.1 1.0
```

**Note**

If the device `GIADDR` is not found in the mapping file, the extension uses the value of the `/docsis/cmts/version/default` property for the DOCSIS version of the cable modem. The default value of this property is 1.0.

To dynamically update the `giaddr-docsis-map.txt` file, edit it and replace it in the RDU via the `replaceExternalFile` API or via the administrator user interface.

**Note**

If the properties for the DOCSIS version selection are not specified on the Class of Service, the original file is used, allowing for systematic upgrades across the network.

Troubleshooting DOCSIS Networks

For information on troubleshooting the DOCSIS technology with respect to BAC and the Cisco uBR7246 CMTS, refer to *Troubleshooting uBR Cable Modems Not Coming Online* at: http://www.cisco.com/en/US/tech/tk86/tk89/technologies_tech_note09186a0080094eb1.shtml



CHAPTER 5

PacketCable Voice Configuration

This chapter describes the tasks you must perform to bring a PacketCable voice deployment into service.

This chapter contains information on these variants of PacketCable:

- [PacketCable Secure eMTA Provisioning, page 5-1](#)
- [PacketCable Basic eMTA Provisioning, page 5-28](#)
- [Euro PacketCable, page 5-29](#)

For information that will help you solve issues that might arise and hinder PacketCable voice technology deployment, see [Troubleshooting PacketCable eMTA Provisioning, page 6-1](#).

This chapter assumes that you are familiar with the contents of the PacketCable Media Terminal Adapter (MTA) Device Provisioning Specification, PKT-SP-PROV1.5-I02-050128. For details, see the PacketCable website.

PacketCable Secure eMTA Provisioning

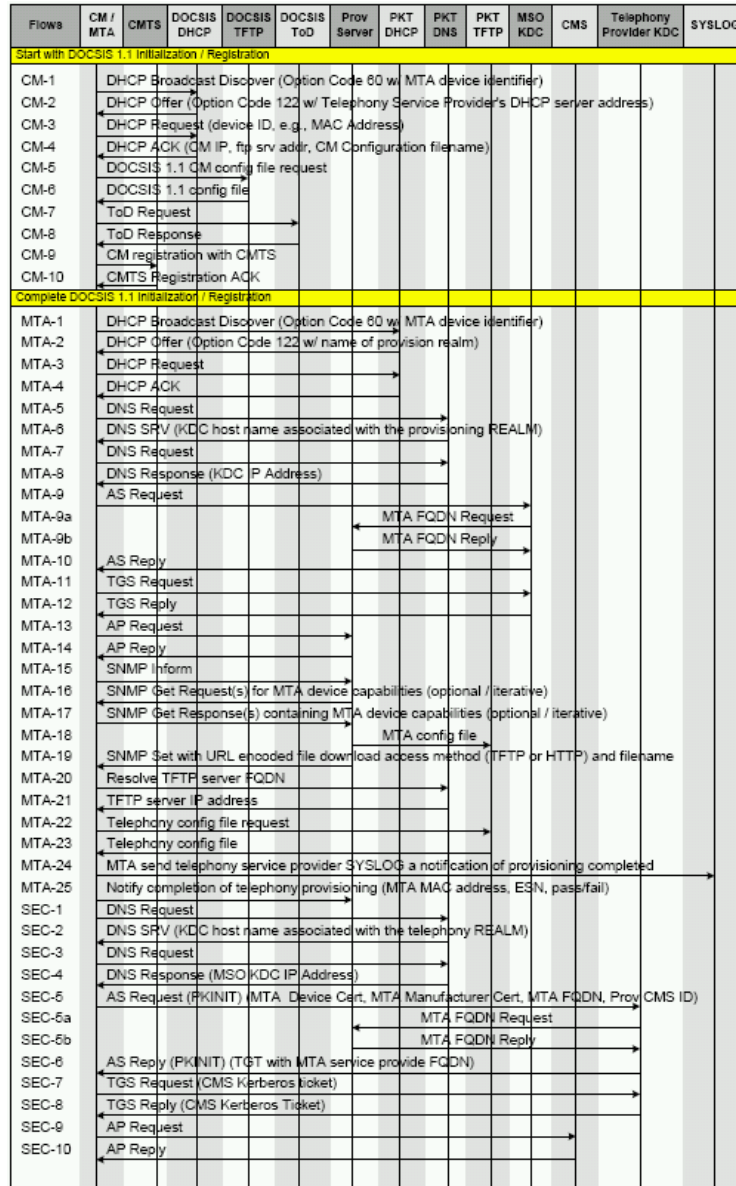
This section deals exclusively with Secure PacketCable voice provisioning. PacketCable Secure is designed to minimize the possibility of theft of telephony service, malicious disruption of service, and so on. PacketCable Secure depends on the Kerberos infrastructure to mutually authenticate the MTA and the provisioning system; in BAC, the Key Distribution Center (KDC) functions as the Kerberos server. SNMPv3 is also used to secure the conversation between the MTA and the provisioning system.

BAC PacketCable Secure Provisioning Flow

All PacketCable provisioning flows are defined as a sequence of steps.

[Figure 5-1](#) illustrates the Secure provisioning flow for PacketCable eMTAs.

Figure 5-1 Embedded-MTA Secure Power-On Provisioning Flow



211035

**Note**

It is strongly recommended that you use a protocol analyzer (protocol sniffer) with the ability to capture data packets to understand exactly which step is failing.

In addition, the content of the KDC log file is critical to understanding the root cause of any KDC failure.

When diagnosing problems in provisioning an embedded Media Terminal Adapters MTA (eMTA), the flow description in [Table 5-1](#) helps identify which step in the PacketCable provisioning flow is failing.

Table 5-1 PacketCable Secure eMTA Provisioning

Step	Workflow	Description
CM-1	DHCP Broadcast Discover	This is the usual DOCSIS cable modem (CM) boot flow with DHCP Option 122 added to provide the MTA with a list of PacketCable DHCP servers from which the MTA is allowed to accept DHCP offers.
CM-2	DHCP Offer	
CM-3	DHCP Request	
CM-4	DHCP Ack	
CM-5	DOCSIS 1.1 CM Config File Request	
CM-6	DOCSIS 1.1 Config File	
CM-7	ToD Request	
CM-8	ToD Response	
CM-9	CM Registration with CMTS	
CM-10	CMTS Registration Ack	

Table 5-1 PacketCable Secure eMTA Provisioning (continued)

Step	Workflow	Description
MTA-1	DHCP Broadcast Discover	<p>Using DHCP, the MTA announces itself as a PacketCable MTA and provides information on the capabilities and provisioning flows it supports (Secure, Basic, and so on.). The MTA also obtains addressing information and DHCP Option 122. DHCP Option 122 contains the PacketCable provisioning server address and the security realm name. This information is used to allow the MTA to contact the KDC and provisioning server.</p> <p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Check the DHCP relay agent on the CMTS for the correct configuration; ensure that your cable modem termination system (CMTS) points to the correct DHCP server. • Verify that you have the correct routing between the MTA, CMTS, DHCP server, and the DPE. • Verify that secondary subnets are configured correctly on the CMTS. • Check the Network Registrar DHCP configuration. Verify if the scopes are configured, if IP addresses are available, and if all secondary subnets are configured. • Check the BAC configuration. Check the <code>cnr_ep.properties</code> file and ensure that the required PacketCable Network Registrar extension properties are configured. For more information, see the PacketCable DHCP Options to BAC Properties Mapping, page B-1. <p>If a packet trace reveals that the MTA is cycling between steps MTA-1 and MTA-2, there could be a problem with the configuration of DHCP Option 122 (realm name or provisioning server FQDN suboptions), DHCP Option 12 (hostname), or DHCP Option 15 (domain name).</p>
MTA-2	DHCP Offer	
MTA-3	DHCP Request	
MTA-4	DHCP Ack	
MTA-5	DNS Request	<p>MTA uses the security realm name (delivered within DHCP Option 122) to perform a DNS SRV lookup on the KDC service and then resolve the KDC IP address.</p>
MTA-6	DNS Srv	
MTA-7	DNS Request	<p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Use a packet sniffer to watch for misdirected or malformed DNS packets sent to the Network Registrar DNS. • Set the Network Registrar DNS log level to detailed packet tracing and verify what arrives there. • Check the DNS configuration—The DNS server specified in <code>cnr_ep.properties</code> must contain the realm zone, the SRV record, and the DNS 'A' record for the KDC.
MTA-8	DNS Response	

Table 5-1 PacketCable Secure eMTA Provisioning (continued)

Step	Workflow	Description
MTA-9	AS Request	<p>The AS-REQ request message is used by the KDC to authenticate the MTA.</p> <p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Check the KDC log file to determine if the AS-REQ arrives and to observe any errors or warnings. • Check that the KDC is configured with the correct MTA_Root certificate. The Manufacturer and Device certificates sent by the MTA within the AS-REQ message must chain with the MTA_Root certificate installed at the KDC.
MTA-9 a	MTA FQDN Request	<p>The KDC extracts the MTA MAC address from the MTA certificate and sends it to the provisioning server for validation. If the provisioning server has the FQDN for that MAC address, it is returned to the KDC. The KDC then compares the FQDN received from the MTA to the FQDN received in the FQDN-REP reply message.</p> <p>Some key troubleshooting hints are:</p> <ul style="list-style-type: none"> • Use a packet sniffer to watch for misdirected or malformed DNS packets. The MTA passes the provisioning server FQDN (which the MTA received in DHCP Option 122) within the AS-REP message to the KDC. The KDC then uses this FQDN to resolve the IP address of the provisioning server. • Check the filenames and content of the KDC key file; the KDC service key in the DPE must match the service key at the KDC. The names of the service key files at the KDC are critical.
MTA-9 b	MTA FQDN Reply	
MTA-1 0	AS Reply (AS-REP)	<p>The KDC grants a provisioning service ticket to the MTA and also sends the Service Provider, Local System Provider (optional), and KDC certificate to the MTA. The MTA then verifies if the certificates sent by the KDC chain to the Service Provider Root certificate stored in the MTA. If these certificates do not chain, the MTA loops back to step MTA-1 of the provisioning flow. See Using the PKCert.sh Tool, page 13-5, for additional information on the KDC.cer file.</p> <p>A key troubleshooting hint: Verify if the KDC log files show that the AS-REP message was sent to the device. If a packet trace reveals the MTA is cycling between steps MTA-1 and MTA-10, there is a problem with the service provider certificate chain.</p>
MTA-1 1	TGS Request	The MTA receives either a service ticket or a ticket-granting-ticket (TGT) following step MTA-10. If the MTA had obtained a TGT instead of a service ticket in step MTA-10, it contacts the ticket-granting-server (KDC) to obtain a service ticket.
MTA-1 2	TGS Reply	The KDC sends a service ticket in the TGS Reply to the MTA.
MTA-1 3	AP Request (AP-REQ)	The MTA presents the ticket (received at step MTA-10) to the provisioning server specified by DHCP Option 122.

Table 5-1 PacketCable Secure eMTA Provisioning (continued)

Step	Workflow	Description
MTA-1 4	AP Reply (AP-REP)	The provisioning server uses the KDC shared secret to decrypt the AP-REQ, validates the provisioning server ticket presented by the MTA, and sends AP-REP with SNMPv3 keys. Subsequent SNMPv3 is now authenticated and (optionally) encrypted.
MTA-1 5	SNMP Inform	The MTA signals to the provisioning server that it is ready to receive provisioning information.
MTA-1 6	SNMP Get Request(s)	SNMPv3—If the provisioning server (DPE) requires additional device capabilities, it sends the MTA one or more SNMPv3 Get requests to obtain the required information on MTA capability. The provisioning server (DPE) may use a GetBulk request to request a bulk of information in a single message.
MTA-1 7	SNMP Get Response(s)	SNMPv3—The MTA sends to the provisioning server (DPE) a response for each GetRequest that contains information on MTA capabilities requested in step MTA-16.
MTA-1 8	MTA Config file	Using information made available in steps MTA-16 and MTA-17, the provisioning server (DPE) determines the contents of the MTA configuration data file.
MTA-1 9	SNMP Set	SNMPv3—The provisioning server performs an SNMPv3 Set to the MTA containing the URL for the MTA configuration file, encryption key for the file, and the file hash value.
MTA-2 0	Resolve TFTP Server FQDN	DNS Request—If the URL-encoded access method contains an FQDN instead of an IPv4 address, the MTA uses the DNS server of the service provider network to resolve the FQDN into an IPv4 address of the TFTP server or the HTTP server.
MTA-2 1	TFTP Server IP Address	DNS Response—The DNS server returns the IPv4 IP address of the service provider network as requested in step MTA-20.
MTA-2 2	Telephony Config File Request	The MTA proceeds to download the VoIP configuration file from the specified TFTP server. Note that BAC integrates the TFTP server into the DPE component.
MTA-2 3	Telephony Config File	
MTA-2 4	MTA Send	The MTA optionally sends a syslog notification to the service provider that provisioning is complete.
MTA-2 5	Notify completion of telephony provisioning	The MTA signals to the provisioning server if the new configuration is acceptable.
SEC-1 to SEC-1 0	These steps are the post-MTA provisioning security flow and are not applicable to BAC provisioning. This flow involves getting Kerberos tickets associated with each CMS with which the MTA communicates. For details, refer to the PacketCable Security Specifications.	

KDC in Provisioning PacketCable Secure eMTAs

PacketCable Secure depends on the Kerberos infrastructure to mutually authenticate the MTA and the provisioning system; in BAC, the KDC functions as the Kerberos server. For an overview of the KDC component, see [Key Distribution Center, page 2-11](#).

For important information related to the KDC, see:

- [Default KDC Properties, page 5-7](#)
- [KDC Certificates, page 5-9](#)
- [KDC Licenses, page 5-9](#)
- [Multiple Realm Support, page 5-10](#)

Default KDC Properties

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.

**Note**

Be careful in editing the `kdc.ini` file if operational requirements dictate. Incorrect values can render the KDC inoperative. If you do make changes, restart the KDC.

The default properties are:

- **interface address**—Specifies 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
```

**Note**

You must enter the interface address and FQDN values through the KDC Realm Name screen during installation. For specific information, refer to the *Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1*.

- **maximum log file size**—Specifies the maximum size, in kilobytes, that the log file that is generated by the KDC can reach. The KDC creates a new log file only when the current file reaches this maximum size.

For example:

```
maximum log file size = 1000
```

- **n saved log files**—Defines the number of old log files that the KDC saves. The default value is 7. You can specify as many as required.

For example:

```
n saved log files = 10
```

- log debug level—Specifies the logging level for the log file.

```
log debug level = 5
```

Table 5-2 describes the available logging levels for the KDC log file.

Table 5-2 KDC Logging Levels

Log Level	Description
0	Error conditions exist. Sets the logging function to save all error messages and those of a more severe nature.
1	Warning conditions exist. Sets the logging function to save all warning messages and those of a more severe nature.
2	Informational messages. Sets the logging function to save all logging messages available.
{3-7}	Debugging messages. Sets the logging function to save all debugging messages at various levels, from level 3 to level 7.

- minimum (maximum) ps backoff—Specifies the minimum (or maximum) time, in tenths of a second, that the KDC waits for BAC to respond to the FQDN-Request.

For example:

```
minimum ps backoff = 150
```

Using the sample values shown above, a sample INI file might contain data similar to that shown in Example 5-1.

Example 5-1 Sample kdc.ini Configuration File

```
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 setting 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 greater control over the authorization of telephony endpoints, it also causes heavier message loads on the KDC and increased network traffic. In most situations, 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**, you can change the units to minutes or days, respectively.

The default value is 168, or seven days. We recommend that you not change this value because this value is the length of time required to conform to the PacketCable security specification.

For example:

```
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 **m** or **d**, you can change the units to minutes or days, respectively.

The default value is 144, or six days. We recommend that you not change this value.

For example:

```
minimum ticket duration = 144
```

KDC Certificates

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 the content in the certificates installed in the MTA.



Note

Certificates are required for the KDC to function.

You can use the PKCert tool to install, and manage, the certificates that the KDC requires for its operation. The PKCert tool installs the CableLabs service provider certificates as certificate files. For information on running this tool, see [Using the PKCert.sh Tool, page 13-5](#).

The PKCert tool is available only if you have installed the KDC component.

KDC Licenses

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

To install a KDC license file:

-
- Step 1** Obtain your license file from your Cisco representative.
 - Step 2** Log in to the BAC host as **root**.
 - Step 3** Change to the *BAC_home/kdc* directory.

Step 4 Copy the license file to this *BAC_home/kdc* directory.



Caution

Be careful not to copy this as an ASCII file. The file contains binary data susceptible to unwanted modification during an ASCII transfer.

Do not copy KDC license files between operating systems because the transfer process may damage the file.

Step 5 To restart the KDC server and make the changes take effect, run the **bprAgent restart kdc** command from the */etc/init.d* directory.

Multiple Realm Support

The BAC KDC supports the management of multiple realms, for which a complete set of valid PacketCable X.509 certificates and a KDC private key must be present. These certificates must reside in the *BAC_home/kdc/solaris/packetcable/certificates* directory.

BAC supports additional realms by installing subdirectories under the *BAC_home/kdc/solaris/packetcable/certificates* directory; each subdirectory is named after a specific realm.

[Table 5-3](#) lists the different certificates, with their corresponding filenames, that must be available in the *BAC_home/kdc/solaris/packetcable/certificates* directory.

Table 5-3 PacketCable Certificates

Certificate	Certificate Filename
MTA Root	MTA_Root.cer
Service Provider Root	CableLabs_Service_Provider_Root.cer
Service Provider CA	Service_Provider.cer
Local System Operator CA	Local_System.cer
KDC	KDC.cer

The primary realm is set up during installation of the KDC component. For the primary realm, the KDC certificate (KDC.cer) resides in the *BAC_home/kdc/solaris/packetcable/certificates* directory. Its private key (KDC_private_key.pkcs8) resides in the *BAC_home/kdc/solaris/* directory.

To configure additional realms, follow this procedure, which is described in detail subsequently.

Step 1 Locate the directory containing your KDC certificates.

Step 2 Create a subdirectory under the directory which stores the KDC certificates.



Note

Match the name of the subdirectory with the name of the specific realm. Use only uppercase characters while naming the subdirectory.

- Step 3** Place the KDC certificate and the private key for the realm in the subdirectory you created.
- 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.



Note Since all realms must be rooted in the same certificate chain, a KDC installation supports only one locale (North American PacketCable or Euro PacketCable) at any given point.

Table 5-4 describes the directory structure and files for a primary realm (for example, IPFONIX.COM) with two secondary realms (for example, IPFONIX2.COM and IPFONIX3.COM). The structure assumes that the higher-level certificates are similar for the primary realm and its secondary realms.

Table 5-4 Directory Structure for Multiple Realms

Directory	File Content in Directory
<i>BAC_home/kdc/solaris</i>	For primary realm IPFONIX.COM: KDC private key
<i>BAC_home/kdc/solaris/packetcable/certificates</i>	For primary realm IPFONIX.COM: <ul style="list-style-type: none"> • MTA_Root.cer • CableLabs_Service_Provider_Root.cer • Service_Provider.cer • Local_System.cer • KDC.cer Directory /IPFONIX2.COM Directory /IPFONIX3.COM
<i>BAC_home/kdc/solaris/packetcable/certificates/IPFONIX2.COM</i>	For secondary realm IPFONIX2.COM: <ul style="list-style-type: none"> • KDC.cer • KDC private key
<i>BAC_home/kdc/solaris/packetcable/certificates/IPFONIX3.COM</i>	For secondary realm IPFONIX3.COM: <ul style="list-style-type: none"> • KDC.cer • KDC private key

Configuring the KDC for Multiple Realms

This section describes the workflow to configure the KDC for multiple realms. Before proceeding, complete the installation of the RDU, the DPE, and the Network Registrar extensions. For installation instructions, refer to the *Installation and Setup Guide for the Cisco Broadband Access Center, 2.7.1*.

The following workflow uses sample realms and directories to describe how to configure the KDC for multiple realms. The primary realm used here is IPFONIX.COM and its secondary realms are IPFONIX2.COM and IPFONIX3.COM.

The setup featured in the following workflow provisions three MTAs: a Motorola SBV 5120 MTA, a Linksys CM2P2 MTA, and an SA WebStar DPX 2203 MTA. Each MTA is to be provisioned in one realm: the Motorola in the IPFONIX.COM realm, the Linksys MTA in the IPFONIX2.COM realm, and the SA MTA in the IPFONIX3.COM realm.

**Note**

The sample output shown in the following procedure has been trimmed for demonstration purposes.

To configure the KDC for multiple realms:

Step 1 Verify the following configuration settings on the DPE:

- a. Ensure that PacketCable services are enabled, by using the **show run** command.

To enable the PacketCable service, use the **packetcable enable** command.

For example:

```
dpe# show run
aaa authentication local
debug dpe events
dpe port 49186
dpe provisioning-group primary default
packetcable enable
snmp-server location equipmenttrack5D
snmp-server udp-port 8001
tacacs-server retries 2
tacacs-server timeout 5
tftp verify-ip
```

For details on the commands, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

- b. Ensure that the security used for communication between the KDC and a DPE is set, by using the **show run** command.

To generate and set the security key, use the **packetcable registration kdc-service-key** command.

For example:

```
dpe# show run
aaa authentication local
debug dpe events
dpe port 49186
packetcable enable
packetcable registration kdc-service-key <value is set>
snmp-server contact AceDuffy-ext1234
tftp verify-ip
```

For details on the commands, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

- c. Ensure that the security key that permits secure communication between the DPE and the RDU for PacketCable SNMPv3 cloning is set. Again, use the **show run** command.

To generate and set the security key, use the **packetcable snmp key-material** command.

For example:

```
dpe# show run
aaa authentication local
debug dpe events
dpe port 49186
packetcable enable
```

```
packetcable registration kdc-service-key <value is set>
packetcable snmp key-material <value is set>
```

For details on the commands, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.



Note When you configure PacketCable settings on the DPE, ensure that you run the **dpe reload** command so that the changes take effect.

- Step 2** In the configuration file for Network Registrar extension points (cnr_ep.properties), verify if the **/ccc/kerb/realm** parameter is set to the primary realm; in this case, IPFONIX.COM. To do this, run the **more cnr_ep.properties** command from the *BAC_home/cnr_ep/conf* directory.

For example:

```
/opt/CSCObpr/cnr_ep/conf# more cnr_ep.properties
#DO NOT MODIFY THIS FILE.
#This file was created on Wed, March 4 06:34:34 EDT 2007
/rdu/port=49187
/rdu/fqdn=dpe4.cisco.com
/cache/provGroupList=Default
/cnr/sharedSecret=fggTaLg0XwKR5
/pktcbl/enable=enabled
/ccc/tgt=01
/ccc/kerb/realm=IPFONIX.COM
/ccc/dhcp/primary=10.10.0.1
/ccc/dns/primary=10.10.0.1
```

- Step 3** Enable static routes appropriately to ensure BAC connectivity with devices behind the CMTS.
- Step 4** Create DNS realm zones for the DNS server that is listed in the cnr_ep.properties file. You can add zones using the Network Registrar administrator user interface via the **DNS > Forward Zones > List/Add Zones** pages.



Note Ensure that the zones you add contain the SRV record and the DNS 'A' record for the KDC server, and that the SRV record for each zone (in this example, IPFONIX.COM, IPFONIX2.COM, and IPFONIX3.COM) point to one KDC.

For information on configuring zones from the administrator user interface, refer to the *Cisco Network Registrar User's Guide, 6.2.1*.

- Step 5** Configure certificates using the PKCert.sh tool.
- Create directories for the secondary realms (for example, IPFONIX2.COM and IPFONIX3.COM) under *BAC_home/kdc/solaris/packetcable/certificates*.

For example:

```
/opt/CSCObpr/kdc/solaris/packetcable/certificates# mkdir IPFONIX2.COM
/opt/CSCObpr/kdc/solaris/packetcable/certificates# mkdir IPFONIX3.COM
```

For more information on creating directories, refer to Solaris documentation.

- Create a directory in which you can copy the following certificates:
 - CableLabs_Service_Provider_Root.cer
 - Service_Provider.cer
 - Local_System.cer

- MTA_Root.cer
- Local_System.der

For example:

```
# cd /var
# mkdir certsInput
```



Note The /certsInput directory created under the /var directory is only an example. You can choose to create any directory under any other directory. For more information on creating directories, refer to Solaris documentation.

- c. Copy the certificates mentioned in the previous step into the directory which you created. For information on copying files, refer to Solaris documentation on the **cp** command.
- d. Copy the following certificates to the *BAC_home/kdc/solaris/packetcable/certificates* directory:
 - CableLabs_Service_Provider_Root.cer
 - Service_Provider.cer
 - Local_System.cer
 - MTA_Root.cer

For information on copying files, refer to Solaris documentation on the **cp** command.

- e. Create the KDC certificate and its associated private key for the primary realm.

For example:

```
# ./opt/CSCObpr/kdc/PKCert.sh -c "-s /var/certsInput -d /var/certsOutput
-k /var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer
-r IPFONIX.COM -n 100 -a bactest.cisco.com -o"
```

```
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: IPFONIX.COM
Serial Number: 100
DNS Name of KDC: bactest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01
CableLabs Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer
```

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObpr/kdc/solaris/packetcable/certificates)

Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObpr/kdc/solaris)

Copy KDC_private_key_proprietary. to the KDC platform directory (i.e. /opt/CSCObpr/kdc/solaris)

For more information on the tool, see [Using the PKCert.sh Tool, page 13-5](#).

- f. Copy the *KDC.cer* file to the KDC certificate directory (*BAC_home/kdc/solaris/packetcable/certificates*). For information on copying files, refer to Solaris documentation on the **cp** command.
- g. Copy the private key *KDC_private_key.pkcs8* to the KDC platform directory (*BAC_home/kdc/solaris*). For information on copying files, refer to Solaris documentation on the **cp** command.
- h. Copy the private key *KDC_private_key_proprietary.* to the KDC platform directory (*BAC_home/kdc/solaris*). For information on copying files, refer to Solaris documentation on the **cp** command.
- i. Create the KDC certificate and its associated private key for the secondary realm; in this case, *IPFONIX2.COM*.

For example:

```
# ./opt/CSCObpr/kdc/PKCert.sh -c "-s /var/certsInput -d /var/certsOutput
-k /var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer
-r IPFONIX2.COM -n 100 -a bactest.cisco.com -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: IPFONIX.COM
Serial Number: 100
DNS Name of KDC: bactest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01
CableLabs Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObpr/kdc/solaris/
packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObpr/
kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e. /opt/CSCObpr/
kdc/solaris)
```

For more information on the tool, see [Using the PKCert.sh Tool, page 13-5](#).

- j. Copy *KDC.cer* to the secondary realm directory; for example, the */IPFONIX2.COM* directory under *BAC_home/kdc/solaris/packetcable/certificates*. For information on copying files, refer to Solaris documentation on the **cp** command.
- k. Copy the private key *KDC_private_key.pkcs8* to the secondary realm directory; for example, the */IPFONIX2.COM* directory under *BAC_home/kdc/solaris/packetcable/certificates*. For information on copying files, refer to Solaris documentation on the **cp** command.
- l. Copy the private key *KDC_private_key_proprietary.* to the secondary realm directory; for example, the */IPFONIX2.COM* directory under *BAC_home/kdc/solaris/packetcable/certificates*. For information on copying files, refer to Solaris documentation on the **cp** command.

- m. Create the KDC certificate and its associated private key for the secondary IPFONIX3.COM realm.

For example:

```
# ./opt/CSCObpr/kdc/PKCert.sh -c "-s /var/certsInput -d /var/certsOutput
-k /var/certsInput/Local_System.der -c /var/certsInput/Local_System.cer
-r IPFONIX3.COM -n 100 -a bactest.cisco.com -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: IPFONIX.COM
Serial Number: 100
DNS Name of KDC: bactest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01
CableLabs Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e. /opt/CSCObpr/kdc/solaris/
packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObpr/
kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e. /opt/CSCObpr/
kdc/solaris)
```

For information on the tool, see [Using the PKCert.sh Tool, page 13-5](#).

- n. Copy KDC.cer to the secondary realm directory; for example, the /IPFONIX3.COM directory under *BAC_home/kdc/solaris/packetcable/certificates*. For information on copying files, refer to Solaris documentation on the **cp** command.
- o. Copy the private key KDC_private_key.pkcs8 to the secondary realm directory; for example, the /IPFONIX3.COM directory under *BAC_home/kdc/solaris/packetcable/certificates*. For information on copying files, refer to Solaris documentation on the **cp** command.
- p. Copy the private key KDC_private_key_proprietary. to the secondary realm directory; for example, the /IPFONIX3.COM directory under *BAC_home/kdc/solaris/packetcable/certificates*. For information on copying files, refer to Solaris documentation on the **cp** command.

Step 6 Generate PacketCable service keys by using the KeyGen tool.



Note Ensure that the password that you use to generate a service key matches the password that you set on the DPE by using the **packetcable registration kdc service-key** command.

For example:

```
# /opt/CSCObpr/kdc/keygen bactest.cisco.com IPFONIX.COM changeme
# /opt/CSCObpr/kdc/keygen bactest.cisco.com IPFONIX2.COM changeme
# /opt/CSCObpr/kdc/keygen bactest.cisco.com IPFONIX3.COM changeme
```

For details, see [Using the KeyGen Tool, page 13-11](#).

Step 7 Ensure that the service keys you generated in step 6 exist in the *BAC_home/kdc/solaris/keys* directory.

For example:

```
/opt/CSCObpr/kdc/solaris/keys# ls -l
total 18
-rw-r--r--  1 root    other 2 Nov  4 09:44 krbtgt,IPFONIX2.COM@IPFONIX2.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 krbtgt,IPFONIX3.COM@IPFONIX3.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 krbtgt,IPFONIX.COM@IPFONIX.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 mtafqdnmap,bactest.cisco.com@IPFONIX2.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 mtafqdnmap,bactest.cisco.com@IPFONIX3.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 mtafqdnmap,bactest.cisco.com@IPFONIX.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 mtaprovsrvr,bactest.cisco.com@IPFONIX2.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 mtaprovsrvr,bactest.cisco.com@IPFONIX3.COM
-rw-r--r--  1 root    other 2 Nov  4 09:44 mtaprovsrvr,bactest.cisco.com@IPFONIX.COM
```

For more information, refer to Solaris documentation.

Step 8 Ensure that the various certificates and service keys exist in the *BAC_home/kdc* directory.

For example:

```
/opt/CSCObpr/kdc# ls
PKCert.sh  internal keygen lib pkcert.log  solaris bacckdc.license

/opt/CSCObpr/kdc# cd /internal/bin
/internal/bin# ls
kdc runKDC.sh shutdownKDC.sh

# cd /opt/CSCObpr/kdc/lib
# ls
libgcc_s.so.1      libstdc++.so.5      libstlport_gcc.so

# cd /opt/CSCObpr/solaris/logs
# ls
kdc.log  kdc.log.1

# cd /opt/CSCObpr/solaris
# ls
logs kdc.ini packetcable KDC_private_key_proprietary.

# cd keys
# ls
krbtgt,IPFONIX2.COM@IPFONIX2.COM
krbtgt,IPFONIX3.COM@IPFONIX3.COM
krbtgt,IPFONIX.COM@IPFONIX.COM
mtafqdnmap,bactest.cisco.com@IPFONIX2.COM
mtafqdnmap,bactest.cisco.com@IPFONIX3.COM
mtafqdnmap,bactest.cisco.com@IPFONIX.COM
mtaprovsrvr,bactest.cisco.com@IPFONIX2.COM
mtaprovsrvr,bactest.cisco.com@IPFONIX3.COM
mtaprovsrvr,bactest.cisco.com@IPFONIX.COM

# cd ./solaris/packetcable/certificates
# ls
KDC.cer
Local_System.cer
CableLabs_Service_Provider_Root.cer  MTA_Root.cer
IPFONIX2.COM                          Service_Provider.cer
IPFONIX3.COM

# cd ./solaris/packetcable/certificates/IPFONIX2.COM
# ls
KDC.cer
KDC_private_key_proprietary.
```

```
# cd ./solaris/packetcable/certificates/IPFONIX3.COM:
# ls
KDC.cer
KDC_private_key_proprietary.
```

For more information, refer to Solaris documentation.

Step 9 Restart the KDC.

For example:

```
# /etc/init.d/bprAgent restart kdc
```

For more information, see [Using the BAC Process Watchdog from the Command Line, page 2-14](#).

Step 10 Configure the BAC administrator user interface for multiple realms.

- a. Add DHCP criteria for the secondary realm; in this case, IPFONIX2.COM.

For example:

1. From **Configuration > DHCP Criteria > Manage DHCP Criteria**, click the **Add** button.
2. The Add DHCP Criteria page appears.
3. Enter **ipfonix2** in the DHCP Name field.
4. Click **Submit**.
5. Return to the Manage DHCP Criteria page, and click the ipfonix2 DHCP criteria. The Modify DHCP Criteria page appears.
6. Under Property Name, select `/ccc/kerb/realms` and enter IPFONIX2.COM in the Property Value field.
7. Click **Add** and **Submit**.

For more information, see [Configuring DHCP Criteria, page 11-24](#).

- b. Add DHCP criteria for the secondary realm; in this case, IPFONIX3.COM.

For example:

1. From **Configuration > DHCP Criteria > Manage DHCP Criteria**, click the **Add** button.
2. The Add DHCP Criteria page appears.
3. Enter **ipfonix3** in the DHCP Name field.
4. Click **Submit**.
5. Return to the Manage DHCP Criteria page, and click the ipfonix3 DHCP criteria. The Modify DHCP Criteria page appears.
6. Under Property Name, select `/ccc/kerb/realms` and enter IPFONIX3.COM in the Property Value field.
7. Click **Add** and **Submit**.

For more information, see [Configuring DHCP Criteria, page 11-24](#).

- c. Add templates as external files to BAC for each of the devices being provisioned; in this step, for the Motorola MTA.

For example:

1. Choose **Configuration > External Files**. The Manage External Files page appears.
2. Click **Add**, and the Add External Files page appears.

3. Add the `mot-mta.tmpl` external file. This file is the template used to provision a Motorola MTA. For template syntax, see [Example 5-2](#).
4. Click **Submit**.

For more information, see [Managing External Files, page 11-26](#).

- d. Add templates as external files to BAC for each of the devices being provisioned; in this step, for the Linksys MTA.

For example:

1. Choose **Configuration > External Files**. The Manage External Files page appears.
2. Click **Add**, and the Add External Files page appears.
3. Add the `linksys-mta.tmpl` external file. This file is the template used to provision a Linksys MTA. For template syntax, see [Example 5-3](#).
4. Click **Submit**.

For more information, see [Managing External Files, page 11-26](#).

- e. Add templates as external files to BAC for each of the devices being provisioned; in this step, for the SA MTA.

For example:

1. Choose **Configuration > External Files**. The Manage External Files page appears.
2. Click **Add**, and the Add External Files page appears.
3. Add the `sa-mta.tmpl` external file. This file is the template used to provision a Linksys MTA. For template syntax, see [Example 5-4](#).
4. Click **Submit**.

For more information, see [Managing External Files, page 11-26](#).

- f. Add a Class of Service for the primary realm; in this case, `IPFONIX.COM`.

For example:

1. Choose **Configuration > Class of Service**.
2. Click **Add**. The Add Class of Service page appears.
3. Enter `mot-mta` as the name of the new Class of Service for the `IPFONIX.COM` realm.
4. Choose the Class of Service Type as `PacketCableMTA`.
5. Select `/cos/packetCableMTA/file` from the Property Name drop-down list and associate it to the `mot-mta.tmpl` template file (which is used to provision the Motorola MTA in the primary `IPFONIX.COM` realm).
6. Click **Add** and **Submit**.

For more information, see [Configuring Class of Service, page 11-1](#).

- g. Add a Class of Service for the secondary realm; in this case, `IPFONIX2.COM`.

For example:

1. Choose **Configuration > Class of Service**.
2. Click **Add**. The Add Class of Service page appears.
3. Enter `linksys-mta` as the name of the new Class of Service for the `IPFONIX2.COM` realm.
4. Choose the Class of Service Type as `PacketCableMTA`.

5. Select `/cos/packetCableMTA/file` from the Property Name drop-down list and associate it to the `linksys-mta.tmpl` template file (which is used to provision the Linksys MTA in the secondary IPFONIX2.COM realm).
6. Click **Add** and **Submit**.

For more information, see [Configuring Class of Service, page 11-1](#).

- h. Add a Class of Service for the secondary realm; in this case, IPFONIX3.COM.

For example:

1. Choose **Configuration > Class of Service**.
2. Click **Add**. The Add Class of Service page appears.
3. Enter `sa-mta` as the name of the new Class of Service for the IPFONIX2.COM realm.
4. Choose the Class of Service Type as `PacketCableMTA`.
5. Select `/cos/packetCableMTA/file` from the Property Name drop-down list and associate it to the `sa-mta.tmpl` template file (which is used to provision the SA MTA in the secondary IPFONIX3.COM realm).
6. Click **Add** and **Submit**.

For more information, see [Configuring Class of Service, page 11-1](#).

- Step 11** Bring the devices online and provision them. Refer to the following examples that describe the provisioning process.

Example 1

The following example describes how you can provision the Motorola SBV5120.

- a. Provision the cable modem part of the device by setting it to use the **sample-bronze-docsis** Class of Service.
- b. To provision the MTA part, go to the **Devices > Manage Devices** page. Search and select the PacketCable device you want to provision. The Modify Device page appears.
- c. Set the domain name. This example uses `bacclab.cisco.com`.
- d. From the drop-down list corresponding to Registered Class of Service, select **mot-mta**. This is the Class of Service that you added in Step 10-f.
- e. From the drop-down list corresponding to Registered DHCP Criteria, select the **default** option.
- f. Click **Submit**.

[Figure 5-2](#) lists device details for the Motorola MTA.

Figure 5-2 Provisioning Motorola MTA—Device Details

Broadband Access Center for Cable Logout

Configuration **Devices** Nodes | Servers | Users

User:admin Role:Administrator

CISCO SYSTEMS **Modify Device**
Use this page to modify a device.
Fields marked with an "*" are required.

Modify Device	
Device Type:	PacketCableMTA
MAC Address:	<input type="text" value="1,8,aa:bb:cc:dd:ee:ff"/>
Host Name:	<input type="text" value="1-8-aa-bb-cc-dd-ee-ff"/>
Domain Name:	<input type="text" value="bacclab.cisco.com"/>
Owner Identifier:	<input type="text"/>
Registered Class Of Service:	<input type="text" value="mot-mta"/>
Registered DHCP Criteria:	<input type="text" value="default"/>

Property Name	Property Value
<input type="text" value="/IPDevice/mustBeBehindDevice"/>	<input type="text"/>

210860

Example 2

The following example illustrates how you can provision the Linksys CM2P2.

- a. Provision the cable modem part of the device by setting it to use the **sample-bronze-docsis** Class of Service.
- b. To provision the MTA part, go to the **Devices > Manage Devices** page. Search and select the PacketCable device you want to provision. The Modify Device page appears.
- c. Set the domain name. This example uses bacclab.cisco.com.
- d. From the drop-down list corresponding to Registered Class of Service, select **linksys-mta**. This is the Class of Service that you added in Step 10-g.
- e. From the drop-down list corresponding to Registered DHCP Criteria, select the **ipfonix2** option. This is the DHCP criteria that you added for the secondary IPFONIX2.COM realm in Step 10-a.
- f. Click **Submit**.

Figure 5-3 lists device details for the Linksys MTA.

Figure 5-3 Provisioning Linksys MTA—Device Details

Broadband Access Center for Cable Logout

Configuration **Devices** Nodes Servers Users

User:admin Role:Administrator

CISCO SYSTEMS **Modify Device**
Use this page to modify a device.
Fields marked with an "*" are required.

Modify Device

Device Type: PacketCableMTA

MAC Address:

Host Name:

Domain Name:

Owner Identifier:

Registered Class Of Service:

Registered DHCP Criteria:

Property Name	Property Value
<input type="text" value="/IPDevice/mustBeBehindDevice"/>	<input type="text"/>

210881

Example 3

The following example illustrates how you can provision the SA WebStar DPX 2203.

- a. Provision the cable modem part of the device by setting it to use the **sample-bronze-docsis** Class of Service.
- b. To provision the MTA part, go to the **Devices > Manage Devices** page. Search and select the PacketCable device you want to provision. The Modify Device page appears.
- c. Set the domain name. This example uses bacclab.cisco.com.
- d. From the drop-down list corresponding to Registered Class of Service, select **sa-mta**. This is the Class of Service that you added in Step 10-h.
- e. From the drop-down list corresponding to Registered DHCP Criteria, select the **ipfonix2** option. This is the DHCP criteria that you added for the secondary IPFONIX3.COM realm in Step 10-b.
- f. Click **Submit**.

Figure 5-4 lists device details for the SA MTA.

Figure 5-4 Provisioning SA MTA–Device Details

Broadband Access Center for Cable Logout

Configuration **Devices** Nodes | Servers | Users

User:admin Role:Administrator

CISCO SYSTEMS **Modify Device**
Use this page to modify a device.
Fields marked with an "*" are required.

Modify Device

Device Type:	PacketCableMTA
MAC Address:	<input type="text" value="1,8,cc:dd:ee:ff:aa:bb"/>
Host Name:	<input type="text" value="1-8-cc-dd-ee-ff-aa-bb"/>
Domain Name:	<input type="text" value="bacclab.cisco.com"/>
Owner Identifier:	<input type="text"/>
Registered Class Of Service:	<input type="text" value="sa-mta"/>
Registered DHCP Criteria:	<input type="text" value="ipfonix3"/>

Property Name	Property Value
<input type="text" value="/IPDevice/mustBeBehindDevice"/>	<input type="text"/>

210882

Step 12 Verify if multiple realm support is operational by using an ethereal trace. Refer to the sample output from the KDC and DPE log files shown here from the sample setup used in this procedure.

Example 1

The following example features excerpts from the KDC and DPE log files for the Motorola SBV 5120 MTA provisioned in the primary IPFONIX.COM realm:

KDC Log Sample Output–Motorola MTA

```
INFO [Thread-4] 2007-02-07 07:56:21,133 (DHHelper.java:114) - Time to create DH key pair(ms): 48
INFO [Thread-4] 2007-02-07 07:56:21,229 (DHHelper.java:114) - Time to create DH key pair(ms): 49
INFO [Thread-4] 2007-02-07 07:56:21,287 (DHHelper.java:150) - Time to create shared secret: 57 ms.
INFO [Thread-4] 2007-02-07 07:56:21,289 (PKAsReqMsg.java:104) - ##MTA-9a Unconfirmed AS Request: 1133717956 Received from /10.10.1.2
INFO [Thread-4] 2007-02-07 07:56:21,298 (KRBProperties.java:612) - Replacing property: 'minimum ps backoff' Old Value:'150' New Value: '150'
INFO [Thread-4] 2007-02-07 07:56:21,324 (KDCMessageHandler.java:257) - AS-REQ contains PKINIT - QA Tag.
INFO [Thread-4] 2007-02-07 07:56:21,325 (KDCMessageHandler.java:279) - PK Request from MTA received. Client is MTA - QA Tag
INFO [Thread-4] 2007-02-07 07:56:21,365 (KDCMessageHandler.java:208) - ##MTA-9b KDC Reply AS-REP Sent to /10.10.1.2:1039 Time(ms): 290
WARN [main] 2005-11-07 07:56:23,193 (KDC.java:113) - Statistics Report ASREP's: 1
```

```

INFO [main] 2005-11-07 07:56:23,195 (KDC.java:121) - /pktcbl/mtaAsRepSent: 10
INFO [main] 2005-11-07 07:56:23,195 (KDC.java:121) - /pktcbl/DHKeygenTotalTime: 1043
INFO [main] 2005-11-07 07:56:23,196 (KDC.java:121) - /pktcbl/mtaAsReqRecvd: 10
INFO [main] 2005-11-07 07:56:23,197 (KDC.java:121) - /pktcbl/DHKeygenNumOps: 20
INFO [main] 2005-11-07 07:56:23,197 (KDC.java:121) - /pktcbl/total: 60

```

DPE Log Sample Output—Motorola MTA

```

dpe.cisco.com: 2007 02 07 07:56:24 EST: %BPR-DPE-6-4178: Adding Replay Packet: []
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BPR-PKTSNMP-6-0764: [System Description for MTA:
<<HW_REV: 1.0, VENDOR: Motorola Corporation, BOOTR: 8.1, SW_REV:
SBV5120-2.9.0.1-SCM21-SHPC, MODEL: SBV5120>>]
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BPR-PKTSNMP-6-0764: [##MTA-15 SNMPv3 INFORM
Received From 10.10.1.2.]
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BPR-DPE-6-0688: Received key material update for
device [1,6,01:11:82:61:5e:30]
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BPR-PKTSNMP-6-0764: [##MTA-19 SNMPv3 SET Sent to
10.10.1.2]
dpe.cisco.com: 2007 02 07 07:56:24 EST: %BPR-TFTP-6-0310: Finished handling [read] request
from [10.10.1.2:1190] for [bpr0106001182615e300001]
dpe.cisco.com: 2007 02 07 07:56:25 EST: %BPR-PKTSNMP-6-0764: [##MTA-25 SNMP Provisioning
State INFORM Received from 10.10.1.2. Value: 1]

```

Example 2

The following example features excerpts from the KDC and DPE log files for the Linksys CM2P2 MTA provisioned in the secondary IPFONIX2.COM realm:

KDC Log Sample Output—Linksys MTA

```

INFO [Thread-8] 2007-02-07 08:00:10,664 (DHHelper.java:114) - Time to create DH key
pair(ms): 49
INFO [Thread-8] 2007-02-07 08:00:10,759 (DHHelper.java:114) - Time to create DH key
pair(ms): 49
INFO [Thread-8] 2007-02-07 08:00:10,817 (DHHelper.java:150) - Time to create shared
secret: 57 ms.
INFO [Thread-8] 2007-02-07 08:00:10,819 (PKAsReqMsg.java:104) - ##MTA-9a Unconfirmed AS
Request: 1391094112 Received from /10.10.1.5
INFO [Thread-8] 2007-02-07 08:00:10,828 (KRBProperties.java:612) - Replacing property:
'minimum ps backoff' Old Value:'150' New Value: '150'
INFO [Thread-8] 2007-02-07 08:00:10,860 (KDCMessageHandler.java:257) - AS-REQ contains
PKINIT - QA Tag.
INFO [Thread-8] 2007-02-07 08:00:10,862 (KDCMessageHandler.java:279) - PK Request from
MTA received. Client is MTA - QA Tag
INFO [Thread-8] 2007-02-07 08:00:10,901 (KDCMessageHandler.java:208) - ##MTA-9b KDC Reply
AS-REP Sent to /10.10.1.5:3679 Time(ms): 296
WARN [main] 2007-02-07 08:00:13,383 (KDC.java:113) - Statistics Report ASREP's: 1
INFO [main] 2007-02-07 08:00:13,384 (KDC.java:121) - /pktcbl/mtaAsRepSent: 11
INFO [main] 2007-02-07 08:00:13,384 (KDC.java:121) - /pktcbl/DHKeygenTotalTime: 1141

```

DPE Log Sample Output—Linksys MTA

```

dpe.cisco.com: 2007 02 07 08:00:10 EST: %BPR-DPE-6-4112: Adding Replay Packet: []
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BPR-DPE-6-4178: Adding Replay Packet: []
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BPR-PKTSNMP-6-0764: [System Description for MTA:
Linksys Cable Modem with 2 Phone Ports (CM2P2) <<HW_REV: 2.0, VENDOR: Linksys, BOOTR:
2.1.6V, SW_REV: 2.0.3.3.11-1102, MODEL: CM2P2>>]
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BPR-PKTSNMP-6-0764: [##MTA-15 SNMPv3 INFORM
Received From 10.10.1.5.]
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BPR-DPE-6-0688: Received key material update for
device [1,6,00:0f:68:f9:42:f6]
dpe.cisco.com: 2007 02 07 08:00:12 EST: %BPR-PKTSNMP-6-0764: [##MTA-19 SNMPv3 SET Sent to
10.10.1.5]

```

```
dpe.cisco.com: 2007 02 07 08:00:18 EST: %BPR-TFTP-6-0310: Finished handling [read] request
from [10.10.1.5:1032] for [bpr0106000f68f942f60001]
dpe.cisco.com: 2007 02 07 08:00:18 EST: %BPR-PKTSNMP-6-0764: [##MTA-25 SNMP Provisioning
State INFORM Received from 10.10.1.5. Value: 1]
```

Example 3

The following example features excerpts from the KDC and DPE log files for the SA WebStar DPX 2203 MTA provisioned in the secondary IPFONIX3.COM realm:

KDC Log Sample Output—SA MTA

```
INFO [Thread-6] 2007-02-07 08:01:31,556 (DHHelper.java:114) - Time to create DH key
pair(ms): 49
INFO [Thread-6] 2007-02-07 08:01:31,652 (DHHelper.java:114) - Time to create DH key
pair(ms): 50
INFO [Thread-6] 2007-02-07 08:01:31,711 (DHHelper.java:150) - Time to create shared
secret: 57 ms.
INFO [Thread-6] 2007-02-07 08:01:31,715 (PKAsReqMsg.java:104) - ##MTA-9a Unconfirmed AS
Request: 575634000 Received from /10.10.1.50
INFO [Thread-6] 2007-02-07 08:01:31,727 (KRBProperties.java:612) - Replacing property:
'minimum ps backoff' Old Value:'150' New Value: '150'
INFO [Thread-6] 2007-02-07 08:01:31,752 (KDCMessageHandler.java:257) - AS-REQ contains
PKINIT - QA Tag.
INFO [Thread-6] 2007-02-07 08:01:31,753 (KDCMessageHandler.java:279) - PK Request from
MTA received. Client is MTA - QA Tag
INFO [Thread-6] 2007-02-07 08:01:31,792 (KDCMessageHandler.java:208) - ##MTA-9b KDC Reply
AS-REP Sent to /10.10.1.50:3679 Time(ms): 292
WARN [main] 2007-02-07 08:01:33,423 (KDC.java:113) - Statistics Report ASREP's: 1
INFO [main] 2007-02-07 08:01:33,424 (KDC.java:121) - /pktcbl/mtaAsRepSent: 12
INFO [main] 2007-02-07 08:01:33,425 (KDC.java:121) - /pktcbl/DHKeygenTotalTime: 1240
INFO [main] 2007-02-07 08:01:33,425 (KDC.java:121) - /pktcbl/mtaAsReqRecvd: 12
INFO [main] 2007-02-07 08:01:33,426 (KDC.java:121) - /pktcbl/DHKeygenNumOps: 24
INFO [main] 2007-02-07 08:01:33,426 (KDC.java:121) - /pktcbl/total: 72
```

DPE Log Sample Output—SA MTA

```
dpe.cisco.com: 2007 02 07 08:01:31 EST: %BPR-DPE-6-4112: Adding Replay Packet: []
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BPR-DPE-6-4178: Adding Replay Packet: []
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BPR-PKTSNMP-6-0764: [System Description for MTA:
S-A WebSTAR DPX2200 Series DOCSIS E-MTA Ethernet+USB (2)Lines VOIP <<HW_REV: 2.0, VENDOR:
S-A, BOOTR: 2.1.6b, SW_REV: v1.0.1r1133-0324, MODEL: DPX2203>>]
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BPR-PKTSNMP-6-0764: [##MTA-15 SNMPv3 INFORM
Received From 10.10.1.50.]
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BPR-DPE-6-0688: Received key material update for
device [1,6,00:0f:24:d8:6e:f5]
dpe.cisco.com: 2007 02 07 08:01:33 EST: %BPR-PKTSNMP-6-0764: [##MTA-19 SNMPv3 SET Sent to
10.10.1.50]
dpe.cisco.com: 2007 02 07 08:01:38 EST: %BPR-TFTP-6-0310: Finished handling [read] request
from [10.10.1.50:1037] for [bpr0106000f24d86ef50001]
dpe.cisco.com: 2007 02 07 08:01:39 EST: %BPR-PKTSNMP-6-0764: [##MTA-25 SNMP Provisioning
State INFORM Received from 10.10.1.50. Value: 1]
```

Authoring Template for Provisioning Devices in Multiple Realms

You can use the template syntax described here to provision devices in a particular realm. The examples shown here are specific to the Motorola SBV5120 MTA, the Linksys CM2P2 MTA, and the SA WebStar DPX2203 MTA. You must modify these templates to suit the specifics of the MTA in your network.

Example 5-2 Template Used to Provision a Motorola MTA

```

#
# Example PacketCable MTA template: mot-mta.tmpl
#
# Note that this template is specific to the TI 401 MTA.
# This template must be modified to the specifics of your MTA.
#
# First, the start marker.
#
option 254 1
#
# Enable MTA
#
option 11 .pktcMtaDevEnabled.0,INTEGER,true
#
# Set CMS FQDN for each endpoint on the MTA.
# NOTE: the indexes (9 and 10 here) will differ per manufacturer.
#
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.9,STRING
,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,STRIN
G,CMS.CISCO.COM
#
# Set the realm org name. This MUST match that contained in the cert chain used by the
device.
#
# "CableLabs, Inc."
option 11
.pktcMtaDevRealmTable.pktcMtaDevRealmEntry.pktcMtaDevRealmOrgName.'IPFONIX.COM',STRING,"'4
3:61:62:6C:65:4C:61:62:73:2C:20:49:6E:63:2E'"
#
# Set the realm name and IPsec control for the CMS.
#
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsIpsecCtrl.'CMS.CISCO.COM',INTEGER,true
option 11
pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsKerbRealmName.'CMS.CISCO.COM',STRING,IP
FONIX.COM
#
# Finally, the end marker.
#
option 254 255

```

Example 5-3 Template Used to Provision a Linksys MTA

Note that, in this template, the realm has been set to IPFONIX2.COM.

```

#
# Example PacketCable MTA template: linksys-mta.tmpl
#
# Note that this template is specific to the TI 401 MTA.
# This template must be modified to the specifics of your MTA.
#
# First, the start marker.
#
option 254 1
#
# Enable MTA
#
option 11 .pktcMtaDevEnabled.0,INTEGER,true

```



```

#
# Set CMS FQDN for each endpoint on the MTA.
# NOTE: the indexes (9 and 10 here) will differ per manufacturer.
#
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.9,STRING
,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,STRIN
G,CMS.CISCO.COM
#
# Set the realm org name. This MUST match that contained in the cert chain used by the
device.
#
# "CableLabs, Inc."
option 11
.pktcMtaDevRealmTable.pktcMtaDevRealmEntry.pktcMtaDevRealmOrgName.'IPFONIX2.COM',STRING,"'
43:61:62:6C:65:4C:61:62:73:2C:20:49:6E:63:2E'"
#
# Set the realm name and IPsec control for the CMS.
#
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsIpsecCtrl.'CMS.CISCO.COM',INTEGER,true
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsKerbRealmName.'CMS.CISCO.COM',STRING,IP
FONIX2.COM
#
# Finally, the end marker.
#
option 254 255

```

Example 5-4 Template Used to Provision an SA MTA

Note that, in the template, the realm has been set to IPFONIX3.COM.

```

#
# Example PacketCable MTA template: sa-mta.tmpl
#
# Note that this template is specific to the TI 401 MTA.
# This template must be modified to the specifics of your MTA.
#
# First, the start marker.
#
option 254 1
#
# Enable MTA
#
option 11 .pktcMtaDevEnabled.0,INTEGER,true
#
# Set CMS FQDN for each endpoint on the MTA.
# NOTE: the indexes (9 and 10 here) will differ per manufacturer.
#
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.9,STRING
,CMS.CISCO.COM
option 11
.pktcNcsEndPntConfigTable.pktcNcsEndPntConfigEntry.pktcNcsEndPntConfigCallAgentId.10,STRIN
G,CMS.CISCO.COM
#
# Set the realm org name. This MUST match that contained in the cert chain used by the
device.
#

```

```

# "CableLabs, Inc."
option 11
.pktcMtaDevRealmTable.pktcMtaDevRealmEntry.pktcMtaDevRealmOrgName.'IPFONIX3.COM',STRING,"'
43:61:62:6C:65:4C:61:62:73:2C:20:49:6E:63:2E'"
#
# Set the realm name and IPsec control for the CMS.
#
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsIpsecCtrl.'CMS.CISCO.COM',INTEGER,true
option 11
.pktcMtaDevCmsTable.pktcMtaDevCmsEntry.pktcMtaDevCmsKerbRealmName.'CMS.CISCO.COM',STRING,IP
FONIX3.COM
#
# Finally, the end marker.
#
option 254 255

```

PacketCable Basic eMTA Provisioning

BAC also supports PacketCable Basic, which offers a simpler, DOCSIS-like, non-secure provisioning flow. [Table 5-5](#) describes the BASIC.1 flow using the provisioning workflow in [Figure 5-1](#).

Table 5-5 PacketCable Basic eMTA Provisioning

Step	Workflow	Description
MTA-1	DHCP Broadcast Discover	Executes as for the Secure flow.
MTA-2	DHCP Offer	If the provisioning system is configured to provision the MTA in BASIC.1 mode, the provisioning system returns a DHCP Offer containing Option 122 suboption 6, which contains the special reserved realm name “BASIC.1”. This reserved realm name commands the MTA to use the BASIC.1 provisioning flow. This Offer also contains the provisioning system IP address in Option 122.3, and the file and siaddr fields are populated with the configuration file location of the MTA.
MTA-3	DHCP Request	The remainder of the MTA DHCP exchange is executed (Request and Ack exchanged).
MTA-4	DHCP Ack	
MTA-22	Telephony Config File Request	The MTA skips directly to Step MTA-22. Using the file and siaddr information, the MTA copies its configuration file from the provisioning system via TFTP. Note that BAC integrates the TFTP server into the DPE component. Note No authentication of MTA/provisioning server or encryption occurs.
MTA-23	Telephony Config File	

The BASIC.2 flow is identical to BASIC.1, with the following exceptions:

- “BASIC.2” is populated into the MTA’s DHCP Option 122 suboption 6.
- The MTA issues a provisioning status SNMPv2c INFORM at the very end of the flow, MTA-25 (DHCP Option 122 suboption 3 specifies the Inform target).

The PacketCable Basic flow is similar to the DOCSIS flow with the following differences:

- There is no ToD exchange between MTA and the provisioning system.
- The MTA configuration file contains an integrity hash. Specifically, the SHA1 hash of the entire content of the configuration file is populated into a pktcMtadevConfigFileHash SNMP VarBind and placed within a TLV 11 just before the end of file TLV.
- BASIC.2 flow issues a provisioning status SNMPv2c Inform after the MTA receives and processes its configuration file. This Inform notifies BAC if MTA provisioning completed successfully. If there is a problem, an error is generated and an event sent from the DPE to the RDU, then on to a BAC client. This Inform is useful while debugging configuration file issues.

For additional information about the DOCSIS flow, see [DOCSIS Configuration, page 4-1](#).

**Note**

Before using the PacketCable Basic provisioning flow, ensure that you are using a PacketCable Basic-capable eMTA. The eMTA must report that it is Basic-capable with its DHCP Discover Option 60, TLV 5.18 (supported flows).

PacketCable TLV 38 and MIB Support

BAC supports the complete set of PacketCable 1.5 MIBs.

BAC supports TLV 38 in PacketCable configuration templates. This TLV lets you configure multiple SNMP notification targets. Configuration of this TLV means that all notifications are also issued to the targets configured through TLV 38.

SNMP v2C Notifications

BAC supports both SNMP v2C TRAP and INFORM notifications from the PacketCable MTA.

Euro PacketCable

Euro PacketCable services are essentially the European equivalent of North American PacketCable services with the following differences:

- Euro PacketCable uses different MIBs.
- Euro PacketCable uses a different set of device certificates (MTA_Root.cer) and service provider certificates (Service Provider Root).

For Euro PacketCable certificates, the kdc.ini file must have the `euro-packetcable` property set to true. The KDC supports Euro PacketCable (tComLabs) certificate chains. The following is a sample Euro PacketCable-enabled KDC configuration file.

```
[general]
interface address = 10.10.10.1
FQDN = servername.cisco.com
maximum log file size = 10000
n saved log files = 100
log debug level = 5 minimum
ps backoff = 150 maximum
ps backoff = 300
euro-packetcable = true
```

When using Euro PacketCable, ensure that the value of the PacketCable property `/pktcbl/prov/locale` is set to EURO. The default is NA (for North America). You can specify the locale in the Configuration File utility. See [Using the Configuration File Utility, page 8-27](#), for more information.

Euro PacketCable MIBs

Euro PacketCable MIBs are essentially snapshots of draft-IETF MIBs. MTA configuration files consist essentially of SNMP VarBinds that reference the MIBs. There are substantial differences between the North American PacketCable and Euro PacketCable MIBs; therefore, the North American PacketCable and Euro PacketCable configuration files are incompatible. During installation, sample files for North American PacketCable (`cw29_config.tmpl`) and Euro PacketCable (`ecw15_mta_config.tmpl`) are copied to the `BAC_home/rdu/samples` directory.

BAC ships with the following Euro PacketCable MIBs:

- DOCS-IETF-BPI2-MIB
- INTEGRATED-SERVICES-MIB
- DIFFSERV-DSCP-TC
- DIFFSERV-MIB
- TCOMLABS-MIB
- PKTC-TCOMLABS-MTA-MIB
- PKTC-TCOMLABS-SIG-MIB

Configuring Euro PacketCable MIBs

To configure BAC to use Euro PacketCable MIBs, you must change the BAC RDU property that specifies the MIBs to be loaded. By default, this property contains the PacketCable MIBs.

You can change the property in one of the following ways:

- Modify `rdu.properties` and restart the RDU.
- On the administrator user interface, navigate to **Configuration > Defaults > System Defaults** and replace the MIB list with the list shown below. You do not need to restart the RDU.
- Use the Prov API `changeSystemDefaults()` call. You do not need to restart the RDU.

The property name is `/snmp/mibs/mibList` (properties file) or `SNMPPropertyKeys.MIB_LIST` (the Prov API constant name). The property value is a comma-separated value (CSV) consisting of the required MIB names, as shown:

```
/snmp/mibs/mibList=SNMPv2-SMI,SNMPv2-TC,INET-ADDRESS-MIB,CISCO-SMI,CISCO-TC,SNMPv2-MIB,RFC1213-MIB,IANAifType-MIB,IF-MIB,DOCS-IF-MIB,DOCS-IF-EXT-MIB,DOCS-BPI-MIB,CISCO-CABLE-SPECTRUM-MIB,CISCO-DOCS-EXT-MIB,SNMP-FRAMEWORK-MIB,DOCS-CABLE-DEVICE-MIB,DOCS-CABLE-DEVICE-MIB-OBSOLETE,DOCS-QOS-MIB,CISCO-CABLE-MODEM-MIB,DOCS-IETF-BPI2-MIB,INTEGRATED-SERVICES-MIB,DIFFSERV-DSCP-TC,DIFFSERV-MIB,TCOMLABS-MIB,PKTC-TCOMLABS-MTA-MIB,PKTC-TCOMLABS-SIG-MIB
```



CHAPTER 6

Troubleshooting PacketCable eMTA Provisioning

This chapter features information that will help you solve possible issues in a PacketCable voice technology deployment.

- [Troubleshooting Tools](#), page 6-4
- [Troubleshooting Scenarios](#), page 6-5
- [Certificate Trust Hierarchy](#), page 6-9

This chapter assumes that you are familiar with the PacketCable Media Terminal Adapter (MTA) Device Provisioning Specification, PKT-SPPROV1.5-I01-050128. See the PacketCable website for details.

Provisioning PacketCable embedded MTAs (eMTAs) is a relatively complex process; however, with the right tools and ‘tricks of the trade,’ getting eMTAs operational can be straightforward.

This chapter assumes that Network Registrar and BAC are both in use; however, much of the information also applies for other deployments. Basic knowledge of Network Registrar (scopes, policies, basic DNS zone setup, and record entry) and BAC (Class of Service, DHCP criteria, external files, and BAC directory structure) is assumed.

The PacketCable eMTA provisioning process consists of 25 steps for the Secure flow; the Basic flow has far fewer steps. To troubleshoot eMTAs, knowledge of these 25 steps from the PacketCable provisioning specification is absolutely essential. See [PacketCable Voice Configuration](#), page 5-1.

This section contains the following topics:

- [Components](#), page 6-1
- [Key Variables](#), page 6-3

Components

Before troubleshooting eMTAs, you should be familiar with the following system components.

- [eMTA](#)
- [DHCP Server](#)
- [DNS Server](#)
- [KDC](#)
- [PacketCable Provisioning Server](#)
- [Call Management Server](#)

eMTA

The eMTA is a cable modem and an MTA in one box, with a common software image. The CM and MTA each have their own MAC addresses and each performs DHCP to get its own IP address. The eMTA contains, at minimum, three certificates. One certificate is a unique MTA certificate. A second certificate identifies the MTA manufacturer. Both the device and manufacture certificates are sent by the MTA to authenticate itself to the KDC. The third certificate is a telephony root certificate used to verify the certificates sent by the KDC to the MTA. The KDC certificates will be chained from the telephony root, therefore the telephony root must reside on the MTA to validate the authenticity of the KDC certificates. The MTA portion receives its own configuration file, which it uses to identify its controlling call agent, among other things.

DHCP Server

The DOCSIS specifications mandate that cable modems negotiate their IP address using the Dynamic Host Configuration Protocol (DHCP). The MTA, like most CPE on a DOCSIS network, must use DHCP to obtain its IP address and other crucial information (DNS servers, PacketCable Option 122 for Kerberos realm name, provisioning server FQDN).



Note

The cable modem portion, in addition to its normally required DHCP options, also requests, and must receive, Option 122 suboption 1, which it passes to the MTA portion as the IP address of the correct DHCP server from which to accept offers.

When using BAC with PacketCable support, be aware that a correctly configured BAC will automatically populate the ToD server, DNS servers, TFTP server, as well as the Option 122 fields; these do not need to be explicitly set in the Network Registrar policy.

DNS Server

The Domain Name System (DNS) server is fundamental in PacketCable provisioning. The PacketCable provisioning server, which is the device provisioning engine (DPE) in a BAC architecture, must have an address (A) record in the appropriate zone, because its fully qualified domain name (FQDN) is provided to the MTA in Option 122 by the DHCP server. The KDC realm must have a zone of the same name as the realm name containing a server (SRV) record that contains the FQDN of the Kerberos server.

The Kerberos server identified in the SRV record must itself have an A record in the appropriate zone. The call management server (CMS) identified in the MTA configuration file must also have an A record in the appropriate zone. Lastly, the MTAs themselves must have A records in the appropriate zone, since the CMS reaches the MTA by resolving its FQDN. Dynamic DNS (DDNS) is the preferred method of creating A records for the MTA. Refer to Cisco Network Registrar documentation for information on configuring and troubleshooting DDNS.

KDC

The KDC is responsible for authenticating 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.

PacketCable Provisioning Server

The PacketCable provisioning server is responsible for communicating the location of the MTA configuration file to the MTA, and/or provisioning MTA parameters via SNMP. SNMPv3 is used for all communication between the MTA and the provisioning server. The keys used to initiate SNMPv3 communication are obtained by the MTA during its authentication phase with the KDC. Provisioning server functionality is provided by the DPE in a BAC architecture.

Call Management Server

The call management server (CMS) is essentially a soft switch, or call-agent, with additional PacketCable functionality to control, among other things, quality of service on a cable network. The MTA sends a network call signaling (NCS) restart in progress (RSIP) message to the CMS upon successful PacketCable provisioning.

Key Variables

This section describes the key variables required to provision an eMTA correctly.

- [Certificates, page 6-3](#)
- [Scope-Selection Tags, page 6-4](#)
- [MTA Configuration File, page 6-4](#)

Certificates

The `MTA_Root.cer` file contains the MTA root certificate (a certificate that is rooted in the official PacketCable MTA root).

You must know in advance what telephony root certificate is required for the MTAs you want to provision. Deployments in production networks use telephony certificates rooted in the PacketCable real root. There is also a PacketCable test root used in lab and testing environments.

The KDC certificates used by the KDC to authenticate itself to the MTA must be rooted in the same telephony root that is stored on the MTA (PacketCable real or test root). Most MTA vendors support test images that have Telnet and/or HTTP login capabilities so that you can determine which telephony root is enabled, and change the root used (in most cases, you can only select between the PacketCable real or test root).

The most common scenario has the KDC loaded with certificates (from the `BAC_home/kdc/solaris/packetcable/certificates` directory) as follows:

- `CableLabs_Service_Provider_Root.cer`
- `Service_Provider.cer`
- `Local_System.cer`
- `KDC.cer`
- `MTA_Root.cer`

The first four certificates comprise the telephony certificate chain. The `MTA_Root.cer` file contains the MTA root used by the KDC to validate the certificates sent by the MTA.

**Note**

Refer to [Using the PKCert.sh Tool, page 13-5](#), for information on installing and managing KDC certificates.

To determine if you are using PacketCable test root, open the `CableLabs_Service_Provider_Root.cer` file in Windows, and validate that the Subject OrgName entry is **O = CableLabs**, and/or check that the Subject Alternative name reads **CN=CABLELABS GENERATED TEST ROOT FOR EQUIPMENT TEST PURPOSES ONLY**.

The KDC certificate (`KDC.cer`) contains the realm name to use. The realm name that BAC (and the corresponding DNS zone) is configured to use must match this realm name. Additionally, the MTA configuration file realm org name must match the organization name as seen in the telephony root.

The KDC certificate has a corresponding private key that must be installed in the `BAC_home/kdc/solaris` directory. Usually it is named `KDC_private_key.pkcs8` or `KDC_private_key_proprietary..`. When changing certificates, you must also change the private key.

Scope-Selection Tags

In most scenarios, BAC is involved in processing all DHCP requests from scopes with scope-selection tags that match selection criteria specified in the DHCP Criteria page of the BAC administrator user interface. Client class can also be used to tie scopes to BAC processing; ensure you make this association before you attempt to provision devices.

MTA Configuration File

The MTA configuration file contains the location of the CMS. Additionally, it must contain an entry for Realm Name. This value must match that of the certificate chain in use.

Certain table entries within the MTA configuration file are indexed by the realm name delivered to the MTA in Option 122. This realm name entry in the MTA configuration file must match that delivered in Option 122. For example, if **DEF.COM** was the realm name delivered in Option 122, MTA configuration file entries in the `pkcMtaDevRealm` table would be indexed with a suffix made up of the ASCII-coded character values (in dot delimited decimal format when using the Cisco Broadband Configurator) of the realm name, for example `68.69.70.46.67.79.77`. There are many free ASCII conversion pages available on the web to make this conversion easier.

Troubleshooting Tools

The 25 eMTA Secure provisioning steps contained in the PacketCable MTA Device Provisioning Specification are shown in [Figure 5-1](#). This section describes:

- [Logs, page 6-5](#)
- [Ethereal, SnifferPro, or Other Packet Capture Tools, page 6-5](#)

Logs

These log files are used to maintain the following information:

- The Network Registrar has two logs (`name_dhcp_1_log` and `name_dns_1_log`), which contain the most recent logging entries from Network Registrar. Look in these files for DHCP- or DNS-related problems.
- The `BAC_home/kdc/logs/kdc.log` file shows all KDC interactions with MTAs, and KDC interactions with the DPE.
- The `BAC_data/dpe/logs/dpe.log` file shows the major steps related to SNMPv3 interaction with the MTA. You can also use the **show log** CLI command if you are working with the hardware DPE.



Note

Turning on the tracing of snmp, registration server and registration server detail messages, using the command line interface, helps to troubleshoot potential PacketCable problems. For information on the appropriate troubleshooting commands, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

Ethereal, SnifferPro, or Other Packet Capture Tools

A packet capture tool is indispensable when troubleshooting the eMTAs. The Ethereal version, as packaged by CableLabs, includes numerous packet decoders specific to PacketCable. These include the Kerberos AS and AP packets.

- If you suspect that a specific failure is DHCP-related, capture packets while filtering on packets sourced from, or destined to, the CMTS cable interface IP address and the DHCP server IP address.
- If you suspect that a specific failure is related to any of the 25 steps occurring after DHCP, filter all packets to and from the eMTA IP address. This provides a very concise, easy-to-follow trace of provisioning steps 5 through 25, as shown in [Figure 5-1](#).

Troubleshooting Scenarios

The scenarios listed in [Table 6-1](#) are possible failures involving eMTAs.

Table 6-1 Troubleshooting Scenarios

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The KDC does not start.	The KDC certificate does not correspond to the private key.	Ensure that you have matching certificates and private key.
	The KDC license expired or is missing.	Restore KDC license to <code>BPR_HOME/kdc</code> directory.

Table 6-1 Troubleshooting Scenarios (continued)

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The MTA device does not appear in the BAC Devices page.	An incorrect cable helper address may have been configured.	Fix the helper address.
	The scope-selection tags do not match the DHCP criteria selected in the BAC user interface.	Verify that the MTA scope-selection tags match those in the PacketCable DHCP criteria created, in BAC, for the relevant MTAs.
	The Network Registrar extension point is not properly installed.	Re-install the Network Registrar extension point. Refer to the <i>Installation and Setup Guide for Cisco Broadband Access Center, 2.7.1</i> .
	The cable modem portion did not receive Option 122.	Verify that the tags on the scope of the cable modem portion match the DOCSIS DHCP criteria configure for BAC.
The MTA device does not accept the DHCP offer and continually cycles through the DHCP flow.	There are invalid DHCP options configured.	Check that scope policy includes DNS server option, and/or check that the <i>cnr_ep.properties</i> file includes entries for primary and secondary DNS servers.
	The DHCP offer may have come from a DHCP server different from the one indicated in the cable modem portion's Option 122 suboption 1.	Check the <i>cnr_ep.properties</i> file to ensure that the main and backup DHCP servers are set correctly.
Both the <i>kdc.log</i> file and the ethereal trace indicate that the MTA device never contacts the KDC.	An incorrect DNS server is specified in the <i>cnr_ep.properties</i> file or the MTA scope policy, or both.	Check or correct <i>cnr_ep.properties</i> DNS servers.
	A zone is missing or has been incorrectly set up for the Kerberos realm.	Make sure a zone with same name as realm is created and contains an 'SRV' record of format '_kerberos._udp 0 0 88 KDC FQDN'.
	There is a missing or incorrect KDC 'A' record entry.	Ensure that an 'A' record exists for the FQDN contained in the Kerberos zone's 'SRV' record.
	The DPE FQDN cannot be resolved.	Ensure that the provFQDNs entry in <i>dpe.properties</i> has the correct FQDN and IP of the DPE.

Table 6-1 Troubleshooting Scenarios (continued)

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The KDC reports failure during the Kerberos AS-Request.	The MTA certificate does not match the MTA root used by KDC.	Verify that the <i>MTA_Root.cer</i> is correct by comparing the <i>MTA_Root.cer</i> against that used on a working system. If it is correct, the MTA itself could have a certificate problem. This situation is extremely rare and if this is the case, contact the MTA manufacturer.
	FQDN lookup by KDC to Prov Server failed. The device may not yet be provisioned in BAC.	Verify that the device appears. It should be given both a Class of Service and a DHCP criteria.
	A clock skew error. See PacketCable Checklists, page 3-6 , for additional information.	Ensure that all BAC network elements are clock-synced via NTP. Refer to the <i>Broadband Access Center DPE CLI Reference, 2.7.1</i> .
	A mismatch may exist between the KDC and the DPE. Note If other devices are provisioned correctly, this is probably not the cause of the problem.	Check that these three entries exist in the <i>BPR_HOME/kdc/solaris/keys</i> directory: <ul style="list-style-type: none"> • mtafqdnmap,dpe.abc.com@DEF.COM • mtaprovsrvr,dpe.abc.com@DEF.COM • krbtgt,DEF.COM@DEF.COM The DPE FQDN and realm name on your system will be different from this example. Contents of these entries must match the entry in either the dpe.properties 'KDCServiceKey' entry, or the keys generated using the KeyGen utility.
The KDC reports success at the AS-Request/Reply (steps 9 and 10 in Figure 5-1) but the MTA device never moves past step 9.	There is a certificate mismatch between the telephony root loaded or enabled on the MTA, and that loaded on the KDC.	Check certificates on MTA and KDC.
	Although highly unlikely, it is possible that there is a corrupted telephony certificate chain. Note If other devices are provisioned correctly, this is not the cause of the problem.	Ensure that the correct certificate is loaded or enabled on MTA. If no devices can be provisioned correctly, try a different certificate on the KDC.
Failure at AP Request/Reply (step 14 in Figure 5-1).	A clock skew error. See PacketCable Checklists, page 3-6 , for additional information.	Ensure that all BAC network elements are clock-synced via NTP. Refer to the <i>Broadband Access Center DPE CLI Reference</i> .
	Cannot resolve Prov Server FQDN.	Make sure that the provisioning server (DPE) has a correct DNS entry. Ensure that dpe.properties provFQDNs entry has the correct FQDN and IP of the provisioning server (DPE).
	There is no route from the MTA to the DPE.	Correct the routing problem.
The MTA device never issues a TFTP request for a configuration file.	There is no route to the TFTP server running on the DPE.	Correct the routing problem.

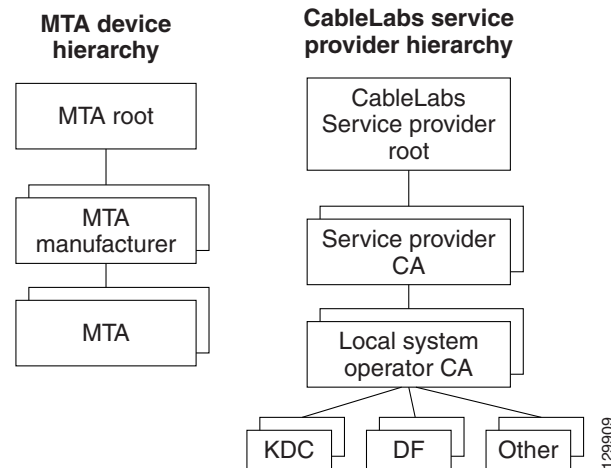
Table 6-1 Troubleshooting Scenarios (continued)

If this problem occurs...	Which indicates this potential cause...	To correct it, you should...
The MTA device never receives the TFTP configuration file.	The configuration file is not cached at the DPE.	Wait until the next provisioning attempt, at which time the file should be cached. If this fails, reset the MTA.
	A conflicting TFTP server option is included in the network registrar MTA scope policy.	Since BAC inserts the DPE address for the TFTP server, you can safely remove this option from the policy.
The MTA device receives a configuration file but the DPE fails to receive the SNMP Inform (step 25 in Figure 5-1) as seen in the <i>dpe.log</i> file.	One of: <ul style="list-style-type: none"> An internal conflict in the configuration file. A conflict with Realm origin of the telephony certificate chain. A conflict with the Realm Name provided in Option 122. 	Ensure that the MTA configuration file is consistent.
	The MTA device reports success (step 25 in Figure 5-1) although an RSIP is not sent.	The MTA cannot resolve the IP address of the CMS FQDN given in the MTA configuration file.
The MTA cannot reach the IP address(es) of the CMS. This is an indication that no route is configured.		Resolve all routing problems.
The MTA device reports success (step 25 in Figure 5-1), although it proceeds to contact the KDC again for CMS service.	The MTA configuration file points to an incorrect cable modem.	Correct the configuration file, or reconfigure the Cisco BTS 10200 to use the FQDN listed in the configuration file.
	The MTA configuration file has its <code>pkcMtaDevCmsIPsecCtrl</code> value missing, or it is set to 1. This means it will perform secure NCS call signaling, or it will use an ASCII suffix that does not match that of the CMS FQDN.	Correct the configuration file. If you intend to perform secure signaling, take the necessary steps to configure the KDC and the BTS for support.
The MTA device reports success (step 25 in Figure 5-1), RSIPs, but gets no response or gets an error in response from the soft switch.	The MTA is unprovisioned or has been incorrectly provisioned on the Cisco BTS 10200.	Provision MTA on the Cisco BTS 10200.
	An eMTA DNS entry does not exist.	Place an entry in the correct DNS zone for the eMTA. Dynamic DNS is the preferred method. Refer to Cisco Network Registrar documentation for information on enabling DDNS.

Certificate Trust Hierarchy

There are two certificate hierarchies affiliated with BAC PacketCable, the MTA Device Certificate Hierarchy and the CableLabs Service Provider Certificate Hierarchy, as shown in [Figure 6-1](#).

Figure 6-1 PacketCable Certificate Hierarchy



Before implementing PacketCable in BAC, you should thoroughly familiarize yourself with these technology documents:

- *RFC 2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile*
- *DOCSIS Baseline Privacy Plus Interface Specification, SP-BPI+-111-040407, April 7, 2004*



Note

While Euro PacketCable uses the security specifications from PacketCable [PKT-SP-SEC-I08-030415], some changes are needed in relation to the digital certificates that are used in a Euro PacketCable environment. To keep Euro PacketCable and PacketCable as alike as possible, Euro PacketCable uses all PacketCable security technology, including new revision of the security specifications [PKTSP-SEC-I08-030415].

The elements of the Euro PacketCable certificates that are different from the PacketCable certificates are indicated in the tables below.

For Euro PacketCable, the Euro PacketCable certificates are the only valid certificates, any requirements that are stated in [PKT-SP-SEC-I08-030415] for PacketCable which refer to PacketCable Certificates are changed to the corresponding requirements for the Euro PacketCable certificates.

Euro PacketCable-compliant eMTAs must have the Euro-DOCSIS root CVC CA's public key stored in the CM's nonvolatile memory instead of the DOCSIS CVC CA's public key. Euro PacketCable-compliant standalone MTAs must have the tComLabs CVC Root Certificate and the tComLabs CVC CA certificate stored in nonvolatile memory. The CVC of manufacturers are verified by checking the certificate chain.

Certificate Validation

PacketCable certificate validation in general involves validation of an entire chain of certificates. For example, when the Provisioning Server validates an MTA Device certificate, the following chain of certificates is validated:

MTA Root Certificate + MTA Manufacturer Certificate + MTA Device Certificate

The signature on the MTA Manufacturer Certificate is verified with the MTA Root Certificate and the signature on the MTA Device Certificate is verified with the MTA Manufacturer Certificate. The MTA Root Certificate is self-signed and is known in advance to the Provisioning Server. The public key present in the MTA Root Certificate is used to validate the signature on this same certificate.

Usually the first certificate in the chain is not explicitly included in the certificate chain that is sent over the wire. In the cases where the first certificate is explicitly included it must already be known to the verifying party ahead of time and must *not* contain any changes to the certificate with the possible exception of the certificate serial number, validity period, and the value of the signature. If changes other than these exist in the CableLabs Service Provider Root certificate that was passed over the wire in comparison to the known CableLabs Service Provider Root certificate, the device making the comparison must fail the certificate verification.

The exact rules for certificate chain validation must fully comply with RFC 2459, where they are referred to as Certificate Path Validation. In general, X.509 certificates support a liberal set of rules for determining if the issuer name of a certificate matches the subject name of another. The rules are such that two name fields may be declared to match even though a binary comparison of the two name fields does not indicate a match. RFC 2459 recommends that certificate authorities restrict the encoding of name fields so that an implementation can declare a match or mismatch using simple binary comparison.

PacketCable security follows this recommendation. Accordingly, the DER-encoded `tbsCertificate.issuer` field of a PacketCable certificate must be an exact match to the DER-encoded `tbsCertificate.subject` field of its issuer certificate. An implementation may compare an issuer name to a subject name by performing a binary comparison of the DER-encoded `tbsCertificate.issuer` and `tbsCertificate.subject` fields.

The sections below specify the required certificate chain, which must be used to verify each certificate that appears at the leaf node (at the bottom) in the PacketCable certificate trust hierarchy illustrated in [Figure 6-1](#).

Validity period nesting is not checked and intentionally not enforced. Thus, the validity period of a certificate need not fall within the validity period of the certificate that issued it.

MTA Device Certificate Hierarchy

The device certificate hierarchy exactly mirrors that of the DOCSIS1.1/BPI+ hierarchy. It is rooted at a CableLabs-issued PacketCable MTA Root certificate, which is used as the issuing certificate of a set of manufacturer certificates. The manufacturer certificates are used to sign the individual device certificates.

The information contained in the following tables contains the PacketCable-specific values for the required fields according to RFC 2459. These PacketCable specific values must be followed according to [Table 6-2](#), except that Validity Periods should be as given in the respective tables. If a required field is not specifically listed for PacketCable, then follow the guidelines in RFC 2459.

MTA Root Certificate

This certificate must be verified as part of a certificate chain containing the MTA Root Certificate, MTA Manufacturer Certificate, and the MTA Device Certificate.

Table 6-2 lists the values relevant to the MTA Root Certificate.

Table 6-2 MTA Root Certificate

MTA Root Certificate		
Subject Name Form	PacketCable	Euro PacketCable
	C=US	C=BE
	O=CableLabs	O=tComLabs
	OU=PacketCable	OU=Euro-PacketCable
	CN=PacketCable Root Device Certificate Authority	CN=Euro-PacketCable Root Device Certificate Authority
Intended Usage	This certificate is used to sign MTA Manufacturer Certificates and is used by the KDC. This certificate is not used by the MTAs and thus does not appear in the MTA MIB.	
Signed By	Self-signed	
Validity Period	20+ years. It is intended that the validity period is long enough that this certificate is never reissued.	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true, pathLenConstraint=1)	

MTA Manufacturer Certificate

This certificate must be verified as part of a certificate chain containing the MTA Root Certificate, MTA Manufacturer Certificate, and the MTA Device Certificate. The state/province, city and manufacturer's facility are optional attributes. A manufacturer may have more than one manufacturer's certificate, and there may exist one or more certificates per manufacturer. All certificates for the same manufacturer may be provided to each MTA either at manufacture time or during a field update. The MTA must select an appropriate certificate for its use by matching the issuer name in the MTA Device Certificate with the subject name in the MTA Manufacturer Certificate. If present, the authorityKeyIdentifier of the device certificate must match the subjectKeyIdentifier of the manufacturer certificate as described in RFC 2459. The CompanyName field that is present in O and CN may be different in the two instances.

Table 6-3 lists the values relevant to the MTA Manufacturer Certificate.

Table 6-3 MTA Manufacturer Certificates

MTA Manufacturer Certificate		
Subject Name Form	PacketCable C=US O=CableLabs OU=PacketCable CN=PacketCable Root Device Certificate Authority	Euro PacketCable C= <i>Country of Manufacturer</i> O= <i>Company Name</i> [stateOrProvinceName = <i>State/Province</i>] [localityName= <i>City</i>] OU=Euro-PacketCable [organizationalUnitName= <i>Manufacturing Location</i>] CN= <i>Company Name</i> Euro-PacketCable CA
Intended Usage	This certificate is issued to each MTA manufacturer and can be provided to each MTA as part of the secure code download as specified by the PacketCable Security Specification (either at manufacture time, or during a field update). This certificate appears as a read-only parameter in the MTA MIB. This certificate along with the MTA Device Certificate is used to authenticate the MTA device identity (MAC address) during authentication by the KDC.	
Signed By	MTA Root Certificate CA	
Validity Period	20 years	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign), subjectKeyIdentifier[n,m], authorityKeyIdentifier[n,m](keyIdentifier= <i>subjectKeyIdentifier value from CA certificate</i>), basicConstraints[c,m](cA=true, pathLenConstraint=0)	

MTA Device Certificate

This certificate must be verified as part of a certificate chain containing the MTA Root Certificate, MTA Manufacturer Certificate and the MTA Device Certificate. The state/province, city and manufacturer's facility are optional attributes. The MAC address must be expressed as six pairs of hexadecimal digits separated by colons, for example, "00:60:21:A5:0A:23". The alpha hexadecimal characters (A-F) must be expressed as uppercase letters. The MTA device certificate should not be replaced or renewed.

Table 6-4 lists the values relevant to the MTA Device Certificate.

Table 6-4 MTA Device Certificates

MTA Device Certificate		
Subject Name Form	PacketCable C=Country O=Company Name [ST=State/Province] [L=City], OU=PacketCable [OU=Product Name] [OU=Manufacturer's Facility] CN=MAC Address	Euro PacketCable C=Country of Manufacturer O=Company Name [ST=State/Province] [L=City] OU=Euro-PacketCable [OU=Product Name] [OU=Manufacturing Location] CN=MAC Address
Intended Usage	This certificate is issued by the MTA manufacturer and installed in the factory. The provisioning server cannot update this certificate. This certificate appears as a read-only parameter in the MTA MIB. This certificate is used to authenticate the MTA device identity (MAC address) during provisioning.	
Signed By	MTA Manufacturer Certificate CA	
Validity Period	At least 20 years	
Modulus Length	1024, 1536 or 2048	
Extensions	keyUsage[c,o](digitalSignature, keyEncipherment) authorityKeyIdentifier[n,m](keyIdentifier=subjectKeyIdentifier value from CA certificate)	

MTA Manufacturer Code Verification Certificates

Code Verification Certificate (CVC) specification for eMTAs must be identical to the DOCSIS 1.1 CVC, specified in DOCSIS specification SP-BPI+-I11-040407.

CableLabs Service Provider Certificate Hierarchy

The Service Provider Certificate Hierarchy is rooted at a CableLabs issued CableLabs Service Provider Root certificate. That certificate is used as the issuing certificate of a set of service provider's certificates. The service provider's certificates are used to sign an optional local system certificate. If the local system certificate exists then that is used to sign the ancillary equipment certificates, otherwise the ancillary certificates are signed by the Service Provider's CA.

The information contained in Table 6-5 contains the specific values for the required fields according to RFC 2459. These specific values must be followed. If a required field is not specifically listed then the guidelines in RFC 2459 must be followed exactly.

CableLabs Service Provider Root Certificate

Before any Kerberos key management can be performed, an MTA and a KDC need to perform mutual authentication using the PKINIT extension to the Kerberos protocol. An MTA authenticates a KDC after it receives a PKINIT Reply message containing a KDC certificate chain. In authenticating the KDC, the MTA verifies the KDC certificate chain, including KDC's Service Provider Certificate signed by the CableLabs Service Provider Root CA.

Table 6-5 lists the values relevant to the CableLabs Service Provider Root Certificate.

Table 6-5 CableLabs Service Provider Root Certificates

CableLabs Service Provider Root Certificate		
Subject Name Form	PacketCable C=US O=CableLabs CN=CableLabs Service Provider Root CA	Euro PacketCable C=BE O=tComLabs CN=tComLabs Service Provider Root CA
Intended Usage	This certificate is used to sign Service Provider CA certificates. This certificate is installed into each MTA at the time of manufacture or with a secure code download as specified by the PacketCable Security Specification and cannot be updated by the Provisioning Server. Neither this root certificate nor the corresponding public key appears in the MTA MIB.	
Signed By	Self-signed	
Validity Period	20+ years. It is intended that the validity period is long enough that this certificate is never reissued.	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign, cRLSign) subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true)	

Service Provider CA Certificate

This is the certificate held by the service provider, signed by the CableLabs Service Provider Root CA. It is verified as part of a certificate chain that includes the CableLabs Service Provider Root Certificate, Telephony Service Provider Certificate, optional Local System Certificate and an end-entity server certificate. The authenticating entities normally already possess the CableLabs Service Provider Root Certificate and it is not transmitted with the rest of the certificate chain.

The fact that a Service Provider CA Certificate is always explicitly included in the certificate chain allows a Service Provider the flexibility to change its certificate without requiring reconfiguration of each entity that validates this certificate chain (for example, MTA validating a PKINIT Reply). Each time the Service Provider CA Certificate changes, its signature must be verified with the CableLabs Service Provider Root Certificate. However, a new certificate for the same Service Provider must preserve the same value of the OrganizationName attribute in the SubjectName. The *Company* field that is present in O and CN may be different in the two instances.

Table 6-6 lists the values relevant to the CableLabs Service Provider CA Certificate.

Table 6-6 CableLabs Service Provider CA Certificates

CableLabs Service Provider Root Certificate		
Subject Name Form	PacketCable	Euro PacketCable
	C=Country O=Company CN=Company CableLabs Service Provider CA	C=Country O=Company CN=Company tComLabs Service Provider CA
Intended Usage	This certificate is used to sign Service Provider CA certificates. This certificate is installed into each MTA at the time of manufacture or with a secure code download as specified by the PacketCable Security Specification and cannot be updated by the Provisioning Server. Neither this root certificate nor the corresponding public key appears in the MTA MIB.	
Signed By	Self-signed	
Validity Period	20+ years. It is intended that the validity period is long enough that this certificate is never reissued.	
Modulus Length	2048	
Extensions	keyUsage[c,m](keyCertSign cRLSign), subjectKeyIdentifier[n,m] basicConstraints[c,m](cA=true)	

Local System CA Certificates

A Service Provider CA may delegate the issuance of certificates to a regional Certification Authority called Local System CA (with the corresponding Local System Certificate). Network servers are allowed to move freely between regional Certification Authorities of the same Service Provider. Therefore, the MTA MIB does not contain any information regarding a Local System Certificate (which might restrict an MTA to KDCs within a particular region).

Table 6-7 lists the values relevant to the Local System CA Certificate.

Table 6-7 Local System CA Certificates

Local System CA Certificate		
Subject Name Form	PacketCable	Euro PacketCable
	C=Country O=Company OU=Local System Name CN=Company CableLabs Local System CA	C=Country O=Company OU=Local System Name CN=Company tComLabs Local System CA
Intended Usage	A Service Provider CA may delegate the issuance of certificates to a regional Certification Authority called Local System CA (with the corresponding Local System Certificate). Network servers are allowed to move freely between regional Certification Authorities of the same Service Provider. Therefore, the MTA MIB does not contain any information regarding a Local System Certificate (which might restrict an MTA to KDCs within a particular region).	

Table 6-7 Local System CA Certificates (continued)

Local System CA Certificate	
Signed By	Service Provider CA Certificate
Validity Period	20 years.
Modulus Length	1024, 1536, 2048
Extensions	keyUsage[c,m](keyCertSign, cRLSign), subjectKeyIdentifier[n,m], authorityKeyIdentifier[n,m](keyIdentifier= <i>subjectKeyIdentifier value from CA certificate</i>), basicConstraints[c,m](cA=true, pathLenConstraint=0)

Operational Ancillary Certificates

All these are signed by either the Local System CA or by the Service Provider CA. Other ancillary certificates may be added to this standard at a later time.

KDC Certificate

This certificate must be verified as part of a certificate chain containing the CableLabs Service Provider Root Certificate, Service Provider CA Certificate and the Ancillary Device Certificates. The PKINIT specification requires the KDC certificate to include the subjectAltName v.3 certificate extension, the value of which must be the Kerberos principal name of the KDC.

[Table 6-8](#) lists the values relevant to the KDC Certificate.

Table 6-8 KDC Certificates

Key Distribution Center Certificate		
Subject Name Form	PacketCable C= <i>Country</i> O= <i>Company</i> , [OU= <i>Local System Name</i>] OU= CableLabs Key Distribution Center CN= <i>DNS Name</i>	Euro PacketCable C= <i>Country</i> O= <i>Company</i> [OU= <i>Local System Name</i>] OU=tComLabs Key Distribution Center CN= <i>DNS Name</i>
Intended Usage	To authenticate the identity of the KDC server to the MTA during PKINIT exchanges. This certificate is passed to the MTA inside the PKINIT replies and is therefore not included in the MTA MIB and cannot be updated or queried by the Provisioning Server.	
Signed By	Service Provider CA Certificate or Local System Certificate	
Validity Period	20 years	
Modulus Length	1024, 1536 or 2048	
Extensions	keyUsage[c,o](digitalSignature)authorityKeyIdentifier[n,m](keyIdentifier= <i>subjectKeyIdentifier value from CA certificate</i>)subjectAltName[n,m]	

Delivery Function (DF)

This certificate must be verified as part of a certificate chain containing the CableLabs Service Provider Root Certificate, Service Provider CA Certificate and the Ancillary Device Certificates. This certificate is used to sign phase 1 IKE intra-domain exchanges between DFs (which are used in Electronic Surveillance). Although Local System Name is optional, it is required when the Local System CA signs this certificate. The IP address must be specified in standard dotted-quad notation, for example, 245.120.75.22.

Table 6-9 lists the values relevant to the DF Certificate.

Table 6-9 DF Certificates

DF Certificate		
Subject Name Form	PacketCable	Euro PacketCable
	C=Country	C=Country
	O=Company	O=Company
	[OU=Local System Name]	[OU=Local System Name]
	OU=PacketCable Electronic Surveillance	OU=Euro-PacketCable Electronic Surveillance
	CN=IP address	CN=IP address
Intended Usage	To authenticate IKE key management, used to establish IPsec Security Associations between pairs of DFs. These Security Associations are used when a subject that is being legally wiretapped forwards the call and event messages containing call info have to be forwarded to a new wiretap server (DF).	
Signed By	Service Provider CA Certificate or Local System CA Certificate	
Validity Period	20 years	
Modulus Length	2048	
Extensions	keyUsage[c,o](digitalSignature) authorityKeyIdentifier[n,m](keyIdentifier=subjectKeyIdentifier value from CA certificate) subjectAltName[n,m] (dNSName=DNSName)	

PacketCable Server Certificates

These certificates must be verified as part of a certificate chain containing the CableLabs Service Provider Root Certificate, Service Provider Certificate, Local System Operator Certificate (if used) and the Ancillary Device Certificates. These certificates are used to identify various servers in the PacketCable system. For example, they may be used to sign phase 1 IKE exchanges or to authenticate a PKINIT exchange. Although the Local System Name is optional, it is REQUIRED when the Local System CA signs this certificate. 2IP address values must be specified in standard dotted decimal notation, for example, 245.120.75.22. DNS Name values must be specified as a fully qualified domain name (FQDN), for example, device.packetcable.com.

Table 6-10 lists the values relevant to the PacketCable Server Certificate.

Table 6-10 PacketCable Server Certificates

PacketCable Server Certificates		
Subject Name Form	<p>PacketCable</p> <p>C=<i>Country</i></p> <p>O=<i>Company</i></p> <p>OU=PacketCable</p> <p>OU=[<i>Local System Name</i>]</p> <p>OU=<i>Sub-System Name</i></p> <p>CN=<i>Server Identifier[:Element ID]</i></p> <p>The value of <i>Server Identifier</i> must be the server's FQDN or its IP address, optionally followed by a colon (:) and an Element ID with no white space before or after the colon.</p> <p><i>Element ID</i> is the identifier that appears in billing event messages. It must be included in the certificate of every server that is capable of generating event messages. This includes a CMS, CMTS and MGC. [8] defines the Element ID as an 5-octet right-justified space-padded ASCII-encoded numerical string. When converting the Element ID for use in a certificate, spaces must be converted to ASCII zeroes (0x48).</p> <p>For example, a CMTS with Element ID 311 and IP address 123.210.234.12 will have a common name "123.210.234.12: 00311".</p> <p>The value of <i>Sub-System Name</i> must be one of the following:</p> <ul style="list-style-type: none"> • For Border Proxy: bp • For Cable Modem Termination System: cmts • For Call Management Server: cms • For Media Gateway: mg•For Media Gateway Controller: mgc • For Media Player: mp • For Media Player Controller: mpc • For Provisioning Server: ps • For Record Keeping Server: rks • For Signaling Gateway: sg 	<p>Euro PacketCable</p> <p>C=<i>Country</i></p> <p>O=<i>Company</i></p> <p>OU=Euro-PacketCable</p> <p>[OU=<i>Local System Name</i>]</p> <p>OU=<i>Sub-system Name</i></p> <p>CN=<i>Server Identifier[:Element ID]</i></p> <p>Please refer to [PKT-SP-SEC-IO8-030415] for additional specifications on the commonName.</p>
Intended Usage	<p>These certificates are used to identify various servers in the PacketCable system. For example, they may be used to sign phase 1 IKE exchanges or to authenticate a device in a PKINIT exchange.</p>	

Table 6-10 PacketCable Server Certificates (continued)

PacketCable Server Certificates	
Signed By	Telephony Service Provider Certificate or Local System Certificate
Validity Period	Set by MSO policy
Modulus Length	2048
Extensions	keyUsage[c,o](digitalSignaturekeyEncipherment) authorityKeyIdentifier[n,m](keyIdentifier= <i>subjectKeyIdentifier value from CA cert</i>) subjectAltName[n,m](dNSName= <i>DNSName</i> iPAddress= <i>IP AddressName</i>) The keyUsage tag is optional. When it is used it must be marked as critical. Unless otherwise described below, the subjectAltName extension must include the corresponding name value as specified in the CN field of the subject.

The CN attribute value for CMS certificates must be the Element ID. The subjectAltName extension must include either the IP Address or the FDQN of the CMS. The CN attribute value for CMTS certificates must be the Element ID. The subjectAltName extension must include either the IP Address or the FDQN of the CMTS.

The CN attribute value for MGC certificates must be the Element ID. The subjectAltName extension must include either the IP Address or the FDQN of the MGC.

Certificate Revocation

Out of scope for PacketCable at this time.

Code Verification Certificate Hierarchy

The CableLabs Code Verification Certificate (CVC) PKI is generic in nature and applicable to all CableLabs projects needing CVCs. This means the basic infrastructure can be re-used for every CableLabs project. There may be differences in the end-entity certificates required for each project, but in the cases where end-entity certificates overlap, one end-entity certificate could be used to support the overlap.

The CableLabs CVC hierarchy does not apply to eMTAs. Refer to section 11 for more information.

Common CVC Requirements

The following requirements apply to all Code Verification Certificates:

- Certificates must be DER encoded.
- Certificates must be version 3.
- Certificates must include the extensions that are specified in the following tables and must *not* include any additional extensions.
- The public exponent must be F4 (65537 decimal).

CableLabs Code Verification Root CA Certificate

This certificate must be validated as part of the certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Code Verification CA, and the Code Verification Certificates. See [Certificate Validation, page 6-10](#), for additional information on how to validate certificates.

[Table 6-11](#) lists the values relevant to the CableLabs Code Verification Root CA Certificate.

Table 6-11 CableLabs Code Verification Root CA Certificates

CableLabs Code Verification Root CA Certificate		
Subject Name Form	PacketCable C=US O=CableLabs CN=CableLabs CVC Root CA	Euro PacketCable C = BE O = tComLabs CN = tComLabs CVC Root CA
Intended Usage	This certificate is used to sign Code Verification CA Certificates. This certificate must be included in the S-MTAs nonvolatile memory at manufacture time.	
Signed By	Self-signed	
Validity Period	20+ years	
Modulus Length	2048	
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectkeyidentifier [n,m] basicConstraints [c,m](cA=true)	

CableLabs Code Verification CA Certificate

The CableLabs Code Verification CA Certificate must be validated as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, CableLabs Code Verification CA Certificate and the Code Verification Certificate. See [Certificate Validation, page 6-10](#), for additional information on how to validate certificates. There may be more than one CableLabs Code Verification CA. A S-MTA must support one CableLabs CVC CA at a time.

[Table 6-12](#) lists the values relevant to the CableLabs Code Verification CA Certificate.

Table 6-12 CableLabs Code Verification CA Certificates

CableLabs Code Verification CA Certificate		
Subject Name Form	PacketCable C=US O=CableLabs CN=CableLabs CVC CA	Euro PacketCable C = BE O = tComLabs CN = tComLabs CVC CA
Intended Usage	This certificate is issued to CableLabs by the CableLabs Code Verification Root CA. This certificate issues Code Verification Certificates. This certificate must be included in the S-MTAs nonvolatile memory at manufacture time.	
Signed By	CableLabs Code Verification Root CA	
Validity Period	Set by CableLabs policy	

Table 6-12 CableLabs Code Verification CA Certificates (continued)

CableLabs Code Verification CA Certificate	
Modulus Length	2048
Extensions	KeyUsage [c,m] (keyCertSign, cRL Sign) subjectKeyIdentifier [n,m] authorityKeyIdentifier [n,m] basicConstraints [c,m](cA=true, pathLenConstraint=0)

Manufacturer Code Verification Certificate

The CableLabs Code Verification CA issues this certificate to each authorized Manufacturer. It is used in the policy set by the cable operator for secure software download.

[Table 6-13](#) lists the values relevant to the Manufacturer Code Verification Certificate.

Table 6-13 Manufacturer Code Verification Certificates

Manufacturer Code Verification Certificate		
Subject Name Form	PacketCable C=Country O=Company Name [ST=State/Province] [L=City] CN=Company Name Mfg CVC	Euro PacketCable C=Country O=Company Name [ST=state/province] [L=City] CN=Company Name Mfg CVC
Intended Usage	The CableLabs Code Verification CA issues this certificate to each authorized Manufacturer. It is used in the policy set by the cable operator for secure software download.	
Signed By	CableLabs Code Verification CA	tComLabs Code Verification CA Certificate
Validity Period	Set by CableLabs policy	
Modulus Length	1024, 1536, 2048	
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m]	

The Company Name in the Organization may be different than the Company Name in the Common Name.

Service Provider Code Verification Certificate

The Service Provider Code Verification Certificate must be validated as part of a certificate chain containing the CableLabs Code Verification Root CA Certificate, the CableLabs Code Verification CA Certificate, and the Service Provider Code Verification Certificate. Refer to [Certificate Validation, page 6-10](#), for additional information on how to validate certificates.

Table 6-14 lists the values relevant to the Service Provider Code Verification Certificate.

Table 6-14 Service Provider Code Verification Certificates

Service Provider Code Verification Certificate		
Subject Name Form	C= <i>Country</i> O= <i>Company Name</i> [ST= <i>State/Province</i>] [L= <i>City</i>] CN= <i>Company Name</i> Service Provider CVC	C= <i>Country</i> O= <i>Company Name</i> [ST= <i>State/Province</i>] [L= <i>City</i>] CN= <i>Company Name</i> Service Provider CVC
Intended Usage	The CableLabs Code Verification CA issues this certificate to each authorized Service Provider. It is used in the policy set by the cable operator for secure software download.	
Signed By	CableLabs Code Verification CA	tComLabs Code Verification CA Certificate
Validity Period	Set by CableLabs policy	
Modulus Length	1024, 1536, 2048	
Extensions	extendedKeyUsage [c,m] (id-kp-codeSigning) authorityKeyIdentifier [n,m]	

The Company Name in the Organization may be different than the Company Name in the Common Name.

Certificate Revocation Lists for CVCs

The S-MTA is not required to support Certificate Revocation Lists (CRLs) for CVCs.



CHAPTER 7

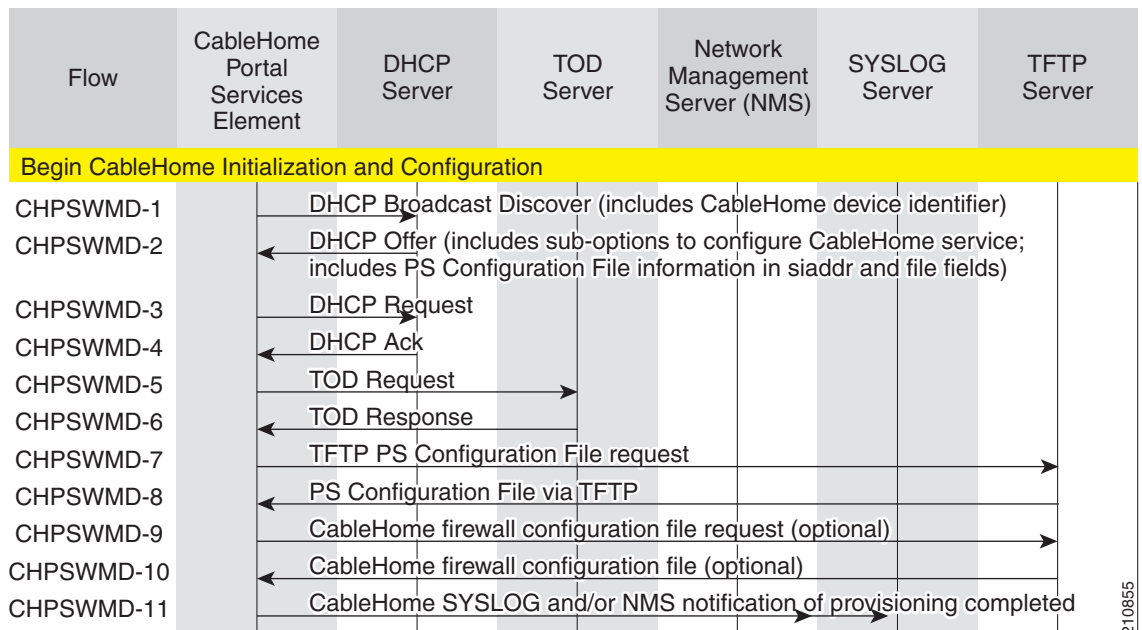
CableHome Configuration

This chapter describes the activities that must be performed to ensure a satisfactory CableHome deployment. There are two versions of the CableHome technology: secure (SNMP) and non-secure (DHCP). This chapter deals exclusively with the non-secure version.

Non-Secure CableHome Provisioning Flow

It is extremely useful to identify which step in the non-Secure CableHome provisioning flow is failing before attempting to diagnose other details. [Figure 7-1](#) provides a summary of the key provisioning flows.

Figure 7-1 Non-Secure CableHome Flow



210855

Table 7-1 describes the provisioning flow in a non-secure CableHome deployment.

Table 7-1 Non-Secure CableHome Provisioning Workflow

Step	Workflow	Description
CHPSWMD-1	DHCP Discover	The WAN-MAN obtains its IP lease.
CHPSWMD-2	DHCP Offer	The provisioning system returns a DHCP Offer with CableHome Option 177 suboptions: <ul style="list-style-type: none"> • 3—Specifies the SNMP Entity Address of the service provider. • 6—Specifies the Kerberos realm name of the provisioning realm. The realm name is required by portal services to permit a DNS lookup for the address of the Key Distribution Center. • 51—Specifies the Kerberos Server IP address, which informs the portal service of the network address of one or more Key Distribution Center servers. This Offer also contains the file information, in the file and siaddr fields, that is required to configure the portal service.
CHPSWMD-3	DHCP Request	The portal service sends the appropriate DHCP server a DHCP Request message to accept the DHCP Offer.
CHPSWMD-4	DHCP Ack	The DHCP server returns a DHCP Ack, which contains the IPv4 address of the portal service. Based on the information received in the DHCP Ack, the portal service modifies the <code>cabhPsDevProvMode</code> parameter, which specifies provisioning in the DHCP (non-secure) mode. Also, the Time of Day server address is stored in the <code>cabhPsDevTimeServerAddr</code> parameter.
CHPSWMD-5	ToD Request	The portal service initiates Time of Day synchronization with the time servers identified in Option 4 of the DHCP Ack message.
CHPSWMD-6	ToD Response	The time of day servers respond with the current time in UTC format.
CHPSWMD-7	PS Configuration File Via TFTP	The portal service sends a TFTP Get Request to obtain a configuration file.
CHPSWMD-8	CableHome Firewall Configuration File Request	The configuration file is downloaded via TFTP. Optionally, if there is a firewall configuration to be loaded and this is the method selected to specify it, the IP address of the name and the hash of the firewall configuration file are included in the configuration file.

Table 7-1 Non-Secure CableHome Provisioning Workflow (continued)

Step	Workflow	Description
CHPSWMD-9	CableHome Firewall Configuration File Request	If the configuration file acquired in step CHPSWMD-8 contains firewall information, portal services may also acquire a firewall configuration file via a TFTP Get Request to the Firewall Configuration TFTP Server. If there is no firewall configuration information in the configuration file, the provisioning process skips steps CHPSWMD-9 and CHPSMWD-10.
CHPSWMD-10	CableHome Firewall Configuration File	The Firewall Configuration TFTP Server sends a TFTP Response containing the firewall configuration file.
CHPSWMD-11	CableHome SYSLOG and/or NMS notification of provisioning completed	Once successfully configured, the portal service sends a syslog message, an SNMP trap, or both, to inform BAC that it has been successfully configured.

Configuring CableHome

This section describes how to configure Network Registrar, the CMTS.

Configuring Network Registrar

-
- Step 1** Create selection tags for provisioned and unprovisioned WAN-MAN and also for provisioned WAN-Data.
 - Step 2** Configure unprovisioned and provisioned client classes and scopes for cable modems, as specified in *Cisco Network Registrar User's Guide*, 6.2.1.
 - Step 3** Configure unprovisioned and provisioned client classes and scopes for WAN-MAN.
 - Step 4** Configure provisioned client classes and scopes for WAN-Data.
 - Step 5** Add routes to all the subnets.
-

Configuring the RDU

To configure CableHome support on the RDU, perform these configurations:

- [Configuring CableHome WAN-MAN, page 7-4](#)
- [Configuring CableHome WAN-Data, page 7-4](#)

Configuring CableHome WAN-MAN

-
- Step 1** Create a DHCP Criteria for the provisioned WAN-MAN. To do this, set the client class to a client-class name that is configured in Network Registrar CableHome WAN-MAN.
- Step 2** Create a Class of Service for the provisioned WAN-MAN.
- Set the /cos/chWanMan/file to a CableHome configuration file appropriate for the Class of Service.
 - Set the /chWanMan/firewall/file to the desired firewall configuration file.
-

Configuring CableHome WAN-Data

Configure these WAN-Data parameters whenever you want portal services to obtain the WAN-Data IP addresses:

-
- Step 1** Create DHCP Criteria for WAN-Data.
- Step 2** Create Class of Service for WAN-Data.
-

Configuring the DPE

To configure the DPE to support the CableHome technology:

-
- Step 1** Open the CableHome device provisioning WAN-MAN config file and verify that DHCP Option 60 is set to either CableHome1.0 or CableHome1.1. Some manufacturers use a proprietary MIB object to instruct a device to behave as a pure cable modem, a non-CableHome router, or a CableHome router. The device appears as a Computer whenever the device DHCP packet does not contain CableHome1.0 or CableHome1.1 in the DHCP Option 60.
- Step 2** If you want the portal services to obtain IP addresses for WAN-Data:
- Ensure that the WAN-MAN configuration file contains TLV 28 that sets cabhCdpWanDataIpAddrCount to a value that is greater than 0.
 - In the cable modem configuration file, set the maximum number of CPE to include the number of WAN-Data IP addresses.
- Step 3** To enable self-provisioning when the CableHome device boots:
- In the unprov-wan-man.cfg portal services configuration file, set the portal services in the passthrough mode.
 - In the cable modem configuration file, set the maximum number of CPE to at least 2 to allow provisioning of the WAN-MAN and a computer. The computer can directly access sign-up web pages to be self-provisioned.
-



CHAPTER 8

Configuration Templates Management

This chapter details the templates that Broadband Access Center (BAC) supports for device configuration and device management. This chapter features:

- [Developing Template Files, page 8-1](#)
 - [Template Grammar, page 8-2](#)
 - [SNMP VarBind, page 8-5](#)
 - [Macro Variables, page 8-6](#)
 - [Adding SNMP TLVs, page 8-8](#)
 - [Encoding Types for Defined Options, page 8-12](#)
 - [DOCSIS Option Support, page 8-15](#)
 - [PacketCable Option Support, page 8-25](#)
 - [Non-Secure CableHome Option Support, page 8-26](#)
- [Using the Configuration File Utility, page 8-27](#)

Developing Template Files

BAC uses templates to help administrators deploy dynamic PacketCable, DOCSIS, and CableHome files. Using templates, you can create a template file in an easily readable format, and edit it quickly and simply. A template is an ASCII text file that represents the PacketCable, DOCSIS, or CableHome options and values used for generating a valid PacketCable, DOCSIS, or CableHome file. BAC uses the .tmpl file extension to identify template files.

You must add template files to the RDU as an external file using either the administrator user interface or the API, before any Class of Service can reference it.

When installing the BAC RDU component, several sample template files are copied to the *BAC_home/rdu/samples* directory.

Although all that you need to create or edit a template is a simple text editor, before attempting to create your own template file, you should thoroughly familiarize yourself with this information:

- BAC provisioning flows
- DOCSIS 1.0, 1.1, and 2.0 RFI specifications
- PacketCable 1.0, 1.1, and 1.5 specifications
- Media Terminal Adapter (MTA) device provisioning specification

- CableHome 1.0 specification
- SNMP MIBs for cable devices (for example, DOCS-CABLE-DEVICE-MIB)

Template Grammar

A template comprises four types of statements:

- [Comments, page 8-2](#)
- [Includes, page 8-3](#)
- [Options, page 8-3](#)
- [Instance Modifier, page 8-4](#)

Comments allow you to document your templates. Includes allow you to create building block templates to be used in other templates. You use options to specify the PacketCable, DOCSIS, or CableHome type length value (TLV) in a descriptive manner. [Table 8-1](#) describes the available template grammar options.

Table 8-1 **Template Grammar**

Option	Description
<comment>	::= #[ascii-string]
<include>	::= include "<filename.tpl>"
<option-description>	::= option <option-num> [instance <instance-num>] <option-value>
<option-num>	::= <unsigned-byte>[.<unsigned-byte>]*
<option-value>	::= <well-defined-value> <custom-value>
<well-defined-value>	::= <option-value-string>[,<option-value-string>]*
<custom-value>	::= <ascii-value> <hex-value> <ip-value> <snmp-value>
<ascii-value>	::= ascii <ascii-string>
<hex-value>	::= hex <hex-string>
<ip-value>	::= ip <ip-string>
<instance-num>	::= <unsigned integer>
<template>	::= <template-statement>*
<template-statement>	::= <comment> <include> <option-description>
<snmp-value>	::= <snmpvar-oid>,<snmpvar-type>,<snmpvar-value>

Comments

Comments provide information only and are always located between the pound (#) symbol and the end of a line. [Example 8-1](#) shows example comment usage.

Example 8-1 **Example Comment Usage**

```
#
# Template for gold service
#
option 3 1 # enabling network access
```


Includes

Include files let you build a hierarchy of similar, but slightly different, templates. This is very useful for defining options that are common across many service classes without having to duplicate the options in several templates.

You can use multiple include statements in a single template although the location of the include statement in the template is significant; the contents of the include file are included wherever the include statement is found in the template. The included template must be added as an external file to the RDU before it can be used. The included file must not contain any location modifiers such as `../..` because the templates are stored without path information in the RDU database. Examples 8-2 and 8-3 illustrate both correct and incorrect usage of the include option.

Example 8-2 Correct Include Statement Usage

```
# Valid, including common options
include "common_options.tpl"
```

Example 8-3 Incorrect Include Statement Usage

```
# Invalid, using location modifier
include "../common_options.tpl"

# Invalid, using incorrect file suffix
include "common_options.common"

# Invalid, not using double quotes
include common_options.tpl
```

Options

PacketCable, DOCSIS, and CableHome configuration files consist of properly encoded option id-value pairs. Two forms of options are supported: defined and custom.

- Well-defined options require the option number and value. The value is encoded based on the encoding type of the option number.
- Custom options require the option number, explicit value encoding type, and the value.

When using compound options, for example, Option 43, you can use the instance modifier to specify the TLV groupings. See [Instance Modifier, page 8-4](#), for additional information.

When specifying one of these well-defined options in a template, it is not necessary to specify a value encoding for the value. See [Encoding Types for Defined Options, page 8-12](#), and [DOCSIS Option Support, page 8-15](#), for additional information on these defined encoding types.

When specifying custom options (for example, Option 43), you must specify the encoding type for the option. The available encoding types are:

- ASCII— ASCII type encodes any given value as an ASCII string without a `NULL` terminator. If the value contains spaces, they must be double quoted.
- hex—The value must be valid hexadecimal and there must be exactly 2 characters for each octet. If `01` is specified as the value, then exactly one octet is used in the encoding. If `0001` is specified as the value, then exactly two octets are used in the encoding process.
- IP address—IP address type encodes any given value as 4 octets. For example, the IP address `10.10.10.1` is encoded as `0A0A0A01`.
- SNMPVarBind—An SNMP OID string, type, and value. Each of these is comma separated.

Use a comma to separate multi-valued options on a given line. Each value is treated as such, so you might have to double quote one of the values, but not the others. A good example of a multi-valued option is Option 11 (SNMP VarBind). See [SNMP VarBind, page 8-5](#), for additional information.

When specifying compound options, there is no need to specify the top level option (for example Option 4 when specifying Option 4.1). Examples [8-4](#) and [8-5](#) illustrate both correct and incorrect usage of the option statement.

Example 8-4 Correct Option Statement Usage

```
# Valid, specifying the number for well known option 3
option 3 1

# Valid, specifying the number for option 4 sub-option 1
option 4.1 1

# Valid, specifying a vendor option as hex
option 43.200 hex 00000C

# Valid, specifying a vendor option as ascii
option 43.201 ascii "enable log"

# Valid, specifying a vendor option as IP
option 43.202 ip 10.4.2.1
```

Example 8-5 Incorrect Option Statement Usage

```
# Invalid, using hex with incorrect hex separator
option 43.200 hex 00.00.0C

# Invalid, not using double quotes when needed
option 43.201 ascii enable log

# Invalid, not specifying IP address correctly
option 43.202 ip 10-10-10-1

# Invalid, specifying the description for option "Network Access Control"
option "Network Access Control" 1

# Invalid, specifying top level option
option 4
```

Instance Modifier

The instance modifier is used to group compound options into specific individual Tag-Length-Values (TLVs). Examples [8-6](#) and [8-7](#) illustrate both correct and incorrect methods of creating separate TLVs. These are required to enable the IOS DOCSIS modem to interpret the IOS commands as two separate commands.

Example 8-6 Correct IOS Command Line Entries

```
# Valid, each IOS command gets its own TLV
option 43.8 instance 1 00-00-0C
option 43.131 instance 1 ascii "login"
option 43.8 instance 2 00-00-0C
option 43.131 instance 2 ascii "password cable"
```

Example 8-7 Incorrect IOS Command Line Entries

```
# Invalid, IOS commands are grouped into one TLV
option 43.8 00-00-0C
option 43.131 ascii "login"
option 43.131 ascii "password cable"

# Invalid, using instance on non-compound options
option 3 instance 1 1
```



Note The encoding type for Option 43.8 is an organizationally unique identifier (OUI). Unlike that shown in [Example 8-4](#), this type only accepts an 00-00-0C format.

SNMP VarBind

You must use an object identifier (OID) when specifying DOCSIS Option 11, PacketCable Option 64, or CableHome Option 28. The MIB that contains the OID must be in one of the following MIBs loaded by the RDU. You must specify as much of the OID as needed to uniquely identify it. You can use the name or the number of the OID. The RDU automatically loads these MIBs:

- SNMPv2-SMI
- SNMPv2-TC
- CISCO-SMI
- CISCO-TC
- SNMPv2-MIB
- RFC1213-MIB
- IANAifType-MIB
- IF-MIB

DOCSIS MIBs

These DOCSIS MIBs are loaded into the RDU:

- DOCS-IF-MIB
- DOCS-BPI-MIB
- CISCO-CABLE-SPECTRUM-MIB
- CISCO-DOCS-EXT-MIB
- SNMP-FRAMEWORK-MIB
- DOCS-CABLE-DEVICE-MIB
- DOCS-CABLE-DEVICE-MIB-OBSOLETE
- CISCO-CABLE-MODEM-MIB

Two versions of the DOCS-CABLE-DEVICE MIB are loaded into the RDU:

- DOCS-CABLE-DEVICE-MIB-OBSOLETE (experimental branch)
- DOCS-CABLE-DEVICE-MIB (mib2 branch)

A fully-qualified MIB OID (.experimental...) always uniquely identifies a MIB OID.

If you use a nonfully qualified MIB OID from DOCS-CABLE-DEVICE-MIB, it will always default to DOCS-CABLE-DEVICE-MIB and not DOCS-CABLE-DEVICE-MIB-OBSOLETE.

Examples 8-8 and 8-9 illustrate using a fully-qualified MIB OID and a nonfully qualified MIB OID.

Example 8-8 Fully Qualified MIB OID

```
# Valid, uniquely identifying an OID
option 11 .experimental.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNmAccess
Entry.docsDevNmAccessStatus.1, Integer, 4
```

Example 8-9 NonFully Qualified MIB OID (Defaults to DOCS-CABLE-DEVICE-MIB)

```
# Valid, Non-Fully Qualified MIB OID.
option 11 .docsDevNmAccessStatus.1, Integer, 4
```

If no DOCSIS CMs in a deployment require DOCS-CABLE-DEVICE-MIB-OBSOLETE, you can always use the shorter form of the MIB OID.

PacketCable MIBs

These PacketCable (North American) MIBs are loaded into the RDU:

- CLAB-DEF-MIB
- PKTC-MTA-MIB
- PKTC-SIG-MIB
- PKTC-EVENT-MIB

CableHome MIBs

These CableHome MIBs are loaded into the RDU:

- CABH-CAP-MIB
- CABH-CDP-MIB
- CABH-CTP-MIB
- CABH-PS-DEV-MIB
- CABH-QOS-MIB
- CABH-SEC-MIB

These additional MIBs are needed but are not part of the BAC product:

- CABH-CTP-MIB needs RMON2-MIB, TOKEN-RING-RMON-MIB
- CABH-SEC-MIB needs DOCS-BPI2-MIB.

Macro Variables

Macro variables are specified as values in templates that let you specify device-specific option values. When a macro variable is encountered in the template, the properties hierarchy is searched for the macro variable name and the value of the variable is then substituted. The variable name is a custom property, which is predefined in the RDU. It must not contain any spaces.

After the custom property is defined, it can be used in this property hierarchy:

- System defaults
- Technology defaults, such as PacketCable, DOCSIS, or CableHome
- DHCP criteria properties
- Class of Service properties
- Device properties

The template parser works bottom up when locating properties in the hierarchy (device first, then the Class of Service, and so on) and converts the template option syntax. The following syntax is supported for macro variables:

- `${var-name}`—This syntax is a straight substitution. If the variable is not found, the parser will generate an error.
- `${var-name, ignore}`—This syntax lets the template parser ignore this option if the variable value is not found in the properties hierarchy.
- `${var-name, default-value}`—This syntax provides a default value if the variable is not found in the properties hierarchy.

Examples 8-10 and 8-11 illustrate both correct and incorrect usage of Option 11.

Example 8-10 Correct Macro Variables Usage

```
# Valid, using macro variable for max CPE's, straight substitution
option 18 ${MAX_CPES}

# Valid, using macro variable for max CPE's, ignore option if variable not found
# option 18 will not be defined in the DOCSIS configuration file if MAX_CPES
# is not found in the properties hierarchy
option 18 ${MAX_CPES, ignore}

# Valid, using macro variable for max CPE's with a default value
option 18 ${MAX_CPES, 1}

# Valid, using macro variable for vendor option
option 43.200 hex ${MACRO_VAR_HEX}

# Valid, using macro variable for vendor option
option 43.201 ascii ${MACRO_VAR_ASCII}

# Valid, using macro variable for vendor option
option 43.202 ip ${MACRO_VAR_IP}

# Valid, using macro variable in double quotes
option 18 "${MAX_CPES}"

# Valid, using macro variable within a value
option 43.131 ascii "hostname ${HOSTNAME}"

# Valid, using macro variables in multi-valued options
option 11 ${ACCESS_CONTROL_MIB,
.mib-2.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNmAccessEntry.docsDevNmAccess
Control.1}, Integer, ${ACCESS_CONTROL_VAL, 3}

# Valid, using macro variable in an include statement
include "${EXTRA_TEMPLATE}"

# Valid, using macro variable in an include statement with a default value
```

```
include "${EXTRA_TEMPLATE, modem_reset.tpl}"

# Valid, using macro variable in an include statement with a default value
include "${EXTRA_TEMPLATE, modem_reset}.tpl"

# Valid, using macro variable in an include statement with an ignore clause
include "${MY_TEMPLATE, ignore}"
```

Example 8-11 Incorrect Macro Variables Usage

```
# Invalid, using macro variable as the option number
option ${MAX_CPES} 1

# Invalid, using macro variable with space in name
option 18 ${MAX CPES}
```

Adding SNMP TLVs

BAC supports SNMP TLVs in dynamic template files, using Option 11 and 64, for:

- DOCSIS—From Broadband Access Center for Cable (BACC) version 2.0 onwards.
- PacketCable—From BACC version 2.5 onwards.
- CableHome—From BACC version 2.6 onwards.

To validate the syntax of the SNMP TLVs in these template files, BAC requires a MIB file containing the corresponding SNMP OID that is referenced in the SNMP TLV. If a template contains an SNMP TLV with an SNMP OID that cannot be found in a MIB, the SNMP TLV generates a syntax error.

The following sections describe how you can add SNMP TLVs without a MIB or with a vendor-specific MIB.

Adding SNMP TLVs Without a MIB

You can add SNMP TLVs in dynamic configuration files (DOCSIS, PacketCable, CableHome) without requiring the MIB be loaded by the RDU. From within RDU configuration extensions, the functionality can be accessed with the DOCSISOOptionFactory interface, using the following method:

```
public OptionValue createOptionValue(OptionSyntax syntax, String optionNumStr, String[]
optionValueList)
```

The public OptionSyntax.SNMP enumerated value can be used in the above method, in conjunction with the optionValueList containing the tuple: OID, Type, Value.

From RDU dynamic configuration templates, the following syntax is used to specify SNMP TLVs that are not validated against the RDU MIBs:

```
option option-number snmp OID, Type, Value
```

Examples:

```
# DOCS-CABLE-DEVICE-MIB:
option 11 snmp .docsDevNmAccessIp.1, IPADDRESS, 192.168.1.1

# Arris vendor specific SNMP TLV (OID numbers only, mix names/numbers)
option 11 snmp .1.3.6.1.4.1.4115.1.3.1.1.2.3.2.0, INTEGER, 6
option 11 snmp .enterprises.4115.1.3.1.1.2.3.2.0, INTEGER, 6

# NOTE: trailing colon required for single octet
option 11 snmp .1.3.6.1.2.1.69.1.2.1.6.3, STRING, 'c0:'
```

The allowed SNMP variable type names are:

ietf standard SMI Data Type	SNMP API name
Integer32	INTEGER
Integer (Enumerated)	INTEGER
Unsigned32	UNSIGNED32
Gauge32	GAUGE
Counter32	COUNTER
Counter64	COUNTER64
Timeticks	TIMETICKS
OCTET STRING	STRING
OBJECT IDENTIFIER	OBJID
IpAddress	IPADDRESS
BITS	STRING

For example, to specify a SMI Integer32 type, the following types are accepted (regardless of case sensitivity): Integer32, INTEGER.

For OCTET STRING type, all of the following types are accepted: OCTET STRING, OCTETSTRING, or STRING.

The custom SNMP TLV template option can be used to specify any SNMP TLV, including those that are present in the RDU MIBs. The custom SNMP TLV error checking is less stringent, and does not detect incorrect scalar/columnar references (for example, .0 vs. .n in OID names).

Adding SNMP TLVs With Vendor-Specific MIBs

Adding a MIB to the RDU enables templates to use the human-readable SNMP OID while also permitting macro variables to be used with the SNMP TLV value.

BACC 2.6 or earlier

If you have the MIB corresponding to the SNMP OID that you want to use, you can add the MIB file to the BAC RDU. After you add the MIB, any SNMP TLV using an SNMP OID referenced in the new MIB is recognized.

To add a new MIB to the RDU:

-
- Step 1** Copy the new MIB file to the *BAC_home/rdu/mibs* directory.
- Step 2** Add the */docsis/mibs/custom/mibList* property, whose value contains a comma-separated list of MIB filenames, to the:
- rdu.properties file, which is used by the RDU and the administrator user interface. This file resides in the *BAC_home/rdu/conf* directory.
 - api.properties file, which the Configuration File Utility (runCfgUtil.sh tool) uses.

**Note**

The `api.properties` file is not created during the BAC installation process. You must manually create this file for initial use, in any text editor. Ensure that you locate this file in the `BAC_home/rdu/conf` directory.

The `api.properties` file contains a `/docsis/mibs/custom/mibList`, which is configured for a set of MIBs that you can use in Arris embedded MTAs (eMTAs).

Step 3 Restart the RDU and the administrator user interface via the BAC process watchdog, using the `/etc/init.d/bprAgent restart rdu` command.

The following example describes the addition of ARRIS MIBs for use in templates to configure ARRIS MTAs.

Assume that you want to use an Arris vendor-specific SNMP TLV:

```
option 11 .ppCfgMtaCountryTemplate.0, INTEGER, 9
```

and the following MIB files were made available:

- ARRIS-MIB
- ARRIS-CM-CAPABILITY-MIB
- ARRIS-CM-DEVICE-MIB
- ARRIS-MTA-DEVICE-MIB
- PACKETPORT-MIB

You must copy the MIB files to the `BAC_home/rdu/mibs` directory, and insert the following property in the `api.properties` and `rdu.properties` files:

```
/docsis/mibs/custom/mibList=ARRIS-MIB,ARRIS-CM-CAPABILITY-MIB,ARRIS-CM-DEVICE-MIB,ARRIS-MTA-DEVICE-MIB,PACKETPORT-MIB
```

BACC 2.7 or later**Note**

Note that the `/docsis/mibs/custom/mibList` property has been renamed `/snmp/mibs/mibList` from BACC version 2.7 onwards.

If you have the MIB corresponding to the SNMP OID that you want to use, you can add the MIB file to the BAC RDU. After you add the MIB, any SNMP TLV using an SNMP OID referenced in the new MIB is recognized.

To add a new MIB to the BAC RDU:

- Step 1** Launch the BAC administrator user interface.
- Step 2** On the navigation bar, click **Configuration > Defaults**.
- Step 3** On the Configure Defaults page that appears, click the System Defaults link on the left pane.
- Step 4** In the MIB List field, paste the content of the new MIB at the end.
- Step 5** Click **Submit**.

**Note**

In version 2.7 and later, the MIB parsing tool has been enhanced; subsequently, the tool sometimes returns errors on MIB versions that parsed without error previously. If you encounter any error that you are unable to resolve by editing the new MIB, contact the Cisco TAC.

Debugging the MIB Load Order

Typically, vendors provide several MIBs requiring a specific load order to satisfy inter-MIB dependencies. But because the vendor frequently does not provide the correct load order, you must determine the correct load order yourself. This section describes how you can use BAC debugging information to resolve MIB load-order issues.

**Note**

The MIB load order in BAC is set by the order in which the MIBs are listed in the:

- `/snmp/mibs/MibList` property, if you are using BACC 2.6.x releases.
- `/docsis/mibs/custom/mibList` property, if you are using BACC 2.7.x releases.

You can use the `runCfgUtil.sh` tool to determine the correct load order for the property specified in the `api.properties` file. The `runCfgUtil.sh` tool resides in the `BAC_home/rdu/bin` directory.

**Note**

This procedure references the `/snmp/mibs/MibList` property that BACC 2.7.x releases use. If you are running 2.6.x or earlier releases, ensure that you use the `/docsis/mibs/custom/mibList` property.

Step 1

Configure `runCfgUtil.sh` via the `api.properties` file using configuration content similar to that described in this step. The `api.properties` file enables BAC tracing to direct MIB debugging information to the user console.

```
#
# Enable logging to the console
#
/server/log/1/level=Info
/server/log/1/properties=level
/server/log/1/service=com.cisco.csrc.logging.SystemLogService
/server/log/1/name=Console
#
# Enable trace categories
#
/server/log/trace/rduserver/enable=enabled
#
# The list of MIBs to be added.
#
/snmp/mibs/MibList=arrishdr.mib, arris_cm_capability.mib, arris_mta_device.mib, arris_sip.mib,
 arris_cm.mib, pp.mib, blp2.mib, dev0.mib, docs_evt.mib, qos.mib, test.mib, usb.mib, snmpv2_conf
.mib, rfc1493.mib, rfc1907.mib, rfc2011.mib, rfc2013.mib, rfc2233.mib, rfc2571.mib, rfc2572.mib, r
fc2573.mib, rfc2574.mib, rfc2575.mib, rfc2576.mib, rfc2665.mib, rfc2669.mib, rfc2670.mib, rfc2786
.mib, rfc2851.mib, rfc2933.mib, rfc 3083.mib
```

Step 2

With `runCfgUtil.sh` so configured, run the tool to encode any template containing an Option 11 or Option 64 (SNMP encoding). The tool attempts to load the MIBs specified within `/snmp/mibs/MibList`, and directs the complete debugging information, along with any MIB load errors, to the user console.

- Step 3** Use the error information to massage the MIB order specified within `/snmp/mibs/MibList` until the complete set of MIBs loads without error and the file encode succeeds.
- Step 4** Once you determine a successful load order, complete the procedure based on the BACC version you are using:

BACC 2.7 or later

- a. From the administrator user interface, click **Configuration > Defaults**, then the System Defaults link.
- b. In the MIB List field, copy the load order information.

The RDU is now configured to encode templates using the vendor-supplied MIBs.



Note You do not need to restart the RDU.

Ensure that you use the `/snmp/mibs/mibList` string in the `api.properties` file and the MIB List field.

BACC 2.6 or later

- a. Copy the load order information to the `/snmp/mibs/MibList` property in the `rdu.properties` file. This file resides in the `BAC_home/rdu/conf` directory.
- b. Restart the RDU via the BAC process watchdog, using the `/etc/init.d/bprAgent restart rdu` command.

The RDU is now configured to encode templates using the vendor-supplied MIBs.

Encoding Types for Defined Options

Table 8-2 identifies the options with defined encoding types.

Table 8-2 Defined Option Encoding Types

Encoding	Input	Example
Boolean	0 for false and 1 for true.	0
Bytes	A series of hexadecimal octets. Each octet must be 2 characters.	000102030405060708
IP Address	Four unsigned integer 8, dot (.) separated.	10.10.10.1
Multiple IP Addresses	Comma-separated list of IP addresses.	10.11.12.13,10.11.12.14
MAC Address	Six hexadecimal octets, colon (:) or dash (-) separated. Each octet must be exactly 2 characters. Colons and dashes must not be mixed.	00:01:02:03:04:05 or 00-01-02-03-04-05

Table 8-2 Defined Option Encoding Types (continued)

Encoding	Input	Example
MAC Address And Mask	Twelve octets colon (:) or dash (-) separated. Each octet must be 2 characters. Colons and dashes must not be mixed. The first six octets represent the MAC address; the last six represent the mask for the MAC address.	00:01:02:03:04:05:06:07:08:09:0A:0B or 00-01-02-03-04-05-06-07-08-09-0A-0B
NVTASCII	An ASCII string. The encoded string will not be NULL terminated.	This is an ASCII string
OID	An SNMP OID string.	sysinfo.0
OIDCF	An SNMP OID string and an unsigned integer (0 or 1) comma separated.	sysinfo.0,1
OUI	Three hexadecimal octets colon (:) or dash (-) separated. Each octet must be 2 characters.	00-00-0C
SNMPVarBind	An SNMP OID string, type, and value. Each of these is comma separated. Valid types are: <ul style="list-style-type: none"> • BITS • Counter • Counter32 • Counter64 • Gauge • Gauge32 • INTEGER • Integer32 • IpAddress • OCTETSTRING • OBJECTIDENTIFIER • Opaque • TimeTicks • Unsigned32 <p>Note The OCTETSTRING can be a string that will be converted to a hexadecimal notation without a trailing NULL, octet string for example, or hexadecimal notation contained in single quotes, 'aa:bb:cc' for example.</p>	.experimental.docsDev.docsDevMI BObjects. docsDevNmAccessTable.docsDevNmA ccessEntry.docsDevNmAccessStatu s.1, INTEGER, 4
Sub Type	One or two comma separated unsigned integer 8.	12 or 12,14

Table 8-2 *Defined Option Encoding Types (continued)*

Encoding	Input	Example
Unsigned integer 8	0 to 255	14
Unsigned integer 16	0 to 65535	1244
Unsigned integer 32	0 to 4294967295	3455335
Unsigned integer 8 and unsigned integer 16	One unsigned integer 8 and one unsigned integer 16, comma separated.	3,12324
Unsigned integer 8 pair	Two unsigned integer 8, comma separated.	1,3
Unsigned integer 8 triplet	Three unsigned integer 8, comma separated.	1,2,3
ZTASCII	An ASCII string. The encoded string will be NULL terminated.	This is an ASCII string

BITS Value Syntax

When using the BITS type, you must specify either the labels (“interval1 interval2 interval3”) or numeric bit location (“0 1 2”). Note that label values are 1-based and bit values are 0-based.

This is the syntax that uses the bit numbers:

```
option 11 .pktcSigDevR0Cadence.0,STRING,"0 1 2 3 4 5 6 7 8 9 10 11 12 13 14"
```

This is the syntax for the customer octet string (FFFE000000000000) that uses the labels:

```
option 11 .pktcSigDevR0Cadence.0,STRING,"interval1 interval2 interval3
interval4 interval5 interval6 interval7 interval8 interval9 interval10
interval11 interval12 interval13 interval14 interval15"
```

OCTETSTRING Syntax

The OCTETSTRING can be either a string that is converted to hexadecimal notation without a trailing NULL (for example, octet string), or hexadecimal notation contained within single quotes, (for example, 'aa:bb:cc').

DOCSIS Option Support

Table 8-3 describes DOCSIS options and identifies the specific version support for each option.

Table 8-3 DOCSIS Options and Version Support

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
0	PAD	No length and no value	N/A	True	✓	✓	✓
1	Downstream Frequency	Unsigned integer 32	Multiples of 62500	False	✓	✓	✓
2	Upstream Channel ID	Unsigned integer 8	None	False	✓	✓	✓
3	Network Access Control	Boolean	None	False	✓	✓	✓
4	Class of Service	Compound	None	False	✓	✓	✓
4.1	Class ID	Unsigned integer 8	Between 1-16 inclusive	False	✓	✓	✓
4.2	Maximum Downstream Rate	Unsigned integer 32	None	False	✓	✓	✓
4.3	Maximum Upstream Rate	Unsigned integer 32	None	False	✓	✓	✓
4.4	Upstream Channel Priority	Unsigned integer 8	Less than 8	False	✓	✓	✓
4.5	Guaranteed Minimum Upstream Channel Data Rate	Unsigned integer 32	None	False	✓	✓	✓
4.6	Maximum Upstream Channel Transmit Burst	Unsigned integer 16	None	False	✓	✓	✓
4.7	Class-of-Service Privacy Enable	Boolean	None	False	✓	✓	✓
6	CM MIC Configuration Setting	Byte 16	None	False	✓	✓	✓
7	CMTS MIC Configuration Setting	Byte 16	None	False	✓	✓	✓
9	Software Upgrade Filename	NVTASCII	None	True	✓	✓	✓
10	SNMP Write-Access Control	OIDCF	None	True	✓	✓	✓
11	SNMP MIB Object	SNMPVarBind	None	True	✓	✓	✓
14	CPE Ethernet MAC Address	MAC Address	None	True	✓	✓	✓
15	Telephony Settings Option	NVTASCII	None	False	✓	✓	✓
15.2	Service Provider Name	NVTASCII	None	False	✓	✓	✓
15.3	Telephone Number (1)	NVTASCII	None	False	✓	✓	✓
15.4	Telephone Number (2)	NVTASCII	None	False	✓	✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
15.5	Telephone Number (3)	NVTASCII	None	False	✓	✓	✓
15.6	Connection Threshold	Unsigned integer 8	None	False	✓	✓	✓
15.7	Login Username	NVTASCII	None	False	✓	✓	✓
15.8	Login Password	NVTASCII	None	False	✓	✓	✓
15.9	DHCP Authentication	Boolean	None	False	✓	✓	✓
15.10	DHCP Server	IP Address	None	False	✓	✓	✓
15.11	RADIUS realm	NVTASCII	None	False	✓	✓	✓
15.12	PPPAAuthentication	Unsigned integer 8	None	False	✓	✓	✓
15.13	Demand Dial Inactivity Timer Threshold	Unsigned integer 8	None	False	✓	✓	✓
16	SNMP IP Address (No Longer Used)	IP Address	None	False	✓	✓	✓
17	Baseline Privacy Configuration Setting	Compound	None	False	✓	✓	✓
17.1	Authorize Wait Timeout	Unsigned integer 32	Between 1 and 30 inclusive	False	✓	✓	✓
17.2	Reauthorize Wait Timeout	Unsigned integer 32	Between 1 and 30 inclusive	False	✓	✓	✓
17.3	Authorization Grace Time	Unsigned integer 32	Between 1 and 1800 inclusive	False	✓		
17.3	Authorization Grace Time	Unsigned integer 32	Between 1 and 6047999 inclusive	False		✓	✓
17.4	Operational Wait Timeout	Unsigned integer 32	Between 1 and 10 inclusive	False	✓	✓	✓
17.5	Rekey Wait Timeout	Unsigned integer 32	Between 1 and 10 inclusive	False	✓	✓	✓
17.6	TEK Grace Time	Unsigned integer 32	Between 1 and 1800 inclusive	False	✓		
17.6	TEK Grace Time	Unsigned integer 32	Between 1 and 302399 inclusive	False		✓	✓
17.7	Authorize Reject Wait Timeout	Unsigned integer 32	Between 1 and 600 inclusive	False	✓	✓	✓
17.8	SA Map Wait Timeout	Unsigned integer 32	Between 1 and 18006 inclusive	False		✓	✓
17.9	Maximum Clock Drift	Unsigned integer 32	Between 1 and 10 inclusive	False		✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
18	Maximum Number of CPE	Unsigned integer 32	Greater than 0	False	✓	✓	✓
19	TFTP Server Timestamp	Unsigned integer 32	None	False	✓	✓	✓
20	TFTP Server Provisioned Modem Address	IP Address	None	False	✓	✓	✓
21	Software Upgrade TFTP Server	IP Address N	None	False	✓	✓	✓
22	Upstream Packet Classification Encoding	Compound	None	True		✓	✓
22.1	Classifier Reference	Unsigned integer 8	Between 1 and 255 inclusive	False		✓	✓
22.2	Classifier Identifier	Unsigned integer 16	Between 1 and 65535 inclusive	False		✓	✓
22.3	Service Flow Reference	Unsigned integer 16	Between 1 and 65535 inclusive	False		✓	✓
22.4	Service Flow Identifier	Unsigned integer 32	Greater than 0	False		✓	✓
22.5	Rule Priority	Unsigned integer 8	None	False		✓	✓
22.6	Classifier Activation State	Boolean	None	False		✓	✓
22.7	Dynamic Service Change Action	Unsigned integer 8	Less than 3	False		✓	✓
22.8	Classifier Error Encodings	Compound	None	False		✓	✓
22.8.1	Error Parameter	Sub Type	None	False		✓	✓
22.8.2	Error Code	Unsigned integer 8	Less than 26	False		✓	✓
22.8.3	Error Message	ZTAASCII	None	False		✓	✓
22.9	IP Packet Classification Encodings	Compound	None	False		✓	✓
22.9.1	IP Type of Service Range and Mask	Unsigned integer 8 triplet	None	False		✓	✓
22.9.2	IP Protocol	Unsigned integer 16	Less than 258	False		✓	✓
22.9.3	IP Source Address	IP Address	None	False		✓	✓
22.9.4	IP Source Mask	IP Address	None	False		✓	✓
22.9.5	IP Destination Address	IP Address	None	False		✓	✓
22.9.6	IP Destination Mask	IP Address	None	False		✓	✓
22.9.7	TCP/UDP Source Port Start	Unsigned integer 16	None	False		✓	✓
22.9.8	TCP/UDP Source Port End	Unsigned integer 16	None	False		✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
22.9.9	TCP/UDP Destination Port Start	Unsigned integer 16	None	False		✓	✓
22.9.10	TCP/UDP Destination Port End	Unsigned integer 16	None	False		✓	✓
22.10	Ethernet LLC Packet Classification Encodings	Compound	None	False		✓	✓
22.10.1	Destination MAC Address	MAC Address and Mask	None	False		✓	✓
22.10.2	Source MAC Address	MAC Address	None	False		✓	✓
22.10.3	Ethertype/DSAP/MacType	Unsigned integer 8 and unsigned integer 16	None	False		✓	✓
22.11	IEEE 802.1P/Q Packet Classification Encodings	Compound	None	False		✓	✓
22.11.1	IEEE 802.1P User_Priority	Unsigned integer 8 pair	Less than 8	False		✓	✓
22.11.2	IEEE 802.1Q VLAN_ID	Unsigned integer 16	None	False		✓	✓
22.43	Vendor Specific Classifier Parameters	Compound	None	False		✓	✓
22.43.8	Vendor ID	OUI	None	False		✓	✓
23	Downstream Packet Classification Encoding	Compound	None	True		✓	✓
23.1	Classifier Reference	Unsigned integer 8	Between 1 and 255 inclusive	False		✓	✓
23.2	Classifier Identifier	Unsigned integer 16		False		✓	✓
23.3	Service Flow Reference	Unsigned integer 16	Between 1 and 65535	False		✓	✓
23.4	Service Flow Identifier	Unsigned integer 32	Between 1 and 65535	False		✓	✓
23.5	Rule Priority	Unsigned integer 8	Greater than 0	False		✓	✓
23.6	Classifier Activation	Boolean	None	False		✓	✓
23.7	Dynamic Service Change Action	Unsigned integer 8	Less than 3	False		✓	✓
23.8	Classifier Error Encodings	Compound	None	False		✓	✓
23.8.1	Error Parameter	Sub Type	None	False		✓	✓
23.8.2	Error Code	Unsigned integer 8	Less than 26			✓	✓
23.8.3	Error Message	ZTASCII	None	False		✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
23.9	IP Classification Encodings	Compound	None	False		✓	✓
23.9.1	IP Type of Service Range and Mask	Unsigned integer 8	None	False		✓	✓
23.9.2	IP Protocol	Unsigned integer 16	Less than 258	False		✓	✓
23.9.3	IP Source Address	IP Address	None	False		✓	✓
23.9.4	IP Source Mask	IP Address	None	False		✓	✓
23.9.5	IP Destination Address	IP Address	None	False		✓	✓
23.9.6	IP Destination Mask	IP Address	None	False		✓	✓
23.9.7	TCP/UDP Source Port Start	Unsigned integer 16	None	False		✓	✓
23.9.8	TCP/UDP Source Port End	Unsigned integer 16	None	False		✓	✓
23.9.9	TCP/UDP Destination Port Start	Unsigned integer 16	None	False		✓	✓
23.9.10	TCP/UDP Destination Port End	Unsigned integer 16	None	False		✓	✓
23.10	Ethernet LLC Packet Classification Encodings	Compound					✓
23.10.1	Destination MAC Address	MAC Address and Mask\	None	False		✓	✓
23.10.2	Source MAC Address	MAC Address	None	False		✓	✓
23.10.3	Ethertype/DSAP/MacType	Unsigned integer 8 and unsigned integer 16	None	False		✓	✓
23.11	IEEE 802.1P/Q Packet Classification Encodings	Compound	None	False		✓	✓
23.11.1	IEEE 802.1P User_Priority	Unsigned integer 8 pair	Less than 8	False		✓	✓
23.11.2	IEEE 802.1Q VLAN_ID	Unsigned integer 16	None	False		✓	✓
23.43	Vendor Specific Classifier Parameters	Compound	None	False		✓	✓
23.43.8	Vendor ID	OUI	None	False		✓	✓
24	Upstream Service Flow Scheduling	Compound	None	True		✓	✓
24.1	Service Flow Reference	Unsigned integer 16	Greater than 0	False		✓	✓
24.3	Service Identifier	Unsigned integer 16	None	False		✓	✓
24.4	Service Class Name	ZTASCII	None	False		✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
24.5	Service Flow Error Encodings	Compound	None	True		✓	✓
24.5.1	Errored Parameter	Unsigned integer 8	None	False		✓	✓
24.5.2	Error Code	Unsigned integer 8	Less than 26	False		✓	✓
24.5.3	Error Message	ZTASCII	None	False		✓	✓
24.6	Quality of Service Parameter Set Type	Unsigned integer 8	Less than 8	False		✓	✓
24.7	Traffic Priority	Unsigned integer 8	Less than 8	False		✓	✓
24.8	Upstream Maximum Sustained Traffic Rate	Unsigned integer 32	None	False		✓	✓
24.9	Maximum Traffic Burst	Unsigned integer 32	None	False		✓	✓
24.10	Minimum Reserved Traffic Rate	Unsigned integer 32	None	False		✓	✓
24.11	Assumed Minimum Reserved Rate Packet Size	Unsigned integer 16	None	False		✓	✓
24.12	Timeout for active QoS Parameters	Unsigned integer 16	None	False		✓	✓
24.13	Timeout for Admitted QoS Parameters	Unsigned integer 16	None	False		✓	✓
24.14	Maximum Concatenated Burst	Unsigned integer 16	None	False		✓	✓
24.15	Service Flow Scheduling Type	Unsigned integer 8	Between 1-6 inclusive	False		✓	✓
24.16	Request/Transmission Policy	Unsigned integer 32	Less than 512	False		✓	✓
24.17	Nominal Polling Interval	Unsigned integer 32	None	False		✓	✓
24.18	Tolerated Poll Jitter	Unsigned integer 32	None	False		✓	✓
24.19	Unsolicited Grant Size	Unsigned integer 16	None	False		✓	✓
24.20	Nominal Grant Interval	Unsigned integer 32	None	False		✓	✓
24.21	Tolerated Grant Jitter	Unsigned integer 32	None	False		✓	✓
24.22	Grants per Interval	Unsigned integer 8	Less than 128	False		✓	✓
24.23	IP Type of Service Overwrite	Unsigned integer 8 pair	None	False		✓	✓
24.24	Unsolicited Grant Time Reference	Unsigned integer 32	None	False		✓	✓
24.43	Vendor Specific PHS Parameters	Compound	None	False		✓	✓
24.43.8	Vendor ID	OUI	None	False		✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
25	Downstream Service Flow Scheduling	Compound	None	True		✓	✓
25.1	Service Flow Reference	Unsigned integer 16	Greater than 0	False		✓	✓
25.3	Service Identifier	Unsigned integer 16	None	False		✓	✓
25.4	Service Class Name	ZTASCII	None	False		✓	✓
25.5	Service Flow Error Encodings	Compound	None	True		✓	✓
25.5.1	Errored Parameter	Unsigned integer 8	None	False		✓	✓
25.5.2	Error Code	Unsigned integer 8	Less than 26	False		✓	✓
25.5.3	Error Message	ZTASCII	None	False		✓	✓
25.6	Quality of Service Parameter Set Type	Unsigned integer 8	Less than 8	False		✓	✓
25.7	Traffic Priority	Unsigned integer 8	Less than 8	False		✓	✓
25.8	Downstream Maximum Sustained Traffic Rate	Unsigned integer 32	None	False		✓	✓
25.9	Maximum Traffic Burst	Unsigned integer 32	None	False		✓	✓
25.10	Minimum Reserved Traffic Rate	Unsigned integer 32	None	False		✓	✓
25.11	Assumed Minimum Reserved Rate Packet Size	Unsigned integer 16	None	False		✓	✓
25.12	Timeout for active QoS Parameters	Unsigned integer 16	None	False		✓	✓
25.13	Timeout for Admitted QoS Parameters	Unsigned integer 16	None	False		✓	✓
25.14	Maximum Downstream Latency	Unsigned integer 32	None	False		✓	✓
25.43	Vendor Specific PHS Parameters	Compound	None	False		✓	✓
25.43.8	Vendor ID	OUI	None	False		✓	✓
26	Payload Header Suppression	Compound	None	True		✓	✓
26.1	Classifier Reference	Unsigned integer 8	Greater than 0	False		✓	✓
26.2	Classifier Identifier	Unsigned integer 16	Greater than 0	False		✓	✓
26.3	Service Flow Reference	Unsigned integer 16	Greater than 0	False		✓	✓
26.4	Service Flow Identifier	Unsigned integer 32	Greater than 0	False		✓	✓
26.5	Dynamic Service Change Action	Unsigned integer 8	Less than 4	False		✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
26.6	Payload Header Suppression Error Encodings	Compound	None	False		✓	✓
26.6.1	Errored Parameter	Unsigned integer 8	None	False		✓	✓
26.6.2	Error Code	Unsigned integer 8	Less than 26	False		✓	✓
26.6.3	Error Message	ZTASCII	None	False		✓	✓
26.7	Payload Header Suppression Field (PHSF)	Bytes	None	False		✓	✓
26.8	Payload Header Suppression Index (PHSI)	Unsigned integer 8	Greater than 0	False		✓	✓
26.9	Payload Header Suppression Mask (PHSM)	Bytes	None	False		✓	✓
26.10	Payload Header Suppression Size (PHSS)	Unsigned integer 8	None	False		✓	✓
26.11	Payload Header Suppression Verification (PHSV)	Boolean	None	False		✓	✓
26.43	Vendor Specific PHS Parameters	Compound	None	False		✓	✓
26.43.8	Vendor ID	OUI	None	False		✓	✓
28	Maximum Number of Classifiers	Unsigned integer 16	None	False		✓	✓
29	Privacy Enable	Boolean	None	False		✓	✓
32	Manufacturer CVC	Bytes	None	False		✓	✓
33	Co-signer CVC	Bytes	None	False		✓	✓
34	SnmpV3 Kickstart Value	Compound	None	False		✓	✓
34.1	SnmpV3 Kickstart Security Name	NVTASCII	None	False		✓	✓
34.2	SnmpV3 Kickstart Manager Public Number	Bytes	None	False		✓	✓
35	Subscriber Management Control	Bytes	None	False		✓	✓
36	Subscriber Management CPE IP Table	Multiple IP Addresses	None	False		✓	✓
37	Subscriber Management Filter Groups	Bytes	None	False		✓	✓
38	Configuration File Element - docsisV3 Notification Receiver	Compound	None	False		✓	✓

Table 8-3 *DOCSIS Options and Version Support (continued)*

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
38.1	IP Address of Trap Receiver	IP address	None	False		✓	✓
38.2	UDP Port Number of Trap Receiver	unsigned integer 16	None	False		✓	✓
38.3	Type of Trap Sent by the PS	unsigned integer 8	None	False		✓	✓
38.4	Timeout	unsigned integer 32	None	False		✓	✓
38.5	Number of Retries When Sending an Inform After Sending the Inform First	unsigned integer 8	None	False		✓	✓
38.6	Notification Filtering Parameters	OID	None	False		✓	✓
38.7	Security Name to Use When Sending SNMP V3 Notification	NVTASCII	None	False		✓	✓
39	Enable 2.0 Mode	Enable/Disable	None	False			✓
40	Enable Test Mode	SubOptions	None	True			✓
41	Downstream Channel List	SubOptions	None	True			✓
41.1	Single Downstream Channel	SubOptions	None	True			✓
41.1.1	Single Downstream Channel Timeout	unsigned integer 16	None	False			✓
41.1.2	Single Downstream Channel Frequency	unsigned integer 32	None	False			✓
41.2	Downstream Frequency Range	SubOptions		True			✓
41.2.1	Downstream Frequency Range Timeout	unsigned integer 16	None	False			✓
41.2.2	Downstream Frequency Range Start	unsigned integer 32	Multiples of 62500	False			✓
41.2.3	Downstream Frequency Range End	unsigned integer 32	Multiples of 62500	False			✓
41.2.4	Downstream Frequency Range Step Size	unsigned integer 32	None	False			✓
41.3	Default Scanning	unsigned integer 32	None	True			✓
42	Multicast MAC Address	MAC Address	None	True			✓
43	Vendor-Specific Information	Compound	None	True	✓	✓	✓

Table 8-3 DOCSIS Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
43.1	Static Downstream Frequency	Unsigned integer 32	None	False	✓	✓	✓
43.2	Sync Loss Timeout	Unsigned integer 32	None	False	✓	✓	✓
43.3	Update Boot Monitor Image	NVTASCII	None	False	✓	✓	✓
43.4	Power Backoff	Unsigned integer 16	None	False	✓	✓	✓
43.8	Vendor ID	OUI	None	False	✓	✓	✓
43.9	Update Factory System Image	Boolean	None	False	✓	✓	✓
43.10	Phone Lines	Unsigned integer 8	None	False	✓	✓	✓
43.11	IP Precedence Settings	Compound	None	True	✓	✓	✓
43.11.1	IP Precedence Value	Unsigned integer 8	None	False	✓	✓	✓
43.11.2	Rate Limit	Unsigned integer 32	None	False	✓	✓	✓
43.128	IOS Configuration Filename	NVTASCII	None	False	✓	✓	✓
43.129	IOS Config File Without Console Disable	NVTASCII	None	False	✓	✓	✓
43.131	IOS CLI Command	NVTASCII	None	True	✓	✓	✓
43.132	1.0 Plus Flow Encodings	Compound	None	False	✓	✓	✓
43.132.1	1.0 Plus Flow ID	Unsigned integer 8	None	False	✓	✓	✓
43.132.2	Class ID	Unsigned integer 8	None	False	✓	✓	✓
43.132.3	Unsolicited Grant Size	Unsigned integer 16	Between 1-65535 inclusive	False	✓	✓	✓
43.132.4	Nominal Grant Interval	Unsigned integer 32	Between 1-65535 inclusive	False	✓	✓	✓
43.132.5	Grants Per Interval	Unsigned integer 8	Between 0-127 inclusive	False	✓	✓	✓
43.132.6	Embedded Voice Calls	Unsigned integer 8	Between 0-127 inclusive	False	✓	✓	✓
43.132.7	Hold Queue Length	Unsigned integer 16	Between 0-4096 inclusive	False	✓	✓	✓
43.132.8	Fair Queue	Compound	None	False	✓	✓	✓

Table 8-3 *DOCSIS Options and Version Support (continued)*

Option Number	Description	Encoding	Validation	Multi-valued	DOCSIS Version		
					1.0	1.1	2.0
43.132.8.1	Congestive Discard Threshold	Unsigned integer 16	Between 1-4096 inclusive	False	✓	✓	✓
43.132.8.2	Number of Dynamic Conversation Queues	Unsigned integer 16	Between 16-4096 inclusive	False	✓	✓	✓
43.132.8.3	Number of Reservable Conversation Queues	Unsigned integer 16	Between 0-1000 inclusive	False	✓	✓	✓
43.132.9	Custom Queue List Length	Unsigned integer 8	Between 1-16 inclusive	False	✓	✓	✓
43.132.10	Random Detection	Boolean	None	False	✓	✓	✓
43.132.11	Priority Group	Unsigned integer 8	Between 1-16 inclusive	False	✓	✓	✓
43.132.12	Service Policy File	NVTASCII	None	False	✓	✓	✓
43.132.13	Inactivity Timer	Unsigned integer 16	Between 1-10080 inclusive	False	✓	✓	✓
43.132.14	COS Tag	NVTASCII	None	False	✓	✓	✓
43.133	Downstream Sub Channel ID	Unsigned integer 8	Between 0-15 inclusive	False	✓	✓	✓
43.134	SU Tag	NVTASCII	None	False	✓	✓	✓
255	End-of-Data Marker	No length and no value	N/A	False	✓	✓	✓

PacketCable Option Support

Table 8-4 identifies the PacketCable 1.0 MTA options that BAC supports.

Table 8-4 *PacketCable MTA 1.0 Options*

Number	Description	Encoding	Validation	Multi-valued	PacketCable Version	
					1.0	1.1
11	SNMP MIB Object	SNMPVarBind with 1 byte length	None	True	✓	✓
38	SNMPv3 Notification Receiver	SubOptions	None	True	✓	✓
38.1	SNMPv3 Notification Receiver IP Address	IPAddress	None	False	✓	✓

Table 8-4 PacketCable MTA 1.0 Options (continued)

Number	Description	Encoding	Validation	Multi-valued	PacketCable Version	
					1.0	1.1
38.2	SNMPv3 Notification Receiver UDP Port Number	Unsigned integer 16	None	False	✓	✓
38.3	SNMPv3 Notification Receiver Trap Type	SNMPTrapType	From 1 to 5	False	✓	✓
38.4	SNMPv3 Notification Receiver Timeout	Unsigned integer 16	None	False	✓	✓
38.5	SNMPv3 Notification Receiver Retries	Unsigned integer 16	From 0 to 255	False	✓	✓
38.6	Notification Receiver Filtering Parameters	OID	None	False	✓	✓
38.7	Notification Receiver Security Name	NVTASCII	None	False	✓	✓
43	Vendor-Specific Information	SubOptions	None	True	✓	✓
43.8	Vendor ID	OUI	None	False	✓	✓
64	SNMP MIB Object	SNMPVarBind with 2 byte length	None	True	✓	✓
254	Telephony Config File Start/End	Unsigned integer 8	Must be 1 or 255	False	✓	✓

Non-Secure CableHome Option Support

Table 8-5 identifies the non-secure CableHome options that BAC supports.

Table 8-5 Non-Secure CableHome Options and Version Support

Option Number	Description	Encoding	Validation	Multi-valued	CableHome Version
					1.0
0	PAD	No length and no value	None	True	✓
9	Software Upgrade Filename	NVTASCII	None	False	✓
10	SNMP Write-Access Control	OIDCF	None	True	✓
12	Modem IP Address	IP Address	None	False	✓
14	CPE Ethernet MAC Address	MACAddress	None	True	✓
21	Software Upgrade TFTP Server	IPAddress	None	False	✓
28	SNMP MIB Object	SNMPVarBind	None	True	✓
32	Manufacturer CVC	Bytes	None	False	✓
33	Co-signer CVC	Bytes	None	True	✓

Table 8-5 Non-Secure CableHome Options and Version Support (continued)

Option Number	Description	Encoding	Validation	Multi-valued	CableHome Version
					1.0
34	SnmpV3 Kickstart Value	SubOptions	None	False	✓
34.1	SnmpV3 Kickstart Security Name	NVTASCII	None	False	✓
38	SNMPv3 Notification Receiver	SubOptions	None	True	✓
38.1	SNMPv3 Notification Receiver IP Address	IPAddress	None	False	✓
38.2	SNMPv3 Notification Receiver UDP Port Number	Unsigned integer 16	None	False	✓
38.3	SNMPv3 Notification Receiver Trap Type	SNMPTrapType	From 1 to 5	False	✓
38.4	SNMPv3 Notification Receiver Timeout	Unsigned integer 16	None	False	✓
38.5	SNMPv3 Notification Receiver Retries	Unsigned integer 16	None	False	✓
38.6	Notification Receiver Filtering Parameters	OID	None	False	✓
38.7	Notification Receiver Security Name	NVTASCII	None	False	✓
43	Vendor-Specific Information	SubOptions	None	True	✓
43.1	Vendor ID	OUI	None	False	✓
53	PS MIC. A 20 octet SHA-1 hash of PS config file	Bytes	None	False	✓
255	End-of-Data Marker	No length and no value	None	False	✓

Using the Configuration File Utility

You use the configuration file utility to test, validate, and view PacketCable 1.0/1.1/1.5, DOCSIS 1.0/1.1/2.0, and CableHome template and configuration files. These activities are critical to successful deployment of individualized configuration files. See [Developing Template Files, page 8-1](#), for more information on templates.

The configuration file utility is available only when the RDU is installed; the utility is installed in the *BPR_HOME/rdu/bin* directory.

Both the template file being encoded and the binary file being decoded must reside in the directory from which the configuration file utility is invoked.

All examples in this section assume that the RDU is operating and that these conditions apply:

- The BAC application is installed in the default home directory (*/opt/CSCObpr*).
- The RDU login name is **admin**.
- The RDU login password is **changeme**.

**Note**

Some of the examples in this section were trimmed whenever the omitted information is of no consequence to the example or its outcome. Instances where this occurs are identified by an ellipsis (...) that precedes the example summary.

This section discusses these topics:

- [Testing Template Processing for a Local Template File, page 8-31](#)
- [Testing Template Processing for an External Template File, page 8-32](#)
- [Specifying Macro Variables at the Command Line, page 8-34](#)
- [Specifying a Device for Macro Variables, page 8-35](#)
- [Specifying Output to a Binary File, page 8-36](#)
- [Viewing a Local Binary File, page 8-37](#)
- [Viewing an External Binary File, page 8-38](#)
- [Activating PacketCable Basic Flow, page 8-39](#)

Running the Configuration File Utility

In subsequent procedures and examples, the phrase “run the configuration file utility” means to enter the **runCfgUtil.sh** command from the directory specified. To run the configuration file utility, run this command from the *BAC_home/rdu/bin* directory:

runCfgUtil.sh *options*

The available *options* include:

- **-c secret**—Specifies the CMTS shared secret when parsing a DOCSIS template file. To specify the default shared secret, enter **-c cisco**.
- **-cablehome**—Identifies the input file as a CableHome portal service configuration file. Do not use this with either the **-docsis** or **-pkt** options.
- **-d**—Decodes the binary input file. Do not use this with the **-e** option.
- **-docsis**—Specifies the input file is a DOCSIS configuration file. Do not use this default with the **-pkt** option.
- **-v version**—Specifies the DOCSIS version being used. For example, if you are using DOCSIS 1.1, enter **-v 1.1**. If you do not specify the version number, the command defaults to use DOCSIS 2.0. The values that BAC supports are 1.0, 1.1, and 2.0.
- **-e**—Encodes the template input file. Do not use this default with the **-d** option.
- **-g**—Generates a template file from either a DOCSIS, PacketCable, or CableHome binary file.
- **-h host:port**—Specifies the host and port. The default port number is 49187.
- **-i device id**—Specifies the device to use when parsing macro variables. For example, if your device ID is 1,6,00:00:00:00:00:01, enter **-i 1,6,00:00:00:00:00:01**. When using this option, you must also use the **-u** and **-p** options, respectively, to specify the username and password. Do not use this with the **-m** option.
- **-l filename**—Identifies the input file as being on the local file system. For example, if your input file is called any_file, enter **-l any_file**. Do not use this with the **-r** option.

- **-loc**—Specifies the PacketCable locale, na (North America) or euro (Europe). The default is na. If the MTA is euro-MTA, then the locale should be set to euro.
- **-m macros**—Specifies key value pairs for macro variables. The format is key=value. If you require multiple macro variables, use a double comma separator between the key value pairs, for example, key_1=value_1,,key_2=value_2. Do not use this with the **-i** option.
- **-p password**—Specifies the password to use when connecting to the RDU. For example, if your password is 123456, enter **-p 123456**.
- **-o filename**—Saves parsed template file as a binary file. For example, if you want the output to be found in a file call *op_file*, enter **-o op_file**.
- **-pkt**—Identifies the input file as a PacketCable MTA configuration file. Do not use this with the **-docsis** option.
- **-r filename**—Identifies the input file as an external file that has been added to the RDU. For example, if your file is called *file25*, enter **-r file25**. When using this option you must also use the **-u** and **-p** options, to specify the username and password, respectively. Do not use this with the **-l** option.
- **-s**—Displays the parsed template or the contents of the binary file in a human readable format.
- **-t**—Specifies the PacketCable encoding type: Secure or Basic (the default is Secure).
- **-u username**—Specifies the username to use when connecting to the RDU. For example, if your username is admin, enter **-u admin**.

**Note**

The configuration file utility does not include Option 19 (TFTP server timestamp) and Option 20 (TFTP server provisioned modem address) in the template file; the BAC TFTP mixing, however, does. Also, options 6 (CM MIC) and 7 (CMTS MIC) are both automatically inserted into the encoded template file. Therefore, you do not have to specify these message integrity checks (MIC).

Adding a Template to BAC

To use the configuration file utility to test BAC templates:

-
- Step 1** Develop the template as described in [Developing Template Files, page 8-1](#). If the template includes other templates, make sure all the referenced templates are in the same directory.
- Step 2** Run the configuration file utility on the local file system. You can check the syntax for the template, or have the configuration file utility process the template as IGS would, and return output.
- If the template contains macro variables, perform these operations in the order specified:
- a. Test with command line substitution.
 - b. Test with a device that has been added to your RDU.
- Step 3** Add the template (and any included templates that are used) to the RDU.
- Step 4** Run the configuration file utility to parse an external file. See [Testing Template Processing for an External Template File, page 8-32](#).
- If the template contains macro variables, perform these operations in the order specified:
- a. Test with command line substitution.
 - b. Test with a device that has been added to your RDU.

Step 5 After all tests succeed, configure a Class of Service to use the template.

Converting a Binary File Into a Template File

Use the **runCfgUtil.sh** command to convert binary configuration memory files into template files. BAC dynamic configuration generation is based on templates that are created. Automatically converting existing, tested, binary files to template files speeds the process and reduces the possibility of introducing errors.

Syntax Description

runCfgUtil.sh -g -l *binary_file* -o *template_file*

- **-g**—Specifies that a template file needs to be generated from an input binary file
- **-l *binary_file***—Specifies the local input file, including the pathname. In all cases, the input binary filename will have a .cm file extension; bronze.cm, for example.
- **-o *template_file***—Specifies the output template file, including the pathname. In all cases, the output template file will have a .tmpl file extension; for example, test.tmpl.

To convert a binary file into a template file:

Step 1 Change directory to /opt/CSCObpr/rdu/samples/.

Step 2 Select a template file to use. This example uses an existing binary file called unprov.cm.

Step 3 Run the configuration file utility using this command:

```
# ./runCfgUtil.sh -g -l unprov.cm -o test.tmpl -docsis
```

-docsis—Specifies the input file to be a DOCSIS configuration file.

After running the utility, results similar to these should appear:

```
Broadband Access Center Configuration Utility
Version: 2.7.1, Revision: 1.26

#####
## Template File Generator
## Generated on Fri Jan 12 16:12:51 EST 2007
#####

#####
## Each generated option will be represented by the following:
## The first line will represent a description of the
## generated option
## The second line will represent the generated option
## The third line will represent the custom version
## of the generated option
#####

# (3) Network Access Control
Option 3 01
# Option 3 hex 01

# (4.1) Class ID
Option 4.1 1
# Option 4.1 hex 01
```

```

# (4.2) Maximum Downstream Rate
Option 4.2 128000
# Option 4.2 hex 0001F400

# (4.3) Maximum Upstream Rate
Option 4.3 64000
# Option 4.3 hex 0000FA00

# (4.4) Upstream Channel Priority
Option 4.4 1
# Option 4.4 hex 01

# (4.5) Guaranteed Minimum Upstream Channel Data Rate
Option 4.5 0
# Option 4.5 hex 00000000

# (4.6) Maximum Upstream Channel Transmit Burst
Option 4.6 1600
# Option 4.6 hex 0640

# (4.7) Class-of-Service Privacy Enable
Option 4.7 00
# Option 4.7 hex 00

# (11) SNMP MIB Object
Option 11
.iso.org.dod.internet.experimental.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNm
mAccessEntry.docsDevNmAccessStatus.1, INTEGER, createAndGo
# Option 11 hex 3082000F060A2B060103530102010701020104

...
# (18) Maximum Number of CPES
Option 18 1
# Option 18 hex 01

```

Testing Template Processing for a Local Template File

Use the `runCfgUtil.sh` command to test processing for template files stored on the local file system.

Syntax Description

`runCfgUtil.sh -pkt -l file`

- `-pkt`—Identifies the input file as a PacketCable MTA file.
- `-l`—Specifies that the input file is on the local file system.
- `file`—Identifies the input template file being parsed.

To parse a template file that is on the local file system:

- Step 1** Change directory to `/opt/CSCObpr/rdu/samples/packet_cable`.
- Step 2** Select a template file to use. This example uses an existing template file called `unprov_packet_cable.tmpl`. The `-pkt` option is used because this is a PacketCable MTA template.
- Step 3** Run the configuration file utility using this command:


```
# runCfgUtil.sh -pkt -l unprov_packet_cable.tmpl
```

unprov_packet_cable.tmpl—Identifies the input template file being parsed.

After running the utility, results similar to these should appear:

Off	File Bytes	Option	Description	Value
0	FE0101	254	Telephony Config File Start/End	1
3	0B153013060E 2B06010401A30B 0202010101 0700020102	11	SNMP MIB Object	.iso.org.dod.internet. private.enterprises.ca bleLabs.clabProject.cl abProjPacketCable.pktc MtaMib.pktcMtaMibObjec ts.pktcMtaDevBase. pktcMtaDevEnable d.0, INTEGER, false(2)
...				
0 error(s), 0 warning(s) detected. Parsing of unprov_packet_cable.tmpl was successful. The file unprov_packet_cable.tmpl was parsed successfully in 434 ms. The parser initialization time was 92 ms. The parser parse time was 342 ms.				

Testing Template Processing for an External Template File

Use the **runCfgUtil.sh** command to test processing of external template files.

Syntax Description

runCfgUtil.sh -r file -u username -p password -docsis

- **-r**—Identifies the input file as an external file that has been added to the RDU.
- *file*—Identifies the input template file being parsed.
- **-u username**—Specifies the username to use when connecting to the RDU.
- **-p password**— Specifies the password to use when connecting to the RDU.
- **-docsis**—Identifies the file as a DOCSIS template.

To parse a template file that has been added to the RDU:

-
- Step 1** Change directory to `/opt/CSCObr/rdu/samples/docsis`.
- Step 2** Select a template file to use. This example uses an existing template file called `unprov.tmpl`. The `-docsis` option is used because a DOCSIS template is being used.
- Step 3** Run the configuration file utility using this command:
- ```
runCfgUtil.sh -r unprov.tmpl -u admin -p changeme -docsis
```
- **unprov.tmpl**—Identifies the input file.
  - **admin**—Identifies the username.
  - **changeme**—Identifies the password.

After running the utility, results similar to these should appear:



**Note** The results shown here are for illustration only and have been trimmed for brevity.

| Off | File Bytes                                   | Option | Description                    | Value                                |
|-----|----------------------------------------------|--------|--------------------------------|--------------------------------------|
| 0   | 030101                                       | 3      | Network Access Control         | On                                   |
| 3   | 041F                                         | 4      | Class of Service               |                                      |
| 5   | 010101                                       | 4.1    | Class ID                       | 1                                    |
| 8   | 02040000FA00                                 | 4.2    | Maximum Downstream Rate        | 128000 bits/sec                      |
| 14  | 03040000FA00                                 | 4.3    | Maximum Upstream Rate          | 64000 bits/sec                       |
| 20  | 040101                                       | 4.4    | Upstream Channel Priority      | 1                                    |
| ... |                                              |        |                                |                                      |
| 252 | 06108506547F<br>C9152B44DB95<br>5420843EF6FE | 6      | CM MIC Configuration Setting   | 8506547FC9152B44<br>DB955420843EF6FE |
| 270 | 0710644B675B<br>70B7BD3E09AC<br>210F794A1E8F | 7      | CMTS MIC Configuration Setting | 644B675B70B7BD3E<br>09AC210F794A1E8F |
| 288 | FF                                           | 255    | End-of-Data Marker             |                                      |
| 289 | 00                                           | 0      | PAD                            |                                      |
| 290 | 00                                           | 0      | PAD                            |                                      |
| 291 | 00                                           | 0      | PAD                            |                                      |

0 error(s), 0 warning(s) detected. Parsing of unprov.tmpl was successful.  
 The file unprov.tmpl was parsed successfully in 375 ms.  
 The parser initialization time was 63 ms.  
 The parser parse time was 312 ms.

## Testing Template Processing for a Local Template File and Adding Shared Secret

Use the **runCfgUtil.sh** command to test processing for a template file and add a shared secret that you specify.

### Syntax Description

**runCfgUtil.sh -e -docsis -l file -c secret**

- **-e**—Identifies the encode option.
- **-docsis**—Identifies the input file as a DOCSIS template file.
- **-l**—Specifies that the input file is on the local file system.
- *file*—Identifies the input template file being parsed.
- **-c**—Specifies the CMTS shared secret when parsing a DOCSIS template file.
- *secret*—Identifies the new shared secret. The default shared secret is **cisco**.

To parse a locally saved template file, and set a user specified shared secret:

- Step 1** Change directory to `/opt/CSCObpr/rdu/samples/docsis`.
- Step 2** Select a template file to parse. This example uses an existing template file called `unprov.tmpl`. The `-docsis` option is used because this is a DOCSIS template.
- Step 3** Run the configuration file utility using this command:

```
runCfgUtil.sh -e -docsis -l unprov.tmpl -c shared
```

- **unprov.tmpl**—Identifies the input file on the local file system.
- **shared**—Identifies that new shared secret.

After running the utility, results similar to these should appear:

| Off | File Bytes                                   | Option | Description                    | Value                                |
|-----|----------------------------------------------|--------|--------------------------------|--------------------------------------|
| 0   | 030100                                       | 3      | Network Access Control         | Off                                  |
| 3   | 041F                                         | 4      | Class of Service               |                                      |
| 5   | 010101                                       | 4.1    | Class ID                       | 1                                    |
| 8   | 02040001F400                                 | 4.2    | Maximum Downstream Rate        | 128000 bits/sec                      |
| 14  | 03040000FA00                                 | 4.3    | Maximum Upstream Rate          | 64000 bits/sec                       |
| 20  | 040101                                       | 4.4    | Upstream Channel Priority      | 1                                    |
| ... |                                              |        |                                |                                      |
| 252 | 06108506547F<br>C9152B44DB95<br>5420843EF6FE | 6      | CM MIC Configuration Setting   | 8506547FC9152B44<br>DB955420843EF6FE |
| 270 | 0710644B675B<br>70B7BD3E09AC<br>210F794A1E8F | 7      | CMTS MIC Configuration Setting | 644B675B70B7BD3E<br>09AC210F794A1E8F |
| 288 | FF                                           | 255    | End-of-Data Marker             |                                      |
| 289 | 00                                           | 0      | PAD                            |                                      |
| 290 | 00                                           | 0      | PAD                            |                                      |
| 291 | 00                                           | 0      | PAD                            |                                      |

0 error(s), 0 warning(s) detected. Parsing of unprov.tmpl was successful.  
The file unprov.tmpl was parsed successfully in 375 ms.  
The parser initialization time was 63 ms.  
The parser parse time was 312 ms.

## Specifying Macro Variables at the Command Line

Use the `runCfgUtil.sh` command to specify macro variables.

**Syntax Description** `runCfgUtil.sh -e -l file -m "macros"`

- **-e**—Identifies the encode option.
- **-l**—Specifies the input file is on the local file system.
- **file**—Identifies the input template file being parsed.



- **-m**—Specifies the macro variables to be substituted when parsing a template.
- “*macros*”—Identifies the desired macros. When multiple macro variables are required, insert a double comma separator between each macro.

To specify values for macro variables at the command line:

- 
- Step 1** Change directory to /opt/CSCObpr/rdu/samples/templates.
- Step 2** Select a template file to use.
- Step 3** Identify the macro variables in the template. In this example, the macro variables are macro1 (option 3) and macro11 (option 4.2).
- Step 4** Identify the values for the macro variables. The value for macro1 will be set to 1, and the value for macro11 to 64000.
- Step 5** Run the configuration file utility using this command:

```
runCfgUtil.sh -e -l macro.tpl -m "macro1=1,,macro11=64000"
```

- **macro.tpl**—Identifies the input file.
- **macro1=1,,macro11=64000**—Identifies the key value pairs for macro variables. Since multiple macro variables are necessary, a double comma separator is inserted between the key value pairs.

After running the utility, results similar to these should appear:

| Off | File Bytes   | Option | Description               | Value          |
|-----|--------------|--------|---------------------------|----------------|
| 0   | 030101       | 3      | Network Access Control    | On             |
| 3   | 041F         | 4      | Class of Service          |                |
| 5   | 010101       | 4.1    | Class ID                  | 1              |
| 8   | 02040000FA00 | 4.2    | Maximum Downstream Rate   | 64000 bits/sec |
| 14  | 03040000FA00 | 4.3    | Maximum Upstream Rate     | 64000 bits/sec |
| 20  | 040101       | 4.4    | Upstream Channel Priority | 1              |

...

```
0 error(s), 0 warning(s) detected. Parsing of macro.tpl was successful.
The file macro.tpl was parsed successfully in 854 ms.
The parser initialization time was 76 ms.
The parser parse time was 778 ms.
```

---

## Specifying a Device for Macro Variables

Use the **runCfgUtil.sh** command to specify a device for macro variables.

**Syntax Description** **runCfgUtil.sh -e -l file -i MAC -u username -p password**

- **-e**—Identifies the encode option.
- **-l**—Specifies the input file is on the local file system.
- *file*—Identifies the input template file being parsed.
- **-i**—Specifies the device to use when parsing macro variables.

- *MAC*—Identifies the MAC address of the device.
- **-u username**—Specifies the username to use when connecting to the RDU.
- **-p password**— Specifies the password to use when connecting to the RDU.

To specify a device to be used for macro variable substitution:

- 
- Step 1** Change directory to `/opt/CSCObpr/rdu/samples/templates`.
- Step 2** Select a template file to use. This example will use the existing template file, `macro.tmpl`.
- Step 3** Identify the macro variables in the template. In this example, the macro variables are `macro1` (option 3) and `macro11` (option 4.2).
- Step 4** Identify the device to use. This example will assume that the device exists in the RDU and has the macro variables set as properties. The value for `macro1` will be set to 1, and the value for `macro11` to 64000.
- Step 5** Run the configuration file utility using this command:

```
runCfgUtil.sh -l macro.tmpl -i "1,6,00:01:02:03:04:05" -u admin -p changeme
```

- **macro.tmpl**—Identifies the input file.
- **1,6,00:01:02:03:04:05**—Identifies the MAC address of the device. The MAC address used here is for example purposes only.
- **admin**—Identifies the default username.
- **changeme**—Identifies the default password.

After running the utility, results similar to these should appear:

| Off | File Bytes   | Option | Description               | Value          |
|-----|--------------|--------|---------------------------|----------------|
| 0   | 030101       | 3      | Network Access Control    | On             |
| 3   | 041F         | 4      | Class of Service          |                |
| 5   | 010101       | 4.1    | Class ID                  | 1              |
| 8   | 02040000FA00 | 4.2    | Maximum Downstream Rate   | 64000 bits/sec |
| 14  | 03040000FA00 | 4.3    | Maximum Upstream Rate     | 64000 bits/sec |
| 20  | 040101       | 4.4    | Upstream Channel Priority | 1              |
| ... |              |        |                           |                |

```
0 error(s), 0 warning(s) detected. Parsing of macro.tmpl was successful.
The file macro.tmpl was parsed successfully in 823 ms.
The parser initialization time was 102 ms.
The parser parse time was 803 ms.
```

## Specifying Output to a Binary File

Use the `runCfgUtil.sh` command to specify the output of a parsed template as a binary file.

---

**Syntax Description** `runCfgUtil.sh -l input_file -o output_file`

- **-l**—Specifies that the input file is on the local file system.

- *input\_file*—Identifies the input template file being parsed.
- **-o**—Specifies that the parsed template file is to be saved as a binary file.
- *output\_file*—Identifies the name of the file in which the binary contents of the parsed template file are stored.

To specify the output from parsing a template to a binary file:

- 
- Step 1** Change directory to /opt/CSCObpr/rdu/samples/templates.
- Step 2** Select a template file to use.
- Step 3** Identify the name of the output file. This example will use unprov.cm.
- Step 4** Run the configuration file utility using this command:

```
runCfgUtil.sh -l unprov.tmpl -o unprov.cm
```

- **unprov.tmpl**—Identifies the existing template file being parsed into a binary file.
- **unprov.cm**—Identifies the output filename to be used.

After running the utility, results similar to these should appear:

```
Broadband Access Center Configuration Utility
Version: 2.7.1

0 error(s), 0 warning(s) detected. Parsing of unprov.tmpl was successful.
The file unprov.tmpl was parsed successfully in 595 ms.
The parser initialization time was 262 ms.
The parser parse time was 333 ms.
```

---

## Viewing a Local Binary File

Use the **runCfgUtil.sh** command to view a binary file stored in the local system.

---

### Syntax Description

**runCfgUtil.sh -d -l file**

- **-d**—Specifies that the command is going to decode a binary input file for viewing.
- **-l**—Identifies that the input file resides on the local file system.
- *file*—Identifies the existing binary input file to be viewed.

To view a binary file that is on the local file system:

- 
- Step 1** Change directory to /opt/CSCObpr/rdu/samples/packet\_cable.
- Step 2** Select a binary file to view.
- Step 3** Run the configuration file utility using this command:

```
runCfgUtil.sh -d -l unprov_packet_cable.bin
```

**unprov\_packet\_cable.bin**—Identifies the existing binary input file to be viewed.

After running the utility, results similar to these should appear:

| Off | File Bytes                                                 | Option | Description                     | Value                                                                                                                                                                                                   |
|-----|------------------------------------------------------------|--------|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 0   | FE0101                                                     | 254    | Telephony Config File Start/End | 1                                                                                                                                                                                                       |
| 3   | 0B153013060E<br>2B06010401A30B<br>02020101010700<br>020102 | 11     | SNMP MIB Object                 | .iso.org.dod.internet.<br>private.enterprises.ca<br>bleLabs.clabProject.<br>clabProjPacketCa<br>ble.pktcMtaMib.pktcMta<br>MibObjects<br>.pktcMtaDevBase.<br>pktcMtaDevEnable<br>d.0, INTEGER, fals e(2) |
| ... |                                                            |        |                                 |                                                                                                                                                                                                         |

## Viewing an External Binary File

Use the `runCfgUtil.sh` command to view an external binary file.

### Syntax Description

`runCfgUtil.sh -d -r file -u username -p password`

- **-d**—Specifies that the command is going to decode a binary input file for viewing.
- **-r**—Identifies the input file an external file that has been added to the RDU.
- *file*—Identifies the existing external binary file in the RDU.
- **-u username**—Specifies the username to use when connecting to the RDU.
- **-p password**— Specifies the password to use when connecting to the RDU.

To view a binary file that has been added to the RDU:

**Step 1** Select a binary file to view. This example will use the existing binary file `unprov.cm`, and assumes that the RDU is `localhost:49187`.

**Step 2** Run the configuration file utility using this command:

```
runCfgUtil.sh -d -r unprov.cm -u admin -p changeme
```

- **unprov.cm**—Identifies the existing external binary file in the RDU.
- **admin**—Identifies the default username.
- **changeme**—Identifies the default password.

After running the utility, results similar to these should appear:

| Off | File Bytes                                   | Option | Description                    | Value                                |
|-----|----------------------------------------------|--------|--------------------------------|--------------------------------------|
| 0   | 030100                                       | 3      | Network Access Control         | Off                                  |
| 3   | 041F                                         | 4      | Class of Service               |                                      |
| 5   | 010101                                       | 4.1    | Class ID                       | 1                                    |
| 8   | 02040001F400                                 | 4.2    | Maximum Downstream Rate        | 128000 bits/sec                      |
| 14  | 03040000FA00                                 | 4.3    | Maximum Upstream Rate          | 64000 bits/sec                       |
| 20  | 040101                                       | 4.4    | Upstream Channel Priority      | 1                                    |
| ... |                                              |        |                                |                                      |
| 252 | 06108506547F<br>C9152B44DB95<br>5420843EF6FE | 6      | CM MIC Configuration Setting   | 8506547FC9152B44<br>DB955420843EF6FE |
| 270 | 0710644B675B<br>70B7BD3E09AC<br>210F794A1E8F | 7      | CMTS MIC Configuration Setting | 644B675B70B7BD3E<br>09AC210F794A1E8F |
| 288 | FF                                           | 255    | End-of-Data Marker             |                                      |
| 289 | 00                                           | 0      | PAD                            |                                      |
| 290 | 00                                           | 0      | PAD                            |                                      |
| 291 | 00                                           | 0      | PAD                            |                                      |

0 error(s), 0 warning(s) detected. Parsing of unprov.tpl was successful.  
 The file unprov.tpl was parsed successfully in 375 ms.  
 The parser initialization time was 63 ms.  
 The parser parse time was 312 ms.

## Activating PacketCable Basic Flow

Use the **runCfgUtil.sh** command to support the generation and insertion of the PacketCable Basic Flow integrity hash into a Basic Flow static configuration file.

### Syntax Description

**runCfgUtil.sh -t {basic | secure} -r filename -u username -p password -pkt**

- **basic**—Calculates and inserts a PacketCable Basic Flow integrity hash into an MTA static configuration file.
- **secure**—Stops the insertion of the PacketCable Basic Flow integrity hash into an MTA static configuration file. This is the default setting.
- **-r**—Identifies the input file an external file that has been added to the RDU.
- **filename**—Identifies the input external file.
- **-u username**—Specifies the username to use when connecting to the RDU.
- **-p password**—Specifies the password to use when connecting to the RDU.
- **-pkt**—Identifies the input file as a PacketCable MTA configuration file.

To support the generation and insertion of the PacketCable Basic Flow integrity hash into a Basic flow static configuration file:

- 
- Step 1** Change directory to `/opt/CSCObpr/rdu/samples`.
  - Step 2** Select the Basic Flow static configuration file into which you want to insert the PacketCable Basic Flow integrity hash. This example uses the `generic_mta.tmpl`.
  - Step 3** Run the configuration file utility using this command:

```
runCfgUtil.sh -t basic -r generic_mta.tmpl -u admin -p changeme -pkt
```

- **generic\_mta.tmpl**—Identifies the Basic Flow static configuration file.
- **admin**—Identifies the default username.
- **changeme**—Identifies the default password.

After running the utility, results similar to these should appear:

```
Broadband Access Center Configuration Utility
Version: 2.7.1, Revision: 1.26

Off File Bytes Option Description Value

0 FE0101 254 Telephony Config File Start/End 1
3 0B153013060E 11 SNMP MIB Object .iso.org.dod.internet.
 2B06010401A3 private.enterprises.ca
 0B0202010101 bleLabs.clabProject.cl
 0700020101 abProjPacketCable.pktc
 MtaMib.pktcMtaMibObjec
 ts.pktcMtaDevBase.pktc
 MtaDevEnabled.0, INTEGE
 R,true(1)

26 0B2530230610 11 SNMP MIB Object .iso.org.dod.internet.
 2B06010401A3 private.enterprises.ca
 0B0202020102 bleLabs.clabProject.cl
 01010109040F abProjPacketCable.pktc
 434D532E4950 SigMib.pktcSigMibObjec
 464F4E49582E ts.pktcNcsEndPntConfig
 434F4D Objects.pktcNcsEndPntC
 onfigTable.pktcNcsEndP
 ntConfigEntry.pktcNcsE
 ndPntConfigCallAgentId
 .9,STRING,CMS.IPFONIX.
 COM

241 FE01FF 254 Telephony Config File Start/End 255
...

0 error(s), 0 warning(s) detected. Parsing of generic_mta.tmpl was successful.
The file generic_mta.tmpl was parsed successfully in 88 ms.
The parser initialization time was 36 ms.
The parser parse time was 52 ms.
```

A file with a `.tmpl` extension is assumed to be a dynamic configuration template, for which the Basic hash calculation and insertion occur transparently during template processing; as a result, you can use the same template for provisioning in the Secure and Basic modes.

However, if you want to convert a Secure static binary configuration file to a Basic static configuration file before inserting the hash, follow this procedure:

- a. Convert the Secure static file to a template, by using:

```
runCfgUtil -l input_static_filename -pkt -g -o output_template_filename
```

- b. Convert the Secure static template into a Basic static configuration file, by using:

```
runCfgUtil -t basic -l input_template_name -o output_Basic_static_filename -pkt
```

This command calculates and inserts the Basic integrity hash into the Basic static configuration file.

---







## CHAPTER 9

# Understanding the Administrator User Interface

---

This chapter describes how to access the Broadband Access Center (BAC) administrator user interface and explains the interface. The topics covered are:

- [Configuring the Administrator User Interface, page 9-1](#)
- [Accessing the Administrator User Interface, page 9-2](#)
- [Studying the Administrator User Interface, page 9-5](#)

## Configuring the Administrator User Interface

Before you use the administrator user interface, examine the *adminui.properties* file. This file contains a variety of controls that specify the behavior of the interface.

You can open this file using any text editor, and change its content to perform the functions that you want. After you save the changes, restart the user interface so that the changes take effect.

To start the administrator user interface, enter:

```
etc/init.d/bprAgent start tomcat
```

To stop the administrator user interface, enter:

```
etc/init.d/bprAgent stop tomcat
```

To restart the administrator user interface, enter:

```
/etc/init.d/bprAgent restart tomcat
```

You can configure the user interface by using the options available in the *adminui.properties* file. These options are controlled by BAC settings or defined in the *adminui.properties* file in the *BAC\_home/rdu/conf* directory. The configuration parameters are:

- */adminui/port*—Specifies the listening port of the RDU. The default port number is 49187.
- */adminui/fqdn*—Specifies the fully qualified domain name of the host on which the RDU is running. The default value is the FQDN of the host; for example, *bac\_test.ACME.COM*.
- */adminui/maxReturned*—Specifies the maximum number of search results. The default value is 1000.
- */adminui/pageSize*—Specifies the number of search results displayed per page. You can set this number at 25, 50, or 75. The default value is 25.
- */adminui/refresh*—Specifies if the refresh function is enabled or disabled. This option is, by default, disabled.

- `/adminui/extensions`—Specifies if the use of extensions in BAC is enabled or disabled. You use extensions to augment BAC behavior or add support for new device technologies. The use of extensions is, by default, enabled.
- `/adminui/maxFileSize`—Specifies the maximum size of a file uploaded to BAC. The default file size is 4 MB.
- `/adminui/refreshRate`—Specifies the duration (in seconds) after which a screen is refreshed. The default value is 90 seconds. Before setting a value for this option, ensure that the `/adminu/refresh` option is enabled.
- `/adminui/timeout`—Specifies the length of time (in seconds) after which an idle session times out. The default period is set as 300 seconds.
- `/adminui/noOfLines`—Specifies the last number of lines from `rdu.log` or `dpe.log` that appear on the user interface. The default number of lines that appear is 250.

### Example 9-1 Sample `adminui.properties` File

```
/adminui/port=49187
/adminui/fqdn=doc.cisco.com
/adminui/maxReturned=1000
/adminui/pageSize=25
/adminui/refresh=disabled
/adminui/extensions=enabled
/adminui/maxFileSize=10000000
/adminui/refreshRate=90
/adminui/timeout=300
/adminui/noOfLines=250
```

## Accessing the Administrator User Interface

You can access the BAC user interface from any computer that can access the URL corresponding to the BAC application.

### Logging In

You can log in to the BAC user interface as an administrative user, a Read/Write user, or a Read-Only user. Although each user type has different capabilities, as described in [User Management, page 10-1](#), you access the user interface in the same way.

Complete this procedure to access the BAC administrator user interface:

---

**Step 1** Launch your web browser.

[Table 9-1](#) lists the browsers supported in this BAC release.

**Table 9-1 Browser Platform Support**

| Platform                      | Supported Browsers                                        |
|-------------------------------|-----------------------------------------------------------|
| Windows 2000 (Service Pack 2) | Internet Explorer 6.0 and above<br>Netscape 4.7 and above |

**Table 9-1** *Browser Platform Support (continued)*

| Platform            | Supported Browsers     |
|---------------------|------------------------|
| Red Hat Linux (7.1) | Netscape 4.7 and above |
| Solaris (2.9)       | Netscape 4.7 and above |

**Step 2** Enter the administrator's location using this syntax:

`http://machine_name:port_number/`

- *machine\_name*—Identifies the computer on which the RDU is running.



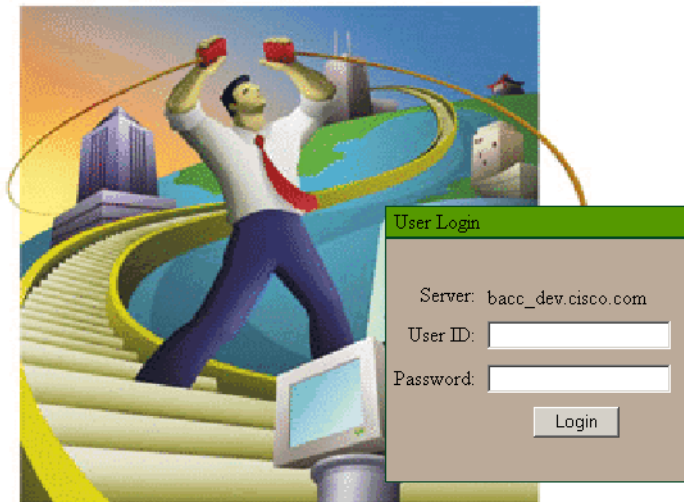
**Note** To access the administrator user interface via HTTP over SSL, also known as HTTPS, enter: `https://machine_name/`

- *port number*—Identifies the computer port on which the server side of the administrator application runs. The default port number is:
  - 8100 for HTTP over TCP
  - 8443 for HTTP over SSL

The main login page, shown in [Figure 9-1](#), appears.

**Figure 9-1** *Login Page*

## Broadband Access Center for Cable



This product contains cryptographic features and is subject to United States and local country laws governing import, export, transfer and use. Delivery of Cisco cryptographic products does not imply third-party authority to import, export, distribute or use encryption. Importers, exporters, distributors and users are responsible for compliance with U.S. and local country laws. By using this product you agree to comply with applicable laws and regulations. If you are unable to comply with U.S. and local laws, return this product immediately. A summary of U.S. laws governing Cisco cryptographic products may be found at: <http://www.cisco.com/www/export/crypto/tool/stqrg.html>. If you require further assistance please contact us by sending email to [export@cisco.com](mailto:export@cisco.com).

129898

**Step 3** Enter the default username (**admin**) and password (**changeme**).

If you are logging in for the first time, the Change Password screen appears. Enter a new password and confirm it.



**Note** Note that the FQDN of the RDU server you are logged in to appears in the User Login area.

**Step 4** Click **Login**.

The Main Menu page, shown in [Figure 9-2](#), appears.

**Figure 9-2** Main Menu Page

**Broadband Access Center for Cable** Logout

User:admin Role:Administrator

**Main Menu**

[Configuration](#)  
Use this page to view or change Broadband Access Center for Cable configuration settings.

[Devices](#)  
Use this page to manage (add, delete or search) IP devices.

[Nodes](#)  
Use this page to manage Nodes and Node types.

[Servers](#)  
Use this page to view the status of the Broadband Access Center for Cable servers.

[Users](#)  
Use this page to manage users.

**CISCO SYSTEMS**

129899



**Note** You can use the link at the top of the page to add the license keys for each technology that you are authorized to use. For details, see [Configuring Broadband Access Center, page 11-1](#).

## Logging Out

Complete this procedure to log out of BAC:

- Step 1** Click the **Logout** tab at the top right corner of any page.
- Step 2** A confirmation dialog appears. Click **OK**.
- This returns you to the User Login page. See [Figure 9-1](#).

## Studying the Administrator User Interface

The BAC administrator user interface, as shown in [Figure 9-3](#), is divided into three separate areas:

- Banner area—Contains all menu options including: a **Logout** button, the Primary Navigation bar, and the Secondary Navigation bar.
- Content area—Contains all BAC data resulting from the functions performed from the Primary and Secondary Navigation bars.
- Results area—Indicates how many pages of information are available. This is particularly useful when reviewing the results of searches.

Some BAC administrator pages may also contain View, Delete, Submit, or Reset controls.

**Figure 9-3 Administrator User Interface**

The screenshot shows the administrator interface for the Broadband Access Center for Cable. The interface is divided into three main areas as indicated by the annotations:

- Banner area:** Contains the title "Broadband Access Center for Cable", a "Logout" button, and the Primary Navigation bar (Configuration, Devices, Nodes, Servers, Users). Below this is the Secondary Navigation bar (Class of Service, Custom Property, Defaults, DHCP Criteria, External Files, License Keys, Publishing) and the user information "User:admin Role:Administrator".
- Content area:** Contains the main content of the page, including the "Manage Class of Service" section with a dropdown menu (currently showing "DOCSISModem") and an "Add" button. Below this is a table listing classes of service with delete buttons.
- Results area:** Located at the bottom of the content area, showing "Result Pages: 1".




| Class of Service                     | Delete |
|--------------------------------------|--------|
| <a href="#">sample-bronze-docsis</a> |        |
| <a href="#">sample-gold-docsis</a>   |        |
| <a href="#">sample-silver-docsis</a> |        |
| <a href="#">unprovisioned-docsis</a> |        |

129900

## Understanding the Icons

The BAC administrator user interface features icons, which you can use to perform specific functions. [Table 9-2](#) defines these icons.

**Table 9-2 Administrator User Interface Icons**

| Icon                                                                              | Description                                                                                                                                                                                                                       |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|  | This icon serves as: <ul style="list-style-type: none"> <li>• View Details icon—Enables you to view details of a given device or file.</li> <li>• Operations icon—Enables you to execute operations on a given device.</li> </ul> |
|  | Delete icon—Deletes a specific object.                                                                                                                                                                                            |
|  | Export icon—Exports the contents of a specific file to the client computer.                                                                                                                                                       |

These icons are used in sections describing procedures that you perform via the administrator user interface. These sections include:

- [Using the Administrator User Interface, page 10-1](#)
- [Configuring Broadband Access Center, page 11-1](#)



# CHAPTER 10

## Using the Administrator User Interface

---

This chapter describes the administration tasks performed from the Broadband Access Center (BAC) administrator user interface. These tasks mainly involve monitoring the actions of various BAC components and include:

- [User Management, page 10-1](#)
- [Device Management, page 10-4](#)
- [Node Management, page 10-16](#)
- [Viewing Servers, page 10-19](#)



**Note**

---

The procedures described in this chapter are presented in a tutorial manner. Wherever possible, examples are included to illustrate the possible results of each procedure.

---

## User Management

Managing users involves adding, modifying, and deleting users who administer BAC. Depending on your user type, you can use this menu to add, modify, and delete users. This menu displays all users configured to use BAC and identifies their user types.

There are three types of BAC users: an Administrator, a Read/Write user, and a Read-Only user. Each has different levels of access, with unique permissions to ensure access control and the integrity of provisioning data.

The assigned user type appears near the top right corner of every screen on the administrator user interface.

## Administrator

BAC recognizes only one administrator and allows this user to view, add, modify, and delete device data, and create other users. As an Administrator, you can also change other users' permissions from Read/Write to Read Only, and vice versa. In addition, you have the ability to change the passwords of any other user type.

You cannot delete the Administrator user.

## Read/Write User

As a Read/Write user, you can perform the same functions as the administrator except creating other users, changing the user types of others, or changing their passwords. Read/Write users can change their own passwords.

## Read-Only User

As a Read-Only user, you have basic access including the ability to change your password and to view, but not change, device data. You cannot perform any action that is considered disruptive. You cannot, for example, perform reset or regenerate configurations.



### Note

During migration from an acceptable previous release to BAC 2.7.1, all migrated users are assigned Read/Write privileges.

You can add and delete users only if you are logged in as the Administrator.

This section contains instructions for managing BAC users including:

- [Adding a New User](#)
- [Modifying Users](#)
- [Deleting Users](#)

## Adding a New User

Adding a new user is a simple process of entering the user's name and creating a password. However, while creating a new user you do have to determine which type of user it will be; a Read/Write user or a Read-Only user. BAC comes with one Administrator user already created; you cannot create an Administrator as a new user. To add a new user:

- Step 1** Click **Users**, from the Main Menu or the Primary Navigation bar. The Manage Users page, described in [Figure 10-1](#), appears.

**Figure 10-1** Manage Users Page

**Broadband Access Center for Cable** Logout

Configuration | Devices | Nodes | Servers | **Users**

User:admin Role:Administrator

---

**CISCO SYSTEMS** | **Manage Users**  
Use this page to manage (add, modify or delete) users.

Add

| User                      | Description             | Role          | Delete |
|---------------------------|-------------------------|---------------|--------|
| <a href="#">Ace Duffy</a> | Assistant Administrator | Read Write    |        |
| <a href="#">admin</a>     | Administrator           | Administrator |        |

Result Pages: 1

129901



- Step 2** Click **Add** to display the Add User page.
- Step 3** Enter the new user's username and a password.
- Step 4** Confirm the new user's password, and select whether the new user's role is to be read only or read/write. See earlier sections for complete descriptions of each user type.
- Step 5** Enter a short description of the new user.



---

**Tip** Use the description field to identify the user's job or position, something that identifies the unique aspects of the new user.

---

- Step 6** Click **Submit**.
- The Manage Users page appears with the new user added.



---

**Note** The new user's password must be recorded and stored in a safe place to help prevent loss or theft of the password and possible unauthorized entry.

---

## Modifying Users


Although any user type can modify their password and user description, only the administrator can modify any other user's information.

To modify user properties:

- 
- Step 1** From the Main Menu or the Primary Navigation bar, click **Users**. The Manage User page appears.
- Step 2** Click the correct user name to display the Modify User page for that user.
- Step 3** Make the necessary changes to the password, user type (you must be logged in as the Administrator), and the user's description.
- Step 4** Click **Submit**.
- The Manage Users page appears with the modified user information.
- 

## Deleting Users

Only the administrator can delete any other user that appears in the Manage Users page. You cannot delete the default user, called **admin**. To delete a user:

- 
- Step 1** From the Main menu or the Primary Navigation bar, click **Users**. The Manage User page appears.
- Step 2** Click the **Delete** icon () corresponding to the user you want to delete. The Delete User dialog box appears.
- Step 3** Click **OK**. The Manage Users page appears without the deleted user.
-

# Device Management

Use the Devices menu to provision and manage various devices. You can:

- Search for a specific device or for a group of devices that share criteria that you specify. See [Searching for Devices, page 10-5](#).
- Add, modify, or delete devices in the RDU database. See:
  - [Adding Device Records, page 10-14](#)
  - [Modifying Device Records, page 10-14](#)
  - [Deleting Devices, page 10-15](#)
- View device data, such as configuration, properties. See [Viewing Device Details, page 10-9](#).
- Regenerate device configurations. See [Regenerating Device Configurations, page 10-15](#).
- Relate and unrelate any device to a specific node. See [Relating and Unrelating Devices, page 10-16](#).
- Reset, or reboot, any selected device. See [Resetting Devices, page 10-16](#).
- Return a device configuration to its default condition without rebooting the device. See [Unregistering a Device, page 10-16](#).

## Manage Devices Page

The Manage Devices page appears when you click **Devices** on the Main menu or the primary navigation bar. This page contains the fields and controls necessary to perform all device management functions.

Figure 10-2 identifies the details for the Manage Devices page.

Figure 10-2 Manage Devices Page

The screenshot shows the 'Manage Devices' page in the Broadband Access Center for Cable. At the top, there is a navigation bar with 'Configuration', 'Devices' (selected), 'Nodes', 'Servers', and 'Users'. Below this, the user is identified as 'admin' with the role of 'Administrator'. The main heading is 'Manage Devices' with a sub-instruction: 'Use this page to manage (add, delete or search) devices, then to view the details of the device listed.' Below the heading is a search section with a 'Search Type' dropdown set to 'MAC Address Search', a text input field for 'MAC Address or MAC Address wildcard' containing an asterisk, and a 'Page Size' dropdown set to '25'. A 'Search' button is to the right. Below the search section are several action buttons: 'Add', 'Delete', 'Regenerate', 'Relate', 'Reset', 'Unrelate', and 'Unregister'. The main content is a table with the following data:

| Identifiers                                    | Device Type | Status        | Details                 |
|------------------------------------------------|-------------|---------------|-------------------------|
| <input type="checkbox"/> 1.6.00:00:00:00:00:00 | DOCSISModem | Unprovisioned | <a href="#">Details</a> |
| <input type="checkbox"/> 1.6.00:00:00:00:00:01 | DOCSISModem | Unprovisioned | <a href="#">Details</a> |
| <input type="checkbox"/> 1.6.00:00:00:00:00:aa | DOCSISModem | Unprovisioned | <a href="#">Details</a> |

At the bottom left, it says 'Result Pages: 1'. On the right side of the page, there is a vertical number '129902'.

## Searching for Devices

By using BAC, you can search for device information in a number of different ways.

To select the search type, from the Manage Devices pages, click the Search Type drop-down list. Subsequent search pages contain screen components that may be unique to the search type selected.

The Manage Device page utilizes two separate but related areas to generate search results that let you perform many device management functions. These areas are the Search Type drop-down list, which defines which search to perform, and the search value field, which qualifies the search type. You can perform these searches:

- **MAC Address Search**—Searches by using the precise MAC address for a specific modem or all devices with a specific vendor-prefix that unambiguously identifies the equipment vendor. The vendor-prefix is the first 3 octets of the MAC address. For example, for MAC address 1,6,aa:bb:cc:dd:ee:ff, the vendor-prefix is “aa:bb:cc”.
- **FQDN Search**—Searches by using the fully qualified domain name (FQDN) associated with the device that is assigned by the DNS Server.
- **IP Address Search**—Searches by returning all devices on the network that currently have the specified DHCP leased IP address.
- **Node search**—Searches devices which are part of a particular node or node type.
- **Owner ID Search**—Searches by using the owner ID associated with the device. The owner ID may identify the service subscriber’s account number, for example. This search function does not support wildcard searching.
- **Class of Service search includes:**
  - **Registered Class of Service search**—Searches by using the Class of Service that a device has been provisioned with.
  - **Related Class of Service search**—Searches by using both the registered and selected Class of Service.
  - **Selected Class of Service search**—Searches by using the Class of Service selected by the RDU for a device that, for one reason or another, cannot retain its registered Class of Service.
- **DHCP Criteria search includes:**
  - **Registered DHCP Criteria search**—Searches for devices that belong to certain DHCP criteria.
  - **Related DHCP Criteria search**—Searches using both the registered and selected DHCP criteria.
  - **Selected DHCP Criteria search**—Searches using the DHCP criteria selected by the RDU for a device that, for one reason or another, cannot retain its registered DHCP Criteria.

**Note**

Under normal circumstances, the Related and Selected Class of Service and the Related and Selected DHCP Criteria should be identical. If they are not, you should investigate the reason and modify the Selected Class of Service/DHCP Criteria to match the Related Class of Service/DHCP Criteria.

Some searches that you can perform allow the use of a wildcard character (\*) to enhance the search function. BAC provides specific wildcards for each search, as described in [Table 10-1](#).

**Table 10-1** Searches Supported for Device Management

| Menu Search                        | Search Type Option                                                                                                                                                                                                                                                       |
|------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MAC Address Search                 | Complete MAC address or partial MAC address followed by a wildcard asterisk (*) character at the end of the string.<br>For example, to search for a device with the MAC address 1,6,aa:bb:cc:dd:ee:ff, you can try 1,6*                                                  |
| FQDN Search                        | Complete FQDN or partial FQDN string beginning with a wildcard asterisk (*) character.<br>For example, to search for a device with the FQDN IGW-1234.ACME.COM, you can try: <ul style="list-style-type: none"> <li>• *.acme.com</li> <li>• *.com</li> <li>• *</li> </ul> |
| IP Address Search                  | IP Address<br>Wildcard searches are not supported. You must enter the complete IP address.                                                                                                                                                                               |
| Node Search                        | Node Name <ul style="list-style-type: none"> <li>• Drop-down list</li> </ul> Complete MAC address or partial MAC address followed by a wildcard asterisk (*) character at the end of the string.                                                                         |
| Owner ID Search                    | Owner ID<br>Wildcard searches are not supported. You must enter the complete owner ID.                                                                                                                                                                                   |
| Registered Class of Service Search | Class of Service (Type) <ul style="list-style-type: none"> <li>• Drop-down list</li> </ul>                                                                                                                                                                               |
| Registered DHCP Criteria Search    | DHCP Criteria (Type) <ul style="list-style-type: none"> <li>• Drop-down list</li> </ul>                                                                                                                                                                                  |
| Related Class of Service Search    | Class of Service (Type) <ul style="list-style-type: none"> <li>• Drop-down list</li> </ul>                                                                                                                                                                               |
| Related DHCP Criteria Search       | DHCP Criteria (Type) <ul style="list-style-type: none"> <li>• Drop-down list</li> </ul>                                                                                                                                                                                  |
| Selected Class of Service Search   | Class of Service (Type) <ul style="list-style-type: none"> <li>• Drop-down list</li> </ul>                                                                                                                                                                               |
| Selected DHCP Criteria Search      | DHCP Criteria (Type) <ul style="list-style-type: none"> <li>• Drop-down list</li> </ul>                                                                                                                                                                                  |

**Note**

Cisco does not recommend using the last wildcard search (\*) in systems that support hundreds of thousands, or more, devices. This can return many thousands of search results, and use extensive system resources sufficient to impact performance.

In addition, a Page Size drop-down lets you limit the number of search results displayed per page. You can select 25, 50, or 75 results for display. If the number of results returned for a search exceeds the number selected, a screen prompt appears at the lower left corner of the page. These controls let you scroll backward or forward one page at a time, or to select a specific page.

**Note**

A maximum of 1,000 results are returned for any query, with a maximum of 75 results displayed per page. You can change the default maximum by modifying the `/adminui/maxReturned` property, in the `BAC_home/rdu/conf/adminui.properties` file, and then running the **bprAgent restart tomcat** command (which resides in the `/etc/init.d/` directory) to restart the BAC Tomcat component.

## Device Management Controls

These buttons are located directly below the search function fields and are generally used in conjunction with the search function. For example, you might search for devices belonging to a specific group of devices in order to perform some sort of management function.

The following buttons are available, although each management function may not be available depending on the search type used.

**Add**

The Add button lets you add a new device to the RDU database. See [Adding Device Records, page 10-14](#).

**Delete**

The Delete button lets you delete any selected devices from the RDU database. See [Deleting Devices, page 10-15](#).

**Regenerate**

Use the Regenerate button to force immediate regeneration of instructions for selected devices.

**Relate**

The Relate button lets you associate a device (by using its MAC address) with a specific node.

**Reset**

The Reset button automatically reboots the selected device.

**Unrelate**

The Unrelate button lets you cancel the relationship between a selected device and the node that the device is currently related to.

**Unregister**

The Unregister button lets you reset a device back to its defaults as if it had just booted on the network. Searching for devices returns results under the following headings or links that appear on the page:

**Identifier**

Identifies all devices matching the search criteria. Each of the identifiers displayed has a link to another page from which you can modify the device.

**Device Type**

Displays the available device types. Available selections include:

- ATA186
- ATA188
- CableHome MAN-Data
- CableHome MAN-WAN
- DOCSIS Modem
- Computer
- PacketCable Media Terminal Adapter (MTA)

**Status**

Identifies whether or not the device is provisioned. A provisioned device has been registered using the application programming interface (API), or the administrator user interface, and has booted on the network.

**Details**

Displays all available details for the selected device. For additional information, see [Viewing Device Details, page 10-9](#).

- Registered DHCP Criteria search—Searches for devices that belong to certain DHCP criteria.
- Selected DHCP Criteria search—Searches using the DHCP criteria selected by the RDU for a device that, for one reason or another, cannot retain its registered DHCP Criteria.
- Related DHCP Criteria search—Searches using both the registered and selected DHCP criteria.

**Note**

Under normal circumstances the Related/Selected Class of Service and Related/Selected DHCP Criteria should be identical. If they are not, you should investigate the reason and modify the Selected Class of Service/DHCP Criteria to match the Related Class of Service/DHCP Criteria.

- Fully qualified domain name (FQDN) search—Is useful when searching for devices that are identified through the FQDN assigned by the DNS Server, especially when the device MAC address is unknown.  
For example, **www.cisco.com** is a fully qualified domain name. Where **www** identifies the host, **cisco** identifies the second-level domain, and **.com** identifies the third-level domain.
- IP address search—Returns all devices on the network that currently have the specified DHCP leased IP address.
- MAC address search— Is best used when you know the precise MAC address for a specific modem or when all devices with a specific vendor-prefix unambiguously identify the equipment vendor. Therefore, if you perform a MAC address search, you can identify, by the MAC address, the manufacturer and type of device. See [Troubleshooting Devices by MAC Address, page 13-21](#), for information on how you can effectively use this search criteria.




**Note** The vendor-prefix is the first 3 octets of the MAC address. For example, for MAC address 1,6,aa:bb:cc:dd:ee:ff, the vendor-prefix is “aa:bb:cc”.

- Node search—Is used to search for nodes and node types that you have already created.

- Owner ID search— Identifies a device, the service subscriber's account number, or anything else that uniquely identifies that device. This function does not support wildcard searching.

## Viewing Device Details

You can view the details of any device identified in the search results. To view any device details, click the **View Details** icon () corresponding to the device you want to view, and the View Device Details page appears.

[Figure 10-3](#) provides a sample View Device Details page.



---

**Note**

The information that appears in the View Device Details page is largely dependent on the type of device you choose. [Figure 10-3](#) identifies the details that typically appear for most devices.

---

Figure 10-3 View Device Details Page

**Broadband Access Center for Cable** Logout

Configuration **Devices** Nodes | Servers | Users

User:admin Role:Administrator

---

**CISCO SYSTEMS** **View Device Details**  
Use this page to view the details of the device listed.

| Device Details                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Device Type:                       | DOCSISModem                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| MAC Address:                       | 1,6,aa:bb:cc:dd:ee:ff                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| FQDN:                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Host Name:                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Domain Name:                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Behind Device:                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Provisioning Group:                | <a href="#">default</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Registered DHCP Criteria:          | default                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CPE DHCP Criteria:                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Device Properties:                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Device Provisioned State:          | Provisioned                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Device Registered State:           | Registered                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Relay Agent Circuit Identifier:    | 80:01:00:22                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Relay Agent Remote Identifier:     | aa:bb:cc:dd:ee:ff                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Client Identifier:                 | aa:bb:00:cc:dd:ee:ff                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Client Request Host Name:          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Registered Class Of Service:       | <a href="#">arris-cm</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Owner Identifier:                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Detected Properties:               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Selected Properties:               | /docsis/version=1.0                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Is Behind Required Device:         | Yes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Is In Required Provisioning Group: | Yes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Selected Access:                   | REGISTERED                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Selected Class of Service:         | <a href="#">arris-cm</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Selected DHCP Criteria:            | <a href="#">unprovisioned-docsis</a>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Selected Explanation:              | Because the device is registered.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Selected Reason:                   | REGISTERED                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Related Node Name (Node Type):     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| DHCP Information                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| DHCP Inform Dictionary:            | provisioning-group=default                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| DHCP Request Dictionary:           | giaddr=10.7.1.1, dhcp-message-type=01, htype=01, relay-agent-circuit-id=01:04:80:01:00:22, dhcp-parameter-request-list=66,67,1,3,2,4,7,128,6,12,15,122,177, client-id-created-from-mac-address=0, relay-agent-remote-id=02:06:00:00:ca:b4:fb:9c, hlen=06, client-id=01:00:00:ca:b4:fb:9c, dhcp-class-identifier=docsis1.1:052401010102010103010104010105010106010107010f0801100901000a01010b01080c0101, chaddr=00:00:ca:b4:fb:9c, vendor-encapsulated-options=01:00:02:03:45:43:4d:03:08:45:43:4d:3a:45:4d:54:41:04:0f:34:35:4e:47:37:44:31:38:37:37:30:30:36:34:34:05:03:34:2e:30:06:12:54:53:2e:30:34:2e:30:31:2e:30:34:2e:30:33:31:35:30:34:07:00:08:06:30:30:30:30:43:41:09:06:54:4d:34:30:32:50:0a:19:41:72:72:69:73:20:49:6e:74:65:72:61:63:74:69:76:65:2c:20:4c:2e:4c:2e:43:2e |
| DHCP Response Dictionary:          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| DHCP Environment Dictionary:       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Lease Information                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| IP Address:                        | 10.7.1.6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| DHCP Lease Properties:             | /network/lastTransTime=83, /network/isLeased=Leased, /network/leaseTime=221                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Technology Specific Information    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| XGCP Ports:                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| DOCSIS Version:                    | 1.1                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

Table 10-2 identifies the fields shown in Figure 10-3.



**Table 10-2** View Device Details Page

| Field or Button                | Description                                                                                                                                                                                        |
|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Device Details</b>          |                                                                                                                                                                                                    |
| Device Type                    | Identifies the device type.                                                                                                                                                                        |
| MAC Address                    | Identifies the MAC address of the device.                                                                                                                                                          |
| FQDN                           | Identifies the fully qualified domain name (FQDN) for the device; for example, IGW-1234.ACME.COM.                                                                                                  |
| Host Name                      | Identifies the host. For example, in the FQDN description above, IGW-1234 is the hostname.                                                                                                         |
| Domain Name                    | Identifies the domain within which the host resides. For example, in the FQDN description above, ACME.COM is the domain name.                                                                      |
| Behind Device                  | Identifies the device that is behind this device.                                                                                                                                                  |
| Provisioning Group             | Identifies the provisioning group to which the device has been pre-assigned or assigned automatically. This is an active link that, if clicked, displays the Provisioning Group Details page.      |
| Registered DHCP Criteria       | Identifies the DHCP criteria used. This is an active link that, if clicked, displays the appropriate Modify DHCP Criteria page.                                                                    |
| CPE DHCP Criteria              | Identifies the DHCP criteria used for customer premises equipment when in the Promiscuous mode.                                                                                                    |
| Device Properties              | Identifies any properties, other than those that appear on this page, that can be set for this device. This field includes the display of custom properties.                                       |
| Device Provisioned State       | Specifies if the device is provisioned. A device is provisioned only when it is registered and has booted on the network.                                                                          |
| Device Registered State        | Identifies if the device is registered.                                                                                                                                                            |
| Relay Agent Circuit Identifier | Identifies the relay agent circuit identifier of the device. This is equivalent to DHCP Option 82, suboption 1.                                                                                    |
| Relay Agent Remote Identifier  | Identifies the relay agent remote identifier of the device. This is equivalent to DHCP Option 82, suboption 2.                                                                                     |
| Client Identifier              | Identifies the client identification used by the device in its DHCP messages.                                                                                                                      |
| Client Request Host Name       | Identifies the hostname that the client requests in its DHCP messages.                                                                                                                             |
| Registered Class of Service    | Identifies the Class of Service assigned to the device. If a different Class of Service has been selected for the device by extension, an additional field with Selected Class of Service appears. |
| Owner Identifier               | Identifies the device. This may be a user ID or an account number; the field may also be blank.                                                                                                    |
| Detected Properties            | Identifies properties returned by the RDU device detection extension(s) when configuration for the device is generated.                                                                            |
| Selected Properties            | Identifies properties returned by the RDU service level selection extension(s) for the detected device type when the configuration for the device is generated.                                    |

Table 10-2 View Device Details Page (continued)

| Field or Button                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Is Behind Required Device         | Specifies “false” if the <code>IPDeviceKeys.MUST_BE_BEHIND_DEVICE</code> property has been used to establish a required relay agent device and the service level selection extension(s) determines that this device did not boot behind the required relay agent.                                                                                                                                                                                                                                                                                                                               |
| Is In Required Provisioning Group | Specifies “false” if the <code>IPDeviceKeys.MUST_BE_IN_PROV_GROUP</code> property has been used to establish a required provisioning group and the service level selection extension(s) determines that this device did not boot in the required provisioning group.                                                                                                                                                                                                                                                                                                                            |
| Selected Access                   | Identifies the access granted to the device by the service level selection extension(s): <ul style="list-style-type: none"> <li>REGISTERED—Indicates that the device was registered and met requirements for access.</li> <li>PROMISCUOUS—Indicates that the device’s provisioning will be based on policies assigned to its relay agent.</li> <li>DEFAULT—Indicates that the device will be provisioned with default access for its device type.</li> <li>OTHER—Not used by the default extensions built into BAC and is provided for use by custom extensions.</li> </ul>                     |
| Selected Class of Service         | Identifies the name of the Class of Service used to generate the configuration for the device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Selected DHCP Criteria            | Identifies the name of the DHCP criteria used to generate the configuration for the device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Selected Explanation              | Provides a textual description of why the service level selection extension(s) selected the access they granted the device. For example, the device may have been granted default access because it did not boot in its required provisioning group.                                                                                                                                                                                                                                                                                                                                            |
| Selected Reason                   | Identifies why the service level selection extension(s) selected the access they granted the device as an enumeration code. The possible values are: <ul style="list-style-type: none"> <li>NOT_BEHIND_REQUIRED_DEVICE</li> <li>NOT_IN_REQUIRED_PROV_GROUP</li> <li>NOT_REGISTERED</li> <li>OTHER</li> <li>PROMISCUOUS_ACCESS_ENABLED</li> <li>REGISTERED</li> <li>RELAY_NOT_IN_REQUIRED_PROV_GROUP</li> <li>RELAY_NOT_REGISTERED</li> </ul> <p>Most of these indicate violations of requirements for granting registered or promiscuous access, resulting in default access being granted.</p> |
| Related Node Name (Node Type)     | Identifies the node(s) name and node type to which this device is related. See <a href="#">Node Management, page 10-16</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**Table 10-2 View Device Details Page (continued)**

| Field or Button                                                                                                                     | Description                                                                                                                                                                         |
|-------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>DHCP Information</b>                                                                                                             |                                                                                                                                                                                     |
| DHCP Inform Dictionary                                                                                                              | Identifies additional information that the Cisco Network Registrar extensions send to the RDU when requesting the generation of a configuration. This is for internal BAC use only. |
| DHCP Request Dictionary                                                                                                             | Identifies the DHCP Discover or DHCP Request packet details sent from the Network Registrar extensions to the RDU when requesting the generation of a configuration.                |
| DHCP Response Dictionary                                                                                                            | This field is for internal BAC use only; it should always be empty.                                                                                                                 |
| DHCP Environment Dictionary                                                                                                         | This field is for internal BAC use only; it should always be empty.                                                                                                                 |
| <b>Lease Information</b>                                                                                                            |                                                                                                                                                                                     |
| IP Address                                                                                                                          | Identifies the IP address of the device.                                                                                                                                            |
| DHCP Lease Properties                                                                                                               | Identifies the lease properties; this field should always be empty.                                                                                                                 |
| <b>Technology-Specific Information</b>                                                                                              |                                                                                                                                                                                     |
| <b>Note</b> The technology-specific information identifies only data that is relevant for the technologies you are licensed to use. |                                                                                                                                                                                     |
| XGCP Ports                                                                                                                          | Identifies the ports on which the Gateway Control Protocol is active.                                                                                                               |
| DOCSIS Version                                                                                                                      | Identifies the DOCSIS version currently in use.                                                                                                                                     |

## Managing Devices

The Devices menu lets you add devices to the RDU database and update preprovisioned data. Device management includes:

- Adding, deleting, and modifying RDU devices records
- Regenerating configurations
- Relating devices to management objects, such as Provisioning Group, Class of Service, and Group.

This section describes how to perform various device management functions on new or existing devices. Several information fields appear consistently in all device management pages. These fields include:

- Device Type—When adding a device, this is a drop-down list that identifies the available device types you can create within BAC. Available selections, as they appear on screen, include:
  - ATA186
  - ATA188
  - CableHomeManData
  - CableHomeManWan
  - DOCSISmodem
  - Computer
  - PacketCableMTA

When modifying a device, the device type cannot be edited or changed.

- **MAC Address**—This is the MAC address of the device being added. Enter the MAC address of the device being added in this field. When doing this, ensure that you enter the commas (,) and colons (:) appropriately. For example, 1,6,00:00:00:00:00:AE.
- **Host Name**—Identifies the device host. For example, from an FQDN of node.cisco.com, node is the hostname.
- **Domain Name**—Identifies the domain within which the host resides. For example, from an FQDN of node.cisco.com, cisco.com is the domain name.
- **Owner Identifier**—Identifies the device by using something other than the hostname. This may be a user ID, an account number, or may be left blank.
- **Class of Service**—Specifies the Class of Service that the device is to be provisioned with.
- **DHCP Criteria**—Specifies the DHCP criteria that the device is to be provisioned with.

Depending on the page displayed, additional information may appear. Where appropriate, this additional information is identified in the following procedures.

## Adding Device Records

To add a device record:

- 
- Step 1** From the Manage Devices page, click **Add**. The Add Device page appears.
  - Step 2** Choose the device type and Class of Service, and complete the other fields on the page. In addition to the fields described earlier in this section, you can optionally add new values for existing property name/value pairs.
    - **Property Name**—Identifies the name of the custom or built-in device property.
    - **Property Value**—Identifies the value of the property.



**Note** To specify a CPE DHCP criteria for a DOCSIS modem for use with promiscuous computers, you must specify the property `/provisioning/cpeDhcpCriteria`. The value must be a valid DHCP criteria.

Use the `/IPDevice/reservation` property to reserve an IP address for a specific device.

---

- Step 3** Click **Submit** to add the device, or **Reset** to clear all fields.
- 

## Modifying Device Records

To modify a device record:

- 
- Step 1** From the Manage Devices page, click the Identifier link corresponding to the correct device. The Modify Device page appears.
  - Step 2** Enter the data in the correct field. You can modify any existing property name/value pairs by clicking **Add**, or delete any of them by clicking **Delete**.
  - Step 3** Click **Submit** to save the changes made to this device, or **Reset** to clear all fields.
-

## Deleting Devices

Deleting device records is a simple process, but one that you should use carefully. To undo the delete, you must restore a previously backed up database or re-add the device.



**Note** If restoration of a backed-up database becomes necessary, see [Database Restore, page 14-6](#).

To delete a device record:

- Step 1** From the Manage Devices page, locate the device that you want to delete. You can use one of the search types for this purpose.
- Step 2** Click the check box to the left of the correct device.
- Step 3** Click **Delete**. The device record stored in the RDU database is removed.

## Regenerating Device Configurations

The Regenerate button or API operation force immediate regeneration of configurations for the device. These configurations are sent to the DPEs in the device's provisioning group. Normally, the process of regenerating the configuration is automatically triggered following changes to device, Class of Service, or other such impacting changes. However, after a change to a Class of Service, the system takes time to regenerate configurations for all devices. This button can be used to expedite regeneration of configurations for a given device. This may be desirable during proactive troubleshooting.

It is sometimes necessary to change many different Class of Service or DHCP criteria parameters. When this happens, existing device configurations become stale and require regeneration of the configuration. To eliminate the need to manually regenerate each configuration, and reduce the potential for introducing errors, BAC provides a configuration regeneration service (CRS) that you can use to automatically regenerate all device configurations.

Device configurations are automatically regenerated whenever:

- A file related to a Class of Service, that is, a template, is updated.
- The default Class of Service or DHCP criteria for a device type is changed.
- A DHCP criteria property is changed.
- The provisioning group object is changed via the administrator user interface or the API.
- The Class of Service object properties are changed.
- The DPE sends a configuration regeneration request to the RDU.
- The device properties or relationship is updated.

Some configurations cannot be automatically regenerated because the BAC system cannot determine if the change impacts device configuration. In such cases, you must manually regenerate configurations by using the `generationConfiguration()` method or the administrator user interface. Configurations that must be manually regenerated are those that become necessary whenever:

- A technology default is changed.
- The system defaults are changed.
- A file that is included within another DOCSIS template is changed.

**Note**

---

Regardless of how configurations are regenerated, they are not propagated to the devices until the device configuration is activated, that is, the device contacts the DPE either on schedule or as a result of a connection request initiated from the DPE.

---

## Relating and Unrelating Devices

The concept of relating devices is somewhat similar to that of Class of Service or DHCP Criteria inasmuch as a device is related to a specific Class of Service or to a specific DHCP criteria. The significant difference is that the Class of Service and DHCP Criteria are considered to be predefined nodes and that you use nodes to group devices into arbitrary groups that you define.

In this context, the Relate function lets you associate a device, using its MAC address, to a specific node, which is in turn associated with a specific node type.

By relating a device to a specific node, information indicating that the device is related to a specific node is stored in the database. If you relate the device to the predefined **system:diagnostics** node, you can use available information to troubleshoot potential problems.

## Resetting Devices

The Reset button lets you reboot any selected device.

## Unregistering a Device

The unregister function lets you reset a device back to its defaults as if it had just booted on the network.

**Note**

---

If the device has never booted on the network, it will be deleted.

---

# Node Management

Node management allows the creation, modification, and deletion of nodes and node types. Within the context of BAC, node types can be considered as groups of nodes, while nodes themselves make up the node type.

## Managing Node Types

Access the Manage Nodes page (shown in [Figure 10-4](#)) by selecting Nodes from either the main menu or the primary menu bar. Node Type is the default setting when this page appears.

Figure 10-4 Manage Nodes Page

**Broadband Access Center for Cable** Logout

Configuration | Devices | **Nodes** | Servers | Users

User:admin Role:Administrator

---

**Manage Nodes**  
Use this page to manage nodes and node types (add, delete, modify or search).

Search Type  
Node Type ▾

Add

| Node Types                | Delete |
|---------------------------|--------|
| <a href="#">EST_Nodes</a> |        |
| <a href="#">Node_ABC</a>  |        |
| <a href="#">system</a>    |        |

Result Pages: 1

129906

## Adding a Node Type

To add a new node type:

- Step 1** Click **Add** and the Add Node Type page appears.
- Step 2** Enter a name for the new node type.
- Step 3** Select the appropriate Property Name from the drop-down list and enter the required Property Value.
- Step 4** Click **Add** and the new node type appears. You can continue adding as many properties as required.
- Step 5** Click **Submit** when complete. The new node type is recorded in the RDU and the Manage Node Types page appears with the new node type added.


## Modifying Node Types

To modify node type properties:

- Step 1** Click the desired node type and the Modify Node Type page appears.
- Step 2** Make the necessary changes to the Property Name/Property Value pairs. If you need to delete a specific pair, click **Delete** next to the desired pair.
- Step 3** Click **Submit** and the Manage Node page appears with the appropriately modified description.

## Deleting Node Types

To delete node types:

- 
- Step 1** In the Manager Node page click the **Delete** icon () corresponding to the desired node type.
  - Step 2** In the Delete Node Type dialog box, click **OK** to delete the selected node type, or **Cancel** to return to the previous page.
- The Manage Nodes page appears without the deleted Node Type.
- 

## Managing Nodes

You can create and modify nodes, and delete unwanted nodes.

### Adding a New Node

To add a new node:

- 
- Step 1** Select **Nodes** from the drop-down list on the Manage Nodes page.
  - Step 2** Click **Add** and the Add Node page appears.
  - Step 3** Enter the new node name and select the appropriate Node Type for this node.  
Click **Submit** when complete and the Manage Node page appears with the new node added.
  - Step 4** Select the appropriate Property Name from the drop-down list and enter the required Property Value.
  - Step 5** Click **Add** to increase the number of applicable Property Name/Property Value pairs.
  - Step 6** Click **Submit** when complete. The new node is recorded in the RDU and the Manage Nodes page appears with the new node added.
- 

### Modifying a Node

To modify node properties:

- 
- Step 1** Click the desired node and the Modify Nodes page appears.
  - Step 2** Make the necessary changes to the Property Name/Property Value pairs. If you need to delete a specific pair, click **Delete** next to the desired pair.
  - Step 3** Click **Submit** and the Manage Node page appears with the appropriately modified description.
- 

### Deleting Nodes

You can delete any node that appears in the Manager Node page by checking the box corresponding to the node being deleted and then click **Delete**. The node is immediately deleted.



## Relating/Unrelating Node Types to Nodes

The relate and unrelate functions are used to establish a relationship between specific node types and nodes. To either relate or unrelate this relationship:

- 
- Step 1** Click **Relate** or **Unrelate**, as desired, for the selected node. Either the Relate Nodes or Unrelate Node page appears.
  - Step 2** Select the appropriate **Node Type** from the drop-down list and select the group to which the node will be related/unrelated.
  - Step 3** Click **Submit**; the Manage Nodes page appears.
- 

## Viewing Node Details

Use the **Devices** tab to search for a specific node. Doing this displays the Node Search function on the Manage Devices page. From this page select the appropriate Node Type and enter the node name. You can use wildcard characters to locate a group of similarly named nodes. See [Searching for Devices](#), page 10-5, for additional information on search functions.

## Viewing Servers

This section describes the BAC server pages:


- [Viewing Device Provisioning Engines](#), page 10-19
- [Viewing Network Registrar Extension Points](#), page 10-23
- [Viewing Provisioning Groups](#), page 10-25
- [Viewing Regional Distribution Unit Details](#), page 10-25

## Viewing Device Provisioning Engines

The Manage Device Provisioning Engines page (**Servers > DPEs**) lets you monitor the list of all DPEs currently registered with the BAC database. Each DPE name that appears on this page is a link to another page that displays the details for that DPE. Click the DPE link to display the details page, which is similar to [Figure 10-5](#).

**Note**

The RDU determines the names of the Network Registrar extensions and DPEs by performing a reverse DNS lookup on the DPE interfaces through which the DPE contacts the RDU.

**Figure 10-5** View Device Provisioning Engines Details Page


**Broadband Access Center for Cable** Logout

Configuration | Devices | Nodes | **Servers** | Users  
 DPEs | NRs | Provisioning Groups | RDU  
 User:admin Role:Administrator

**CISCO SYSTEMS** View Device Provisioning Engines Details  
 Use this page to view the current values for the device provisioning engine that you selected.

| Device Provisioning Engine Details     |                                        |
|----------------------------------------|----------------------------------------|
| Host Name:                             | bac-test-u5-22.cisco.com               |
| Port:                                  | 49186                                  |
| IP Address:                            | 10.86.10.225                           |
| Primary Provisioning Group(s):         | <a href="#">default</a>                |
| Secondary Provisioning Group(s):       |                                        |
| PacketCable Enabled:                   | Yes                                    |
| CableHome Enabled:                     | Yes                                    |
| Properties:                            |                                        |
| Version:                               | BAC 2.7.1 (SOL_BAC2_7_1_00000000_0000) |
| UpTime:                                | 14 hour(s) 52 min(s) 55 sec(s)         |
| State:                                 | Ready                                  |
| Log Files                              |                                        |
| DPE Log File                           | <a href="#">🔗</a>                      |
| Cache Statistics                       |                                        |
| Hits:                                  | 0                                      |
| Misses:                                | 0                                      |
| Files:                                 | 0                                      |
| Configurations:                        | 0                                      |
| TFTP Statistics                        |                                        |
| Packets Received:                      | 0                                      |
| Packets Dropped:                       | 0                                      |
| Packets Successful:                    | 0                                      |
| Packets Failed:                        | 0                                      |
| Time Of Day Statistics                 |                                        |
| Packets Received:                      | 0                                      |
| Packets Dropped:                       | 0                                      |
| Packets Successful:                    | 0                                      |
| Packets Failed:                        | 0                                      |
| PacketCable SNMP Statistics            |                                        |
| SNMP Informs Successful:               | 0                                      |
| SNMP Sets Successful:                  | 0                                      |
| SNMP Configuration Informs Successful: | 0                                      |
| SNMP Configuration Informs Failed:     | 0                                      |
| PacketCable MTA Statistics             |                                        |
| MTA AP Requests Received:              | 0                                      |
| MTA AP Responses Sent:                 | 0                                      |
| PacketCable KDC Statistics             |                                        |
| KDC FQDN Requests Received:            | 0                                      |
| KDC FQDN Responses Sent:               | 0                                      |

2103965

**Note**

The details that appear on the administrator user interface for a hardware DPE and a Solaris DPE are identical.

Table 10-3 identifies the fields and buttons shown in Figure 10-5.

**Table 10-3** View Device Provisioning Engines Details Page

| Field or Button                           | Description                                                                      |
|-------------------------------------------|----------------------------------------------------------------------------------|
| <b>Device Provisioning Engine Details</b> |                                                                                  |
| Host Name                                 | Identifies the DPE hostname.                                                     |
| Port                                      | Identifies the DPE port number from which DPE established connection to the RDU. |
| IP Address                                | Identifies the IP address of the DPE.                                            |

**Table 10-3** View Device Provisioning Engines Details Page (continued)

| Field or Button                 | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Primary Provisioning Group(s)   | Identifies the primary provisioning groups that the selected DPE belongs to. This is an active link that, if clicked, displays the Provisioning Group Details page for that provisioning group.                                                                                                                                                                                                                                                                                                                                           |
| Secondary Provisioning Group(s) | Identifies the secondary provisioning group (provided that this DPE belongs to a secondary provisioning group) that the selected DPE belongs to.                                                                                                                                                                                                                                                                                                                                                                                          |
| PacketCable Enabled             | Identifies whether the PacketCable voice technology is enabled on this DPE.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| CableHome Enabled               | Identifies whether the home networking technology is enabled on this DPE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Properties                      | Identifies the properties configured for the DPE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Version                         | Identifies the version of DPE software currently in use.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Up Time                         | Specifies the total duration that the DPE has been operational since its last start-up.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| State                           | <p>Identifies whether the DPE is ready for operations. These states include:</p> <ul style="list-style-type: none"> <li>• Registering</li> <li>• Initializing</li> <li>• Synchronizing</li> <li>• Populating</li> <li>• Ready</li> <li>• Offline</li> </ul> <p>For details on each state, see <a href="#">DPE-RDU Synchronization, page 2-7</a>.</p> <p><b>Note</b> If this field reads Offline, the options from the Uptime field onwards do not appear. The DPE is prepared to service client requests in any state except Offline.</p> |

**Table 10-3** View Device Provisioning Engines Details Page (continued)

| Field or Button                       | Description                                                                                                                                                                                                                                                                               |
|---------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Log Files</b>                      |                                                                                                                                                                                                                                                                                           |
| DPE Log File                          | Features the View Details icon that if clicked displays the View Log File Contents page, which provides details of <i>dpe.log</i> .<br><br><b>Note</b> If you are running Broadband Access Center for Cable 2.6.x, you cannot view the DPE log file via the administrator user interface. |
| <b>Cache Statistics</b>               |                                                                                                                                                                                                                                                                                           |
| Hits                                  | Identifies the number of cache hits that occurred since the last time the DPE was started.                                                                                                                                                                                                |
| Misses                                | Identifies the number of cache misses that occurred since the last time the DPE was started.                                                                                                                                                                                              |
| Files                                 | Identifies the number of cache files that are currently stored in the DPE.                                                                                                                                                                                                                |
| Configurations                        | Identifies how many device configuration files are saved in cache.                                                                                                                                                                                                                        |
| <b>TFTP Statistics</b>                |                                                                                                                                                                                                                                                                                           |
| Packets Received                      | Identifies the number of TFTP packets that were received by the selected DPE.                                                                                                                                                                                                             |
| Packets Dropped                       | Identifies the number of TFTP packets that were dropped due to the DPE being overloaded.                                                                                                                                                                                                  |
| Packets Successful                    | Identifies the number of TFTP packets that were transmitted successfully.                                                                                                                                                                                                                 |
| Packets Failed                        | Identifies the number of TFTP packets that failed during transmission.                                                                                                                                                                                                                    |
| <b>Time of Day Statistics</b>         |                                                                                                                                                                                                                                                                                           |
| Packets Received                      | Identifies the number of Time of Day packets that were received by the selected DPE.                                                                                                                                                                                                      |
| Packets Dropped                       | Identifies the number of Time of Day packets that were dropped due to the DPE being overloaded.                                                                                                                                                                                           |
| Packets Successful                    | Identifies the number of Time of Day packets that were transmitted successfully.                                                                                                                                                                                                          |
| Packets Failed                        | Identifies the number of Time of Day packets that failed during transmission.                                                                                                                                                                                                             |
| <b>PacketCable SNMP Statistics</b>    |                                                                                                                                                                                                                                                                                           |
| SNMP Informs Successful               | Identifies the number of inform requests that were successfully sent.                                                                                                                                                                                                                     |
| SNMP Sets Successful                  | Identifies the number of successful SNMP sets.                                                                                                                                                                                                                                            |
| SNMP Configuration Informs Successful | Identifies the number of SNMP informs received from PacketCable MTAs indicating that they were successfully provisioned.                                                                                                                                                                  |
| SNMP Configuration Informs Failed     | Identifies the number of SNMP informs received from PacketCable MTAs indicating that they failed to be provisioned.                                                                                                                                                                       |

**Table 10-3** View Device Provisioning Engines Details Page (continued)

| Field or Button                   | Description                                                               |
|-----------------------------------|---------------------------------------------------------------------------|
| <b>PacketCable MTA Statistics</b> |                                                                           |
| MTA AP Requests Received          | Specifies the number of AP-REQ messages received by the DPE from the MTA. |
| MTA AP Responses Sent             | Specifies the number of AP-REP messages sent by the DPE to the MTA.       |
| <b>PacketCable KDC Statistics</b> |                                                                           |
| KDC FQDN Requests Received        | Specifies the number of FQDN-REQ messages sent by the KDC to the DPE.     |
| KDC FQDN Responses Sent           | Specifies the number of FQDN-REP messages sent by the DPE to the KDC.     |

## Viewing Network Registrar Extension Points

The Manage Network Registrar Extension Points page (**Servers > NRs**) lists the extension points for all Network Registrar servers that have been registered with the RDU, and are configured for use with BAC. Network Registrar servers automatically register with the RDU when those servers are started.

Each Network Registrar extension points that appears on this page is a link to a secondary page that displays details of that extension point. Click the Network Registrar extension point link to display the details page, which is similar to [Figure 10-6](#).

**Figure 10-6** View Network Registrar Extension Point Details Page

The screenshot shows the BACC administrator interface. The top navigation bar includes 'Configuration', 'Devices', 'Nodes', 'Servers', and 'Users'. The 'Servers' tab is active, and the 'NRs' sub-tab is selected. The user is identified as 'admin' with the role of 'Administrator'. The main content area is titled 'View Network Registrar Extension Point Details' and includes a Cisco Systems logo. Below the title, there is a brief instruction: 'Use this page to view the current values for the Network Registrar extension point that you selected.' The details are organized into three sections:

| Network Registrar Extension Point Details    |                                                  |
|----------------------------------------------|--------------------------------------------------|
| Host Name:                                   | bac-test-u5-22.cisco.com                         |
| IP Address:                                  | 10.24.250.45                                     |
| Provisioning Group:                          | default                                          |
| PacketCable Enabled:                         | Yes                                              |
| CableHome Enabled:                           | Yes                                              |
| Properties:                                  |                                                  |
| Version:                                     | BPR 2.7 (bacc-271-Solaris-bac2_7_1-000000000000) |
| UpTime:                                      | 15 hour(s) 25 min(s) 35 sec(s)                   |
| State:                                       | Ready                                            |
| Network Registrar Extension Point Statistics |                                                  |
| Packets Received:                            | 0                                                |
| Packets Ignored:                             | 0                                                |
| Packets Dropped:                             | 0                                                |
| Packets Successful:                          | 0                                                |
| Packets Failed:                              | 0                                                |
| Device Provisioning Engine(s) Details        |                                                  |
| DPE:                                         | bac-test-u5-22.cisco.com (10.86.10.225)          |
| Port:                                        | 49186                                            |
| Type:                                        | Primary Device Provisioning Engine               |
| Status:                                      | Ready Overloaded                                 |

[Table 10-4](#) identifies the fields and buttons shown in [Figure 10-6](#).

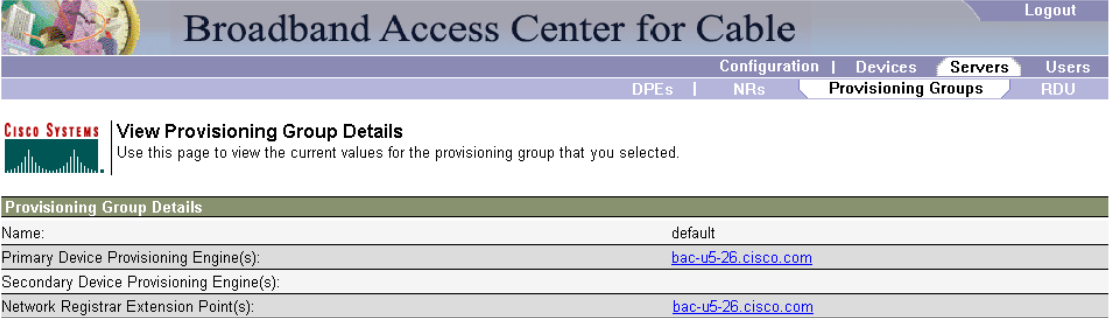
**Table 10-4** View Network Registrar Extension Point Details Page

| Field or Button                                                                                       | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Network Registrar Extension Point Details</b>                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Host Name                                                                                             | Displays the hostname of the system running Network Registrar.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| IP Address                                                                                            | Identifies the IP address of the Network Registrar server.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Provisioning Group                                                                                    | Identifies the provisioning group for the Network Registrar server. This is an active link that, if clicked, displays the Provisioning Group Details page for that provisioning group.                                                                                                                                                                                                                                                                                                                                                    |
| PacketCable Enabled                                                                                   | Identifies whether PacketCable voice technology is enabled.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Properties                                                                                            | Identifies the properties that are applied to the Network Registrar server.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Version                                                                                               | Identifies the extension point software currently in use.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Up Time                                                                                               | Specifies the total time that the Network Registrar extension point has been operational since its last start-up. This is indicated in hours, minutes, and seconds.                                                                                                                                                                                                                                                                                                                                                                       |
| State                                                                                                 | <p>Identifies whether the DPE is ready for operations. These states include:</p> <ul style="list-style-type: none"> <li>• Registering</li> <li>• Initializing</li> <li>• Synchronizing</li> <li>• Populating</li> <li>• Ready</li> <li>• Offline</li> </ul> <p>For details on each state, see <a href="#">DPE-RDU Synchronization, page 2-7</a>.</p> <p><b>Note</b> If this field reads Offline, the options from the Uptime field onwards do not appear. The DPE is prepared to service client requests in any state except Offline.</p> |
| <b>Network Registrar Extension Point Statistics</b>                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Packets Received                                                                                      | Identifies the number of packets that were received.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| Packets Ignored                                                                                       | Identifies the number of packets that were ignored.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Packets Dropped                                                                                       | Identifies the number of packets that were dropped.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Packets Successful                                                                                    | Identifies the number of packets that transferred successfully.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Packets Failed                                                                                        | Identifies the number of packets that failed to be transferred.                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Device Provisioning Engine(s) Details</b>                                                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Note</b> The following fields appear for each DPE that connects with the Network Registrar server. |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| DPE                                                                                                   | Identifies the IP address of the DPE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Port                                                                                                  | Identifies the port number from which DPE established connection to the RDU.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Type                                                                                                  | Identifies whether this DPE is a primary or secondary DPE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Status                                                                                                | Identifies whether the DPE is operational.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

## Viewing Provisioning Groups

The Manage Provisioning Groups page (**Servers > Provisioning Groups**) lets you monitor all current provisioning groups. Each provisioning group appearing in this list is a link to its own details page. Click this link to display the details page, which is similar to [Figure 10-7](#).

**Figure 10-7** View Provisioning Group Details Page



The screenshot shows the 'View Provisioning Group Details' page in the Broadband Access Center for Cable. The page header includes the Cisco Systems logo and the title 'View Provisioning Group Details'. Below the header, there is a navigation menu with 'Servers' selected. The main content area displays a table with the following details:

| Provisioning Group Details               |                                     |
|------------------------------------------|-------------------------------------|
| Name:                                    | default                             |
| Primary Device Provisioning Engine(s):   | <a href="#">bac-u5-26.cisco.com</a> |
| Secondary Device Provisioning Engine(s): |                                     |
| Network Registrar Extension Point(s):    | <a href="#">bac-u5-26.cisco.com</a> |

101573

[Table 10-5](#) identifies the fields and buttons shown in [Figure 10-7](#). The fields described in [Table 10-5](#) may include active links that, if clicked, display the appropriate details page.

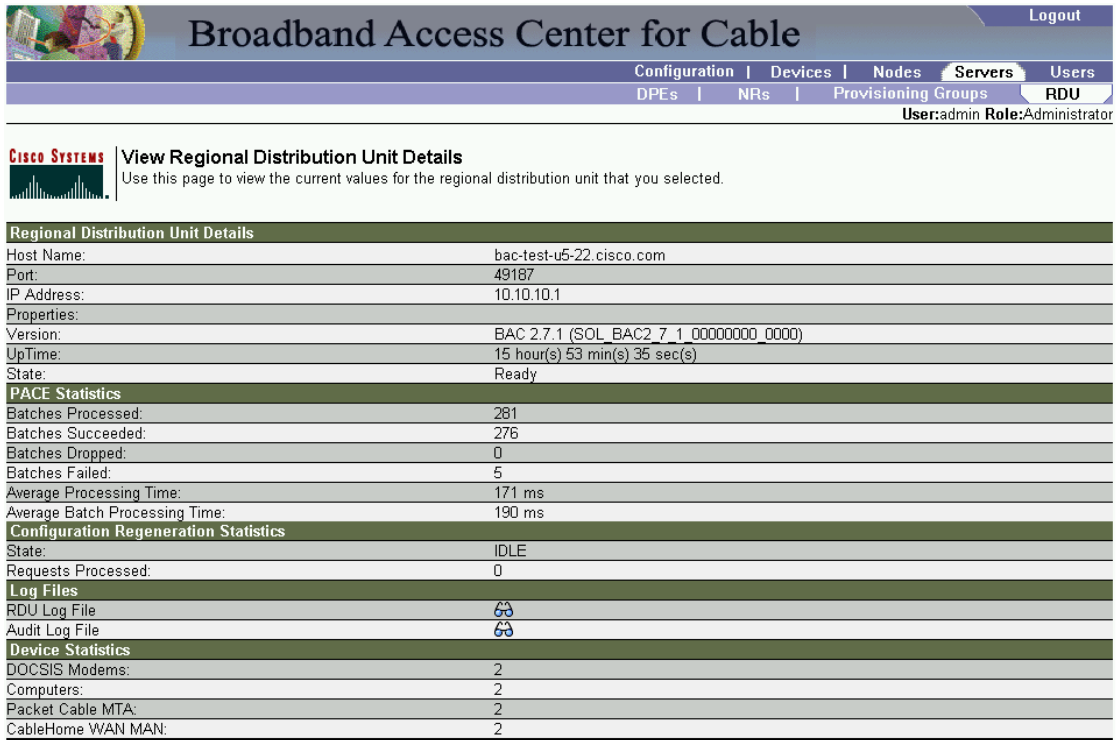
**Table 10-5** View Provisioning Groups Details Page

| Field or Button                         | Description                                                                                  |
|-----------------------------------------|----------------------------------------------------------------------------------------------|
| Name                                    | Identifies the provisioning group name selected from the Manage Provisioning Groups page.    |
| Primary Device Provisioning Engine(s)   | Identifies the hostnames of the DPEs that are primary for this provisioning group.           |
| Secondary Device Provisioning Engine(s) | Identifies the hostnames of the DPEs that are secondary for this provisioning group.         |
| Network Registrar Extension Points      | Identifies the hostname of the Network Registrar server assigned to this provisioning group. |

## Viewing Regional Distribution Unit Details

The RDU option, from the Servers menu, displays details of the RDU. [Figure 10-8](#) illustrates a sample RDU details page.

Figure 10-8 View Regional Distribution Unit Details Page



**Broadband Access Center for Cable** Logout

Configuration | Devices | Nodes | **Servers** | Users  
DPEs | NRs | Provisioning Groups | **RDU**

User:admin Role:Administrator

---

**CISCO SYSTEMS** View Regional Distribution Unit Details  
Use this page to view the current values for the regional distribution unit that you selected.

| Regional Distribution Unit Details           |                                        |
|----------------------------------------------|----------------------------------------|
| Host Name:                                   | bac-test-u5-22.cisco.com               |
| Port:                                        | 49187                                  |
| IP Address:                                  | 10.10.10.1                             |
| <b>Properties:</b>                           |                                        |
| Version:                                     | BAC 2.7.1 (SOL_BAC2_7_1_00000000_0000) |
| UpTime:                                      | 15 hour(s) 53 min(s) 35 sec(s)         |
| State:                                       | Ready                                  |
| <b>PACE Statistics</b>                       |                                        |
| Batches Processed:                           | 281                                    |
| Batches Succeeded:                           | 276                                    |
| Batches Dropped:                             | 0                                      |
| Batches Failed:                              | 5                                      |
| Average Processing Time:                     | 171 ms                                 |
| Average Batch Processing Time:               | 190 ms                                 |
| <b>Configuration Regeneration Statistics</b> |                                        |
| State:                                       | IDLE                                   |
| Requests Processed:                          | 0                                      |
| <b>Log Files</b>                             |                                        |
| RDU Log File                                 | <a href="#">📄</a>                      |
| Audit Log File                               | <a href="#">📄</a>                      |
| <b>Device Statistics</b>                     |                                        |
| DOCSIS Modems:                               | 2                                      |
| Computers:                                   | 2                                      |
| Packet Cable MTA:                            | 2                                      |
| CableHome WAN MAN:                           | 2                                      |

210864

Table 10-6 identifies the fields and buttons shown in Figure 10-8.

Table 10-6 View Regional Distribution Unit Details Page

| Field or Button                           | Description                                                                                                                                                               |
|-------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Regional Distribution Unit Details</b> |                                                                                                                                                                           |
| Host Name                                 | Identifies the hostname of the system that is running the RDU.                                                                                                            |
| Port                                      | Identifies the RDU listening port number for connections from DPEs. The default port number is 49187, but you can select a different port number during RDU installation. |
| IP Address                                | Identifies the IP address assigned to the RDU.                                                                                                                            |
| Properties                                | Identifies the properties configured for the RDU.                                                                                                                         |
| Version                                   | Specifies the version of RDU software currently in use.                                                                                                                   |
| Up Time                                   | Specifies the total time that the RDU has been operational since its last period of downtime.                                                                             |
| State                                     | Identifies whether the RDU is ready to respond to requests. The only state visible on the administrator user interface is Ready.                                          |



**Table 10-6** View Regional Distribution Unit Details Page (continued)

| Field or Button                              | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>PACE Statistics</b>                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Batches Processed                            | Identifies how many individual batches have been processed since the last RDU start-up.                                                                                                                                                                                                                                                                                                                                                                                                     |
| Batches Succeeded                            | Identifies how many individual batches have been successfully processed since the last RDU start-up.                                                                                                                                                                                                                                                                                                                                                                                        |
| Batches Dropped                              | Identifies how many batches have been dropped since the last RDU start-up.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Batches Failed                               | Identifies how many batches have failed processing since the last RDU start-up.                                                                                                                                                                                                                                                                                                                                                                                                             |
| Average Processing Time                      | Identifies the average time, in milliseconds, that it takes to process the batch excluding the time it spends in the queue if RDU is too busy.                                                                                                                                                                                                                                                                                                                                              |
| Average Batch Processing Time                | Identifies the average time, in milliseconds, that it takes to process the batch including the time it spends in the queue if RDU is too busy.                                                                                                                                                                                                                                                                                                                                              |
| <b>Configuration Regeneration Statistics</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| State                                        | Identifies the operational state of the configuration generation service. This could be: <ul style="list-style-type: none"> <li>• Idle—Specifies that the CRS is not processing regeneration requests.</li> <li>• Regeneration—Specifies that the CRS is processing regeneration requests.</li> <li>• Waiting Regeneration—Specifies that the CRS is unable to regenerate configurations for a device. When the CRS is stuck in this state, check the <i>rdulog</i> for details.</li> </ul> |
| Requests Processed                           | Identifies the number of configuration regeneration requests processed since the last RDU start-up.                                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Log Files</b>                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| RDU Log File                                 | Features the View Details icon, that, if clicked, displays the View Log File Contents page, which provides details of the <i>rdulog</i> file.                                                                                                                                                                                                                                                                                                                                               |
| Audit Log File                               | Features the View Details icon, that, if clicked, displays the View Log File Contents page, which provides details of the <i>auditlog</i> file.                                                                                                                                                                                                                                                                                                                                             |
| <b>Device Statistics</b>                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Note</b>                                  | The Device Statistics section appears only when the appropriate devices are present.                                                                                                                                                                                                                                                                                                                                                                                                        |
|                                              | Identifies the number of devices in the RDU database. The information presented in this area depends on the technologies licensed and configured. These devices may include: <ul style="list-style-type: none"> <li>• DOCSIS Modems</li> <li>• Computers</li> <li>• ATA 186 and 188 devices</li> <li>• PacketCable MTAs</li> <li>• CableHome WAN-Data/WAN-MAN devices</li> </ul>                                                                                                            |
| <b>Note</b>                                  | If external Jar files are installed, detailed information on the installed extension Jar files and the loaded extension class files appears after the Device Statistics section.                                                                                                                                                                                                                                                                                                            |





# CHAPTER 11

## Configuring Broadband Access Center

---

This chapter describes the Broadband Access Center (BAC) configuration tasks that you perform by selecting the options in the Configuration menu:

- [Configuring Class of Service, page 11-1](#)
- [Configuring Custom Properties, page 11-5](#)
- [Configuring Defaults, page 11-6](#)
- [Configuring DHCP Criteria, page 11-24](#)
- [Managing External Files, page 11-26](#)
- [Managing License Keys, page 11-30](#)
- [Managing RDU Extensions, page 11-32](#)
- [Publishing Provisioning Data, page 11-35](#)
- [Configuring SRV Records in the Network Registrar DNS Server, page 11-36](#)
- [Configuring SNMPv3 Cloning on the RDU and DPE for Secure Communication with PacketCable MTAs, page 11-37](#)
- [Automatic FQDN Generation, page 11-38](#)

### Configuring Class of Service

By using the BAC administrator user interface, you can configure the Class of Service offered to your customers. For example, you can associate DOCSIS options with different DOCSIS Class of Service. You use the BAC administrator user interface to add, modify, view, or delete any selected Class of Service.

[Figure 11-1](#) describes the Manage Class of Service page.

Figure 11-1 Manage Class of Service Page

The screenshot shows the 'Manage Class of Service' page in the Broadband Access Center for Cable. The page header includes the Cisco Systems logo and the title 'Manage Class of Service'. Below the header, there is a search bar for 'Class of Service' with a dropdown menu currently showing 'DOCSISModem'. An 'Add' button is located below the search bar. A table lists the existing classes of service, each with a delete icon. The table has columns for 'Class of Service' and 'Delete'. The classes listed are 'sample-bronze-docsis', 'sample-gold-docsis', 'sample-silver-docsis', and 'unprovisioned-docsis'. At the bottom left, it says 'Result Pages: 1'. At the bottom right, there is a vertical text '179882'.

Table 11-1 identifies the fields and buttons shown in Figure 11-1.

Table 11-1 Manage Class of Service Page

| Field or Button         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Class of Service</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| Class of Service        | <p>A drop-down list that identifies the technology Class of Service that you can search for. Available options are:</p> <ul style="list-style-type: none"> <li>• ATA 186</li> <li>• ATA 188</li> <li>• CableHome WAN-Data</li> <li>• CableHome WAN-MAN</li> <li>• Computer</li> <li>• DOCSIS Modem</li> <li>• PacketCable Media Terminal Adapter (MTA)</li> </ul> <p><b>Note</b> For additional information on these areas of technology, see <a href="#">Configuring Defaults</a>, page 11-6.</p> |
| <b>Add</b>              | Lets you add a new Class of Service.                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Class of Service list   | Displays the names of Class of Service objects.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Delete</b>           | Lets you delete the selected Class of Service.                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

## Adding a Class of Service

To add a specific Class of Service:

**Step 1** Choose **Configuration > Class of Service**.

**Step 2** Click **Add**. The Add Class of Service page appears. This page identifies the various settings for the selected Class of Service.

**Step 3** Enter the name of your new Class of Service.

**Step 4** Choose a **Class of Service Type**.

For example, assume that you want to create a new Class of Service called Gold-Classic for DOCSIS modems. You might enter **Gold-Classic** as the Class of Service Name, and choose **DOCSIS** from the service type drop-down list.

**Step 5** Enter a **Property Name** and **Property Value** in the appropriate fields.

For example, choose as the property name `/cos/docsis/file`. Enter **Gold-Classic.cm** in the Property Value field, and continue with the rest of this procedure.

Multiple Property Name:Property Value pairs could appear on this page. You use the **Delete** button to remove any unwanted pairs from the Class of Service.



**Note**

When adding a DOCSISModem Class of Service, you must specify the `/cos/docsis/file` property with the value being the name of a previously added external file. This file is used when provisioning a DOCSIS device that has this Class of Service.

BAC provides automatic selection of a cable modem configuration file that enables the highest DOCSIS version compatible with the modem. To enable this feature, you must configure the Class of Service with multiple configuration files, one for each DOCSIS level. Use the following properties to allow the selection of a configuration file specific to a DOCSIS version:

- `/cos/docsis/file/1.0`—Selects a configuration file specific to DOCSIS 1.0.
- `/cos/docsis/file/1.1`—Selects a configuration file specific to DOCSIS 1.1.
- `/cos/docsis/file/2.0`—Selects a configuration file specific to DOCSIS 2.0.

When adding a PacketCable Class of Service, you must specify the `/cos/packetCableMTA/file` property with the value being the name of a previously added external file. This file is used when provisioning a PacketCable device that has this Class of Service.

When adding a CableHome WAN-MAN Class of Service, you must specify the `/cos/cableHomeWanMan/file` property with the value being the name of a previously added external file. This file is used when provisioning a CableHome WAN-MAN device that has this Class of Service.

**Step 6** Click **Add** to add the property to the defining Class of Service.

**Step 7** Click **Submit** to finalize the process or **Reset** to return all fields to their previous setting.

After submitting the Class of Service, the Manage Class of Service page appears to show the newly added Class of Service for that particular device type.

## Modifying a Class of Service

You modify your Class of Service by selecting the various properties and assigning appropriate property values. When creating a Class of Service for the first time you must select all the required properties and assign values to them. If you make a mistake, or your business requirements for a certain Class of Service change, you can either change the property value before submitting your previous changes or delete the Property Name:Property Value pair altogether.

**Note**

---

Changes to the Class of Service object trigger the Configuration Regeneration Service (CRS) to regenerate configurations for all affected devices and send configurations to the DPEs. The CRS performs this task as a background job.

You can view the status of the CRS from the View RDU Details page.

---

To add, delete, or modify Class of Service properties:

---

- Step 1** Choose **Configuration > Class of Service**.
- Step 2** Choose the Class of Service to be modified.
- Step 3** Click the link corresponding to the correct Class of Service. The Modify Class of Service page appears; note that the selected Class of Service name and type appear below the page description.
  - To add a new property to the selected Class of Service:
    - Select the first property that you want assigned to the selected Class of Service, from the Property Name drop-down list and then, after choosing the appropriate value for that property, click **Add**.
    - Repeat for any other properties you want to assign to the selected Class of Service.
  - To delete a property for the selected Class of Service:
    - Locate the unwanted property in the list immediately above the Property Name drop-down list.
    - Click **Delete**.
  - To modify the value currently assigned to a property:
    - Delete the appropriate property as described above.
    - Add the same property back to the Class of Service while entering the new Property Value.

**Note**

---

If you delete a property that is required for your business process, you must add it back, and select the appropriate value, before you submit the change.

---

- Step 4** Click **Submit** to make the modifications to the Class of Service. Each property added to a Class of Service appears when you click **Submit**. After doing so, a confirmation page appears to regenerate the configurations for the devices with the selected Class of Service.
  - Step 5** Click **OK**. The modified Class of Service will be available in the Manage Class of Service page.
-

## Deleting a Class of Service

You can delete any existing Class of Service, but before you attempt to do so, you must ensure that there are no devices associated with that Class of Service.

**Tip**


When large numbers of devices associated with a Class of Service need to be deleted, use the BAC application programming interface (API) to write a program to iterate through these devices to reassign another Class of Service to the devices.

To delete a Class of Service:

**Step 1**

Choose **Configuration > Class of Service**.

**Step 2**

Click the **Delete** icon () for the correct Class of Service, and a confirmation dialog box appears.

**Note**

A Class of Service cannot be deleted if devices are associated with it or if it is designated as the default Class of Service. Therefore, you cannot delete the **unprovisioned-docsis** Class of Service object.

**Step 3**

Click **OK** to delete the file, or **Cancel** to return to the Manage Class of Service page (see [Figure 11-1](#)).

If you try to delete a Class of Service with devices associated with it, this error message appears:

```
The following error(s) occurred while processing your request.
```

```
Error: Class Of Service [sample-COS] has devices associated with it, unable to delete
```

```
Please correct the error(s) and resubmit your request.
```

The specific Class of Service is specified within the error message. In this example, this information is represented by *sample-COS*.

## Configuring Custom Properties

Custom properties let you specify additional customizable device information to be stored in the RDU database. The Manage BAC Custom Properties configuration page is found under the Configuration menu. You use this page to add or delete custom properties.

**Caution**

Although you can delete custom properties if they are currently in use, doing so could cause extreme difficulty to other areas where the properties are in use.

After the custom property is defined, you can use it in this property hierarchy. Properties can be configured on the following objects for use in the property hierarchy:

- Device
- Provisioning Group
- Class of Service

- Device Type
- System defaults

Additionally, properties can be configured on Node and Node Type objects, but they will not be part of the property hierarchy.

To configure custom properties:

---

**Step 1** Choose **Configuration** on the Primary Navigation bar.

**Step 2** Choose **Custom Property** on the Secondary Navigation bar.

The Manage BAC Custom Properties page appears.

- To add a custom property:
  - Click **Add** on the Manage BAC Custom Properties page, and the Add Custom Property page appears.
  - Enter the name of the new custom property.
  - Choose a custom property type from the drop-down list.
  - Click **Submit** when complete.

After the property has been added to the database, the Manage BAC Custom Properties page appears.

- To delete a custom property:
    - Identify the custom property to be deleted from the Manage BAC Custom Properties page.
    - Click the **Delete** icon corresponding to the desired custom property, and the dialog box for deleting custom properties appears.
    - Click **OK** to delete the custom property.
- 

## Configuring Defaults

The Configure Defaults page, found under the Configuration option, lets you access the default settings for the overall system, including the Regional Distribution Unit (RDU), Network Registration extensions, and all supported technologies.

## Selecting Configuration Options

The procedure for configuring specific default types is identical. Complete this procedure to access the defaults page and then refer to the appropriate section for a description of the various page components.

---

**Step 1** Choose **Configuration** on the Primary Navigation bar or Main Menu page.

**Step 2** Choose **Defaults** from the Secondary Navigation bar. The Configure Defaults page appears.

**Step 3** Click the specific default link from the Default links on the left of the screen. The appropriate defaults page appears.

---



# ATA 186 Defaults

The Cisco ATA 186 is a handset-to-Ethernet adaptor that turns a traditional telephone into an Ethernet IP telephone. You can take advantage of the many IP telephony applications by connecting an existing analog telephone to this device.

The ATA 186 Defaults page displays a list of default values currently available to support the ATA 186. See Figure 11-2.

**Figure 11-2** Configure Defaults–ATA 186 Defaults Page

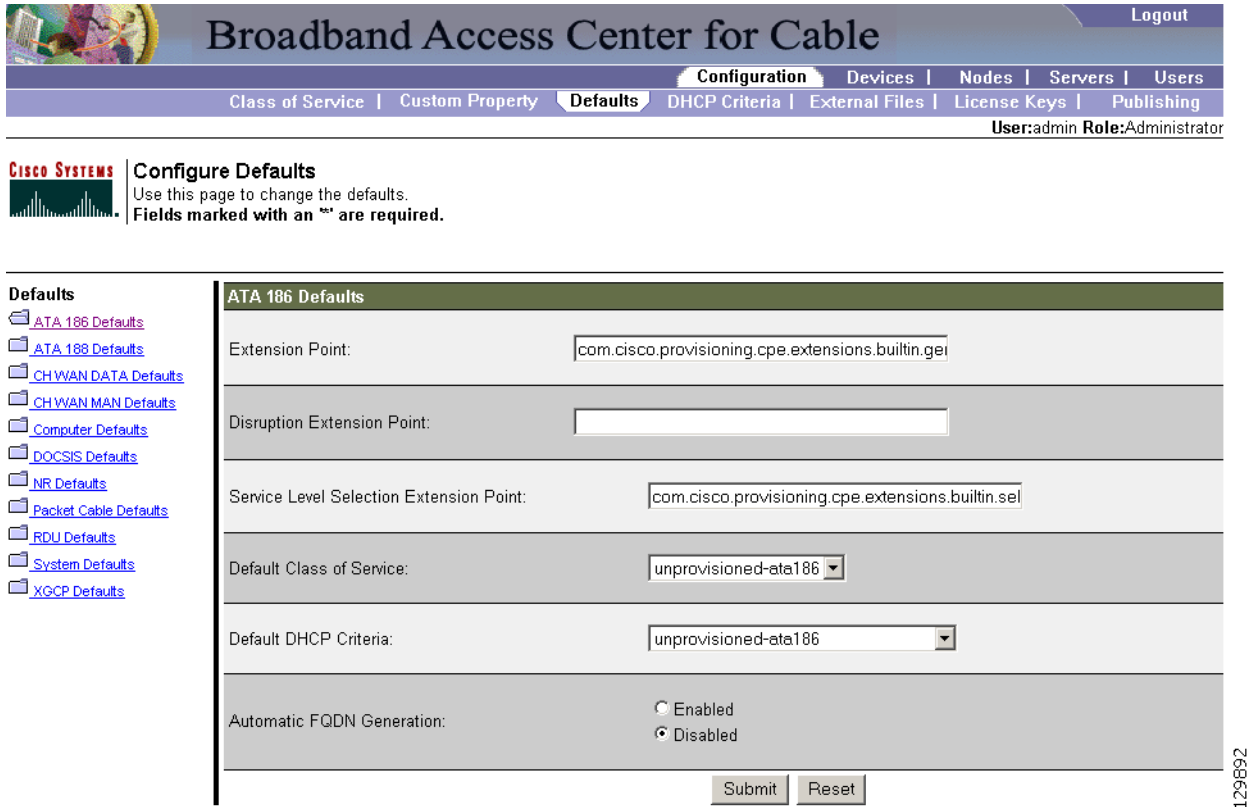


Table 11-2 identifies the fields and buttons shown in Figure 11-2. In many cases, the parameters that appear on this page also appear in other default pages.

**Table 11-2** Configure Defaults–ATA 186 Defaults Page

| Field or Button                         | Description                                                                                                |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------|
| Extension Point                         | Identifies the extension point to execute when generating a configuration for a device of this technology. |
| Disruption Extension Point              | Identifies the extension point to be executed to disrupt a device of this technology.                      |
| Service Level Selection Extension Point | Identifies the extension used to determine the DHCP criteria and Class of Service required for a device.   |

**Table 11-2** *Configure Defaults–ATA 186 Defaults Page (continued)*

| Field or Button           | Description (continued)                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Default Class of Service  | Identifies the current default Class of Service for a specific device technology, in this example, ATA186. New, unrecognized devices of that technology type will be assigned to this Class of Service. Use the drop-down list to select a new default value.                                                                                                                         |
| Default DHCP Criteria     | Identifies the current default DHCP criteria for a specific device technology, in this example, ATA186. New, unrecognized devices of that technology type will have this default DHCP criteria assigned. Use the drop-down list to select a new default value.                                                                                                                        |
| Automatic FQDN Generation | Automatically generates a host and domain name for the device. Two selectable options are available: <ul style="list-style-type: none"> <li>Enabled—Automatic generation of the FQDN is enabled.</li> <li>Disabled—Automatic generation of the FQDN is disabled.</li> </ul> <p><b>Note</b> See <a href="#">Automatic FQDN Generation, page 11-38</a>, for additional information.</p> |
| Submit                    | Activates the changes you have made. After the administrative database has been updated the Configure Defaults page will reflect the changes you have made.                                                                                                                                                                                                                           |
| Reset                     | Returns all settings to their previous setting.                                                                                                                                                                                                                                                                                                                                       |

## ATA 188 Defaults

The Cisco ATA 188 interfaces regular telephones with IP-based ethernet telephony networks. The ATA 188 provides true, next-generation VoIP terminations to support the needs of the enterprise, small-office environments, and emerging VoIP managed voice services and local services market.

The Configure ATA 188 Defaults page displays a list of default values currently available to support the ATA 188. The default parameters displayed for the ATA 188 are identical to those displayed for the ATA 186, although the values you select could be different.

## CableHome WAN Defaults

There are two distinct CableHome WAN default screens: one for WAN-Data devices and one for WAN-MAN devices. In either case, select the desired defaults from the list on the left pane. Each WAN default page contains identical fields and buttons as described in [Table 11-3](#).

**Table 11-3** *Configure Defaults–CH WAN-Data/CH WAN-MAN Defaults Page*

| Field or Button                         | Description                                                                                              |
|-----------------------------------------|----------------------------------------------------------------------------------------------------------|
| Extension Point                         | Identifies the extension point to execute when generating a configuration for a WAN device.              |
| Disruption Extension Point              | Identifies the extension point to be executed to disrupt a WAN device.                                   |
| Service Level Selection Extension Point | Identifies the extension used to determine the DHCP criteria and Class of Service required for a device. |

**Table 11-3** *Configure Defaults—CH WAN-Data/CH WAN-MAN Defaults Page (continued)*

| <b>Field or Button</b>    | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Default Class of Service  | Identifies the current default Class of Service for a WAN-Data. New, unrecognized WAN devices are assigned to this Class of Service. Use the drop-down list to select a new default value.                                                                                                                                                                                         |
| Default DHCP Criteria     | Identifies the current default DHCP criteria for a specific device technology. New, unrecognized WAN devices are assigned this default DHCP criteria. Use the drop-down list to select a new default value.                                                                                                                                                                        |
| Automatic FQDN Generation | Automatically generates a host and domain name for the device. Two selectable options are available: <ul style="list-style-type: none"> <li>• Enabled—Automatic generation of the FQDN is enabled.</li> <li>• Disabled—Automated FQDN generation is disabled.</li> </ul> <p><b>Note</b> See <a href="#">Automatic FQDN Generation, page 11-38</a>, for additional information.</p> |
| <b>Submit</b>             | Activates the changes you have made. After the administrative database has been updated the Configure Defaults page will reflect the changes you have made.                                                                                                                                                                                                                        |
| <b>Reset</b>              | Returns all settings to their previous setting.                                                                                                                                                                                                                                                                                                                                    |

## CableHome WAN-Data Defaults

When you select the CH WAN-Data Defaults link, the CableHome WAN-Data Defaults page appears. See [Figure 11-3](#). Use this page to configure the WAN-Data device.

**Figure 11-3** Configure Defaults—CableHome WAN-Data Defaults Page

**Broadband Access Center for Cable** Logout

Configuration | Devices | Nodes | Servers | Users

Class of Service | Custom Property | **Defaults** | DHCP Criteria | External Files | License Keys | Publishing

User:admin Role:Administrator

---

**CISCO SYSTEMS** **Configure Defaults**  
Use this page to change the defaults.  
Fields marked with an \* are required.

**Defaults**

- [ATA 186 Defaults](#)
- [ATA 188 Defaults](#)
- [CH WAN DATA Defaults](#)
- [CH WAN MAN Defaults](#)
- [Computer Defaults](#)
- [DOCSIS Defaults](#)
- [NR Defaults](#)
- [Packet Cable Defaults](#)
- [RDU Defaults](#)
- [System Defaults](#)
- [XGCP Defaults](#)

**CableHome WAN DATA Defaults**

|                                          |                                                                                |
|------------------------------------------|--------------------------------------------------------------------------------|
| Extension Point:                         | <input type="text" value="com.cisco.provisioning.cpe.extensions.builtin.gei"/> |
| Disruption Extension Point:              | <input type="text"/>                                                           |
| Service Level Selection Extension Point: | <input type="text" value="com.cisco.provisioning.cpe.extensions.builtin.sel"/> |
| Default Class of Service:                | <input type="text" value="unprovisioned-cablehome-wan-data"/> ▼                |
| Default DHCP Criteria:                   | <input type="text" value="unprovisioned-cablehome-wan-data"/> ▼                |
| Automatic FQDN Generation:               | <input type="radio"/> Enabled<br><input checked="" type="radio"/> Disabled     |

179963

## CableHome WAN-MAN Defaults

When you select the CH WAN-MAN Defaults link, the CableHome WAN-MAN Defaults page appears. See [Figure 11-4](#). Use this page to configure the WAN-MAN device type.

**Figure 11-4** Configure Defaults—CableHome WAN-MAN Defaults Page

**Broadband Access Center for Cable** Logout

[Configuration](#) | [Devices](#) | [Nodes](#) | [Servers](#) | [Users](#)  
[Class of Service](#) | [Custom Property](#) | **Defaults** | [DHCP Criteria](#) | [External Files](#) | [License Keys](#) | [Publishing](#)

User:admin Role:Administrator

---

**CISCO SYSTEMS** **Configure Defaults**  
 Use this page to change the defaults.  
 Fields marked with an \* are required.

**Defaults**

- [ATA 186 Defaults](#)
- [ATA 188 Defaults](#)
- [CH WAN DATA Defaults](#)
- [CH WAN MAN Defaults](#)
- [Computer Defaults](#)
- [DOCSIS Defaults](#)
- [NR Defaults](#)
- [Packet Cable Defaults](#)
- [RDU Defaults](#)
- [System Defaults](#)
- [XGCP Defaults](#)

**CableHome WAN MAN Defaults**

Extension Point:

Disruption Extension Point:

Service Level Selection Extension Point:

Default Class of Service:

Default DHCP Criteria:

Automatic FQDN Generation:  Enabled  Disabled

129894

## Computer Defaults

The Computer Defaults page displays a list of default values currently applied to the computers supported by BAC. See [Figure 11-5](#).

**Figure 11-5** Configure Defaults—Computer Defaults Page

The screenshot shows the 'Configure Defaults' page for 'Computer Defaults'. The page header includes 'Broadband Access Center for Cable' and a navigation menu with 'Configuration' selected. The user is identified as 'admin' with the role of 'Administrator'. The main content area is titled 'Computer Defaults' and contains the following fields:

- Extension Point:
- Disruption Extension Point:
- Service Level Selection Extension Point:
- Default Class of Service:
- Default DHCP Criteria:
- Automatic FQDN Generation:  Enabled,  Disabled

At the bottom of the form are 'Submit' and 'Reset' buttons. A sidebar on the left lists other default categories like ATA 186, CHWAN DATA, and DOCSIS. A vertical ID '129895' is visible on the right side of the form area.

Refer to [Table 11-2](#) for the description of all fields and buttons appearing in [Figure 11-5](#).



### Note

Changes to the default Class of Service or default DHCP criteria cause regeneration to occur. Other changes made to this page do not affect existing devices.

# DOCSIS Defaults

When the DOCSIS Defaults option is selected, the DOCSIS Defaults page appears. See [Figure 11-6](#). Use this page to display a list of default DOCSIS values currently applied to cable modems that BAC supports.

**Figure 11-6** Configure Defaults–DOCSIS Defaults Page

The screenshot shows the 'Broadband Access Center for Cable' web interface. The top navigation bar includes 'Logout', 'Configuration', 'Devices', 'Nodes', 'Servers', and 'Users'. Below this, a secondary navigation bar lists 'Class of Service', 'Custom Property', 'Defaults', 'DHCP Criteria', 'External Files', 'License Keys', and 'Publishing'. The user is identified as 'admin' with the role of 'Administrator'.

The main content area is titled 'Configure Defaults' and includes the instruction: 'Use this page to change the defaults. Fields marked with an \* are required.' A sidebar on the left lists various default categories, with 'DOCSIS Defaults' selected and highlighted in red.

The 'DOCSIS Defaults' configuration table is as follows:

|                                                      |                                                                                |
|------------------------------------------------------|--------------------------------------------------------------------------------|
| Extension Point:                                     | <input type="text" value="com.cisco.csrc.extensions.DOCSISExtension"/>         |
| Disruption Extension Point:                          | <input type="text" value="com.cisco.csrc.extensions.DOCSISDeviceDisrup"/>      |
| Service Level Selection Extension Point:             | <input type="text" value="com.cisco.provisioning.cpe.extensions.builtin.sel"/> |
| Default Class of Service:                            | <input type="text" value="unprovisioned-docsis"/>                              |
| Default DHCP Criteria:                               | <input type="text" value="unprovisioned-docsis"/>                              |
| TFTP Modem Address Option:                           | <input type="radio"/> Enabled<br><input checked="" type="radio"/> Disabled     |
| TFTP Time Stamp Option:                              | <input type="radio"/> Enabled<br><input checked="" type="radio"/> Disabled     |
| Automatic FQDN Generation:                           | <input type="radio"/> Enabled<br><input checked="" type="radio"/> Disabled     |
| CMTS Shared Secret:                                  | <input type="text" value="*****"/>                                             |
| CMTS Default Docsis Version:                         | <input type="text" value="1.0"/>                                               |
| Relay Agent IP Address to CMTS Version Mapping file: | <input type="text"/>                                                           |

At the bottom of the configuration area are 'Submit' and 'Reset' buttons.



**Note**

Changes to the default Class of Service or default DHCP criteria cause regeneration to occur. Changes to any TFTP option come into effect starting from the next TFTP transfer.

129896

Refer to [Table 11-4](#) for the description of all fields and buttons appearing in [Figure 11-6](#).

**Table 11-4** *Configure Defaults—DOCSIS Defaults Page*

| Field or Button                                     | Description                                                                                                                                                                                                                                                                                                                                                                        |
|-----------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Extension Point                                     | Identifies the extension point to execute when generating a configuration for a DOCSIS device.                                                                                                                                                                                                                                                                                     |
| Disruption Extension Point                          | Identifies the extension point to be executed to disrupt a DOCSIS device.                                                                                                                                                                                                                                                                                                          |
| Service Level Selection Extension Point             | Identifies the extension used to determine the DHCP criteria and Class of Service required for a device.                                                                                                                                                                                                                                                                           |
| Default Class of Service                            | Identifies the current default Class of Service for a device. New, unrecognized devices are assigned to this Class of Service. Use the drop-down list to select a new default value.                                                                                                                                                                                               |
| Default DHCP Criteria                               | Identifies the current default DHCP criteria for a specific device technology. New, unrecognized devices are assigned this default DHCP criteria. Use the drop-down list to select a new default value.                                                                                                                                                                            |
| TFTP Modem Address Option                           | Identifies whether the TFTP modem address option is enabled.                                                                                                                                                                                                                                                                                                                       |
| TFTP Time Stamp Option                              | Identifies whether the TFTP server will issue a timestamp.                                                                                                                                                                                                                                                                                                                         |
| Automatic FQDN Generation                           | Automatically generates a host and domain name for the device. Two selectable options are available: <ul style="list-style-type: none"> <li>• Enabled—Automatic generation of the FQDN is enabled.</li> <li>• Disabled—Automated FQDN generation is disabled.</li> </ul> <p><b>Note</b> See <a href="#">Automatic FQDN Generation, page 11-38</a>, for additional information.</p> |
| CMTS Shared Secret                                  | Identifies the character string that BAC uses in the calculation of the CMTS MIC in the configuration file. The CMTS uses it to authenticate the configuration file that a cable modem submits to the CMTS for authorization.                                                                                                                                                      |
| CMTS Default Docsis Version                         | Specifies the default DOCSIS version used by all CMTSs. If you do not enter a DOCSIS version in this field, it will default to version 1.0.                                                                                                                                                                                                                                        |
| Relay Agent IP Address to CMTS Version Mapping file | Identifies the mapping file used by the CMTS. This file specifies the DOCSIS version that the CMTS will use.                                                                                                                                                                                                                                                                       |
| <b>Submit</b>                                       | Activates the changes you have made. After the administrative database has been updated the Configure Defaults page will reflect the changes you have made.                                                                                                                                                                                                                        |
| <b>Reset</b>                                        | Returns all settings to their previous setting.                                                                                                                                                                                                                                                                                                                                    |



**Note**

If you enable either or both of the TFTP options on this page, that appropriate TFTP information is included in the TFTP file before it is sent to the DOCSIS cable modem.



## Network Registrar Defaults

BAC provides Network Registrar (NR) extension points that allow BAC to pull information from incoming DHCP packets to detect a device's technology. The extension points also let BAC respond to device DHCP requests with options that correspond to the configuration stored at the DPE.

When the NR Defaults option is selected, the NR Defaults page appears. See [Figure 11-7](#).

**Figure 11-7** Configure Defaults—NR Defaults Page

**Broadband Access Center for Cable** Logout

[Configuration](#) | [Devices](#) | [Nodes](#) | [Servers](#) | [Users](#)  
[Class of Service](#) | [Custom Property](#) | **[Defaults](#)** | [DHCP Criteria](#) | [External Files](#) | [License Keys](#) | [Publishing](#)

User:admin Role:Administrator

---

**CISCO SYSTEMS** **Configure Defaults**  
 Use this page to change the defaults.  
 Fields marked with an "\*" are required.

**Defaults**

- [\\_ATA 186 Defaults](#)
- [\\_ATA 188 Defaults](#)
- [\\_CHWAN DATA Defaults](#)
- [\\_CHWAN MAN Defaults](#)
- [\\_Computer Defaults](#)
- [\\_DOCSIS Defaults](#)
- [\\_NR Defaults](#)
- [\\_Packet Cable Defaults](#)
- [\\_RDU Defaults](#)
- [\\_System Defaults](#)
- [\\_XGCP Defaults](#)

**Extension Point Settings**

|                                                                           |                                                                                |
|---------------------------------------------------------------------------|--------------------------------------------------------------------------------|
| NR Attributes from Request Dictionary (version 2.0):                      | <input type="text" value="chaddr,client-id,client-id-created-from-mac-addr"/>  |
| NR Attributes from Request Dictionary as Bytes (version 2.5 and above):   | <input type="text" value="chaddr,client-id,client-id-created-from-mac-addr"/>  |
| NR Attributes from Request Dictionary as Strings (version 2.5 and above): | <input type="text" value="dhcp-class-identifier,dhcp-parameter-request-list"/> |
| NR Attributes from Environment Dictionary:                                | <input type="text"/>                                                           |
| NR Attributes Required in Request Dictionary:                             | <input type="text" value="relay-agent-remote-id"/>                             |

Refer to [Table 11-5](#) for the description of all fields and buttons appearing in [Figure 11-7](#).

**Table 11-5** Configure Defaults—Network Registrar Defaults Page

| Field or Button                                                        | Description                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NR Attributes from Request Dictionary (version 2.0)                    | Identifies a comma-separated list of attributes pulled from the Network Registrar request dictionary, as strings, when sending a request to the RDU to generate a configuration for the current device.<br><br><b>Note</b> This property applies only to the BPR 2.0 Network Registrar extensions.           |
| NR Attributes from Request Dictionary as Bytes (version 2.5 and above) | Identifies a comma-separated list of attributes pulled out of the Network Registrar request dictionary as bytes when sending a request to the RDU to generate a configuration for the current device.<br><br><b>Note</b> This property applies only to the BACC 2.5 (or later) Network Registrar extensions. |

**Table 11-5** *Configure Defaults–Network Registrar Defaults Page (continued)*

| Field or Button                                                         | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NR Attributes from Request Directory as Strings (version 2.5 and above) | Identifies a comma-separated list of attributes pulled from the Network Registrar request dictionary as strings when sending a request to the RDU to generate a configuration for the current device.<br><b>Note</b> This property applies only to the BACC 2.5 (or later) Network Registrar extensions.                                                                                                                                                                      |
| NR Attributes from Environment Directory                                | Identifies a comma-separated list of attributes pulled out of the Network Registrar environment dictionary as strings when sending a request to the RDU to generate a configuration for the current device.<br><b>Note</b> This property applies to both BPR 2.0 and BACC 2.5 (or later) Network Registrar extensions.                                                                                                                                                        |
| NR Attributes Required in Request Dictionary                            | Identifies a comma-separated list of attributes pulled from the Network Registrar request dictionary for Network Registrar extensions to submit a request to the RDU to generate a configuration for the current device.<br>The default value for this field is the relay agent remote ID option. If you do not set the <b>relay-agent-remote-id</b> value in this field, Network Registrar extensions reject devices from triggering a request for configuration generation. |
| <b>Submit</b>                                                           | Activates or implements the changes you have made. After the administrative database has been updated to reflect the changes you make, modified changes appear in the Configure Defaults page.                                                                                                                                                                                                                                                                                |
| <b>Reset</b>                                                            | Returns all settings to their previous setting.                                                                                                                                                                                                                                                                                                                                                                                                                               |

**Note**

Changes made to this page do not take effect until the Network Registrar extensions are reloaded.

# PacketCable Defaults

The PacketCable Defaults page identifies those defaults necessary to support the PacketCable voice technology. When selected the PacketCable Defaults page appears. See [Figure 11-8](#).

**Figure 11-8** *Configure Defaults–PacketCable Defaults Page*

**CISCO SYSTEMS** | **Configure Defaults**  
 Use this page to change the defaults.  
 Fields marked with an "\*" are required.

**Defaults**

- [ATA 186 Defaults](#)
- [ATA 188 Defaults](#)
- [CHWAN DATA Defaults](#)
- [CHWAN MAN Defaults](#)
- [Computer Defaults](#)
- [DOCSIS Defaults](#)
- [NR Defaults](#)
- [Packet Cable Defaults](#)
- [RDU Defaults](#)
- [System Defaults](#)
- [XGCP Defaults](#)

**Packet Cable Defaults**

Extension Point:

Disruption Extension Point:

Service Level Selection Extension Point:

Default Class of Service:

Default DHCP Criteria:

SNMP Set Timeout (secs):

MTA Provisioning Notification:

Automatic FQDN Generation:  Enabled  Disabled

129991

[Table 11-6](#) identifies the fields and buttons that are unique to this defaults page.

**Table 11-6** *Configure Defaults–PacketCable Defaults Page*

| Field or Button                         | Description                                                                                                                                                                          |
|-----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Extension Point                         | Identifies the extension point to execute when generating a configuration for a device of this technology.                                                                           |
| Disruption Extension Point              | Identifies the extension point to be executed to disrupt a device of this technology.                                                                                                |
| Service Level Selection Extension Point | Identifies the extension used to determine what DHCP criteria and Class of Service required for a device.                                                                            |
| Default Class of Service                | Identifies the current default Class of Service for a device. New, unrecognized devices are assigned to this Class of Service. Use the drop-down list to select a new default value. |

**Table 11-6** *Configure Defaults—PacketCable Defaults Page (continued)*

| <b>Field or Button</b>        | <b>Description</b>                                                                                                                                                                                                                                                                                                                  |
|-------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Default DHCP Criteria         | Identifies the current default DHCP criteria for a specific device technology. New, unrecognized devices are assigned this default DHCP criteria. Use the drop-down list to select a new default value.                                                                                                                             |
| SNMP Set Timeout              | Identifies the SNMP set timeout in seconds.                                                                                                                                                                                                                                                                                         |
| MTA Provisioning Notification | Notification that an MTA event has taken place. An event occurs when the MTA sends its provisioning complete inform based on the selected choice. Options available include: <ul style="list-style-type: none"> <li>• On Failure</li> <li>• On Success</li> <li>• During Provisioning</li> <li>• Always</li> <li>• Never</li> </ul> |
| Automatic FQDN Generation     | Identifies whether a fully qualified domain name (FQDN) will be generated.                                                                                                                                                                                                                                                          |
| <b>Submit</b>                 | Activates the changes you have made. After the administrative database has been updated, the Configure Defaults page reflects the changes you have made.                                                                                                                                                                            |
| <b>Reset</b>                  | Returns all settings to their previous setting.                                                                                                                                                                                                                                                                                     |

## RDU Defaults

When you select the RDU defaults link, the RDU Defaults page appears. See [Figure 11-9](#). Use this page to configure the RDU to communicate with Network Registrar. For additional information, refer to the *Cisco Network Registrar User's Guide, 6.2.1*.

**Figure 11-9** Configure Defaults–RDU Defaults Page

**Broadband Access Center for Cable** Logout

Class of Service | Custom Property | **Defaults** | DHCP Criteria | External Files | License Keys | Publishing

User:admin Role:Administrator

**CISCO SYSTEMS** **Configure Defaults**  
Use this page to change the defaults.  
Fields marked with an \* are required.

**Defaults**

- [ATA 186 Defaults](#)
- [ATA 188 Defaults](#)
- [CHYWAN DATA Defaults](#)
- [CHYWAN MAN Defaults](#)
- [Computer Defaults](#)
- [DOCSIS Defaults](#)
- [NR Defaults](#)
- [Packet Cable Defaults](#)
- [RDU Defaults](#)
- [System Defaults](#)
- [XGCP Defaults](#)

**RDU Defaults**

Configuration Extension Point:

Device Detection Extension Point:

Publishing Extension Point:

Extension Point Jar File Search Order:

CCM Server IP Address:

CCM Server Port:

CCM Server User:

CCM Server Password:

CCM Server Confirm Password:

CCM Server:  Enabled  Disabled

CCM Server Timeout (secs):

210668

Table 11-7 describes all fields and buttons appearing in Figure 11-9.

**Table 11-7** *Configure Defaults—RDU Defaults Page*

| Field or Button                       | Description                                                                                                                                                          |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Configuration Extension Point         | Identifies the common extension points executed before any other technology extension point is executed.                                                             |
| Device Detection Extension Point      | Identifies the extension point used to determine a device type (for example, DOCSIS or computer) based on information pulled from the device DHCP Discover requests. |
| Publishing Extension Point            | Identifies the extension point to be used for an RDU publishing plug-in. This information is useful when you need to publish RDU data into another database.         |
| Extension Point Jar File Search Order | Specifies the sequence in which the classes are searched in the Jar files that are listed in the preceding four fields.                                              |
| CCM Server IP Address                 | Identifies the IP address of the CCM server.                                                                                                                         |
| CCM Server Port                       | Identifies the CCM server port on which BAC communicates.                                                                                                            |
| CCM Server User                       | Identifies the CCM server username and is used in conjunction with the password fields.                                                                              |
| CCM Server Password                   | Identifies the password used to authenticate the CCM Server User.                                                                                                    |
| CCM Server Confirm Password           | Authenticates the CCM Server Password.                                                                                                                               |
| CCM Server                            | Specifies whether the BAC interface to the CCM Server is enabled or disabled.                                                                                        |
| CCM Server Timeout                    | Specifies the length of time, in seconds, that BAC attempts to connect with the CCM Server until BAC declares the connection down.                                   |
| <b>Submit</b>                         | Activates the changes you have made. After the administrative database has been updated, the Configure Defaults page reflects the latest changes.                    |
| <b>Reset</b>                          | Returns all settings to the previous setting.                                                                                                                        |



**Note**

See [Managing RDU Extensions, page 11-32](#), for information on RDU extension points.

# System Defaults

When you select the Systems Defaults link, the System Defaults page appears. See [Figure 11-10](#).

**Figure 11-10** Configure Defaults–System Defaults Page



**Note** You can configure the default values by using the BAC API.

[Table 11-8](#) describes all fields and buttons appearing in [Figure 11-10](#).

**Table 11-8** Configure System Defaults Page

| Field or Button             | Description                                                                                                                                            |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| SNMP Write Community String | Identifies the default write community string for any device that may require SNMP information. The default write community string is <b>private</b> . |
| SNMP Read Community String  | Identifies the default read community string for any device that can read or access the SNMP MIB. The default read community string is <b>public</b> . |

Table 11-8 Configure System Defaults Page (continued)

| Field or Button                               | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Promiscuous Mode                              | Identifies whether the Promiscuous mode is enabled. There are two options: <ul style="list-style-type: none"> <li>• Enable—Enables the Promiscuous mode within BAC.</li> <li>• Disable—Disables the Promiscuous mode within BAC.</li> </ul>                                                                                                                                                                                                                                                                                                         |
| Default Provisioned Promiscuous DHCP Criteria | Identifies the default DHCP criteria used to provision a CPE in the Promiscuous mode, when the device that the CPE is behind does not have a CPE DHCP criteria specified.                                                                                                                                                                                                                                                                                                                                                                           |
| Default Device Type for Device Detection      | Identifies the default device type for a device not previously registered in the RDU. The options include: <ul style="list-style-type: none"> <li>• DOCSIS</li> <li>• COMPUTER</li> <li>• PacketCableMTA</li> <li>• CableHomeWanMan</li> <li>• CableHomeWanData</li> <li>• None</li> </ul> <p><b>Note</b> If the device detection extension is unable to identify the device type, the “default type” (for example, COMPUTER) specifies the device type. If you set the Default Device Type to None, the device record is not added to the RDU.</p> |
| Maximum Diagnostic Device Count               | Identifies the maximum number of MAC addresses (devices) that you can troubleshoot at any one time.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| MIB List                                      | Identifies a list of MIBs used by the RDU that do not require restarting the RDU.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Supplemental MIB List                         | Identifies an extended list of MIBs used by the RDU.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Excluded MIB Tokens                           | Defines those key words, or tokens, that cannot be redefined by a MIB.                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Excluded Supplemental MIB Tokens              | Defines those additional key words, or tokens, that cannot be redefined by a MIB and do not appear in the Excluded MIB Tokens list.                                                                                                                                                                                                                                                                                                                                                                                                                 |
| <b>Submit</b>                                 | Activates the changes you have made. After the administrative database has been updated, the Configure Defaults page reflects the changes you have made.                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Reset</b>                                  | Returns all settings to their previous setting.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |



## Gateway (xGCP) Control Protocol Defaults

XGCP is a Gateway Control Protocol that lets external call agents control gateways in a VoIP environment. The xGCP Defaults page (Figure 11-11) displays a list of default values currently applied to the xGCP gateway devices supported by BAC.

Figure 11-11 Configure Defaults–XGCP Defaults Page

The screenshot shows the 'Broadband Access Center for Cable' web interface. The top navigation bar includes 'Logout' and a menu with 'Configuration', 'Devices', 'Nodes', 'Servers', and 'Users'. Below this is a secondary menu with 'Class of Service', 'Custom Property', 'Defaults', 'DHCP Criteria', 'External Files', 'License Keys', and 'Publishing'. The user is identified as 'admin' with the role of 'Administrator'. The main content area is titled 'Configure Defaults' and includes a Cisco Systems logo and instructions: 'Use this page to change the defaults. Fields marked with an \* are required.' A sidebar on the left lists various default configuration categories, with 'XGCP Defaults' selected. The main form area is titled 'XGCP Defaults' and contains three fields: 'Signalling Type' with a text input containing 'S', 'Version Number' with a text input containing '1.1', and 'Use old format for merit-dump string' with radio buttons for 'Enabled' and 'Disabled' (selected). At the bottom of the form are 'Submit' and 'Reset' buttons. A vertical ID number '129885' is visible on the right side of the form area.

Table 11-9 describes all fields and buttons appearing in Figure 11-11.

Table 11-9 Configure XGCP Defaults Page

| Field or Button                      | Description                                                                                                                                              |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Signalling Type                      | Identifies the xGCP signaling type, such as: S, M, and so on.                                                                                            |
| Version Number                       | Identifies the xGCP version number in use.                                                                                                               |
| Use old format for merit-dump string | Enables or disables the use of the old string format, which does not include the version number.                                                         |
| <b>Submit</b>                        | Activates the changes you have made. After the administrative database has been updated, the Configure Defaults page reflects the changes you have made. |
| <b>Reset</b>                         | Returns all settings to their previous setting.                                                                                                          |



### Note

Subsequent device configurations will include the changes you implement here. However, all existing configurations are not changed. To make the changes in any existing configuration, you must regenerate the configuration using the API.

## Configuring DHCP Criteria

In BAC, DHCP criteria describe the specific criteria for a device when selecting a scope in Network Registrar. For example, a DHCP criteria called **provisioned-docsis** has an inclusion selection tag called **tagProvisioned**. The DHCP criteria is associated with a DOCSIS modem. When this modem requests an IP address from the Network Registrar, Network Registrar looks for scopes associated with the scope-selection tag **tagProvisioned**.

To access the DHCP Criteria page:

- 
- Step 1** Choose **Configuration** on the Primary Navigation bar.
  - Step 2** Choose **DHCP Criteria** from the Secondary Navigation bar and the Manage DHCP Criteria page appears.
- 

## Adding DHCP Criteria

To add a DHCP criteria:

- 
- Step 1** Click **Add**, on the DHCP Criteria page, and the Add DHCP Criteria page appears.
  - Step 2** Enter the name of the DHCP criteria you want to create.
  - Step 3** Enter the DHCP Criteria client-class name.
  - Step 4** Enter the inclusion and exclusion selection tags.



**Note** When creating new DHCP criteria, the client-class and Inclusion and Exclusion selection tag names you enter must be the exact names from within Network Registrar. For additional information on client class and selection tags, refer to the *Cisco Network Registrar User's Guide, 6.2.1*, and the *Cisco Network Registrar CLI Reference, 6.2.1*. You should specify either the client class, or inclusion and exclusion selection tag names, when creating a new DHCP criteria.

---

- Step 5** You can add or modify the properties that are added on the DHCP criteria. Enter or select a Property Name, or select an existing name, and enter or modify the appropriate Property Value.
  - Step 6** Click **Add** after changing or creating the property name-property value pair.
  - Step 7** Click **Submit**. After the DHCP criteria is successfully added in the RDU database, it will be visible in the Manage DHCP Criteria Page.
-

## Modifying DHCP Criteria

To modify existing DHCP criteria:

- 
- Step 1** On the Manage DHCP criteria page, click the DHCP criteria link that you want to modify and the Modify DHCP Criteria page appears.
  - Step 2** Make the desired changes to the client class, inclusion and exclusion selection tags, and the property value settings.
  - Step 3** Click **Submit**. After successful modification of the DHCP criteria in the RDU Database, the Manage DHCP Criteria page appears.
- 

**Note**

Subsequent device configurations will include the changes you implement here. All existing configurations are regenerated, although the devices on the network will not get the new configuration until they are rebooted.

---

## Deleting DHCP Criteria

Deleting DHCP criteria using the administrator application does not delete the actual DHCP server configurations from the DHCP server. You must delete the DHCP server configurations manually. To delete an existing criteria:

- 
- Step 1** Choose **Configuration** on the Primary Navigation bar.
  - Step 2** Choose **DHCP Criteria** from the Secondary Navigation bar and the Manage DHCP Criteria page appears.
  - Step 3** Click the **Delete** icon corresponding to the criteria you want to delete, and a deletion dialog box appears.
  - Step 4** Click **OK** to delete the criteria or click **Cancel** to abort the operation. The Manage DHCP Criteria page appears.

**Note**

You can delete a DHCP criteria only if there are no devices associated with that criteria, and it is not designated as the default DHCP criteria. If a DHCP criteria has devices associated with it, you must associate a different DHCP criteria before deleting the criteria.

---

# Managing External Files

By using the BAC administrator user interface, you can manage the TFTP server files or template files for dynamic generation for DOCSIS, PacketCable MTAs, and WAN-MAN files, or software images for devices. See [Figure 11-12](#). Use this page to add, delete, replace, or export any file type, including:

- Template files—These are text files that contain DOCSIS, PacketCable, or CableHome options and values that, when used in conjunction with a particular Class of Service, provide dynamic file generation.



**Note** Template files can be created in any text editor, but must have a .tmpl file type. For additional template information, refer to [Developing Template Files, page 8-1](#).

- Static configuration files—These files are used as a configuration file for a device. For example, a static configuration file, called *gold.cm*, would identify the gold DOCSIS Class of Service. BAC treats this file type like any other binary file.
- IOS images—These are images stored in firmware for a Cisco device. The Cisco device can upload the image to upgrade its functionality. BAC treats this file type like any other binary file.



**Note** Once you click the Search button on the View External Files page, [Figure 11-12](#) appears.

**Figure 11-12 View External Files Page**

**Broadband Access Center for Cable** Logout

Configuration | Devices | Nodes | Servers | Users  
 Class of Service | Custom Property | Defaults | DHCP Criteria | **External Files** | License Keys | Publishing  
 User:admin Role:Administrator

**CISCO SYSTEMS** **View External Files**  
 Use this page to view an external file.

External File or External File wildcard: \* Page Size: 25 Search

Delete Add

| External Files                                                   | View | Export |
|------------------------------------------------------------------|------|--------|
| <input type="checkbox"/> <a href="#">bronze.cm</a>               |      |        |
| <input type="checkbox"/> <a href="#">changeloggers.jar</a>       |      |        |
| <input type="checkbox"/> <a href="#">gold.cm</a>                 |      |        |
| <input type="checkbox"/> <a href="#">removetimeservers.jar</a>   |      |        |
| <input type="checkbox"/> <a href="#">unprov_packet_cable.bin</a> |      |        |
| <input type="checkbox"/> <a href="#">unprov_wan_man.cfg</a>      |      |        |

Result Pages: 1

129887

Table 11-10 identifies the fields and buttons shown in Figure 11-12.

**Table 11-10 View External Files Page**

| Field or Button     | Description                                                                                                                                                                                                                                                    |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| External Files      | Identifies the filename. An asterisk (*) can be used as a wildcard character to allow searching for partial filenames. For example, you can enter *.cm to list all external files ending with the .cm extension. An example of an invalid wildcard is bronze*. |
| Page Size           | Identifies the length of page to be displayed.                                                                                                                                                                                                                 |
| Search              | Initiates the search for an external file with a name that matches the entry in the External Files field.                                                                                                                                                      |
| Delete              | Removes any selected external file from the database.                                                                                                                                                                                                          |
| Add                 | Adds a new file.                                                                                                                                                                                                                                               |
| External Files list | Displays a list of external files that match the search criteria.<br><b>Note</b> The check boxes immediately to the left of any selected item in this list must be checked before it can be deleted.                                                           |
| View                | Displays the details of the selected binary file.                                                                                                                                                                                                              |
| Export              | Exports any selected file to the client's computer.                                                                                                                                                                                                            |

## Adding External Files

To add an existing external file:

- 
- Step 1** Choose **Configuration** on the Primary Navigation bar.
  - Step 2** Choose **External Files** from the Secondary Navigation bar. The View External Files page appears.
  - Step 3** Click **Add** and the Add External Files page appears.
  - Step 4** Enter the **Source filename** and the **External filename**.



**Note** If you do not know the exact name of the source file, use the **Browse** function to navigate to the desired directory and select the file. File sizes up to 4 MB are supported.

- Step 5** Click **Submit**. The View External Files page appears to indicate that the file has been added.
- 

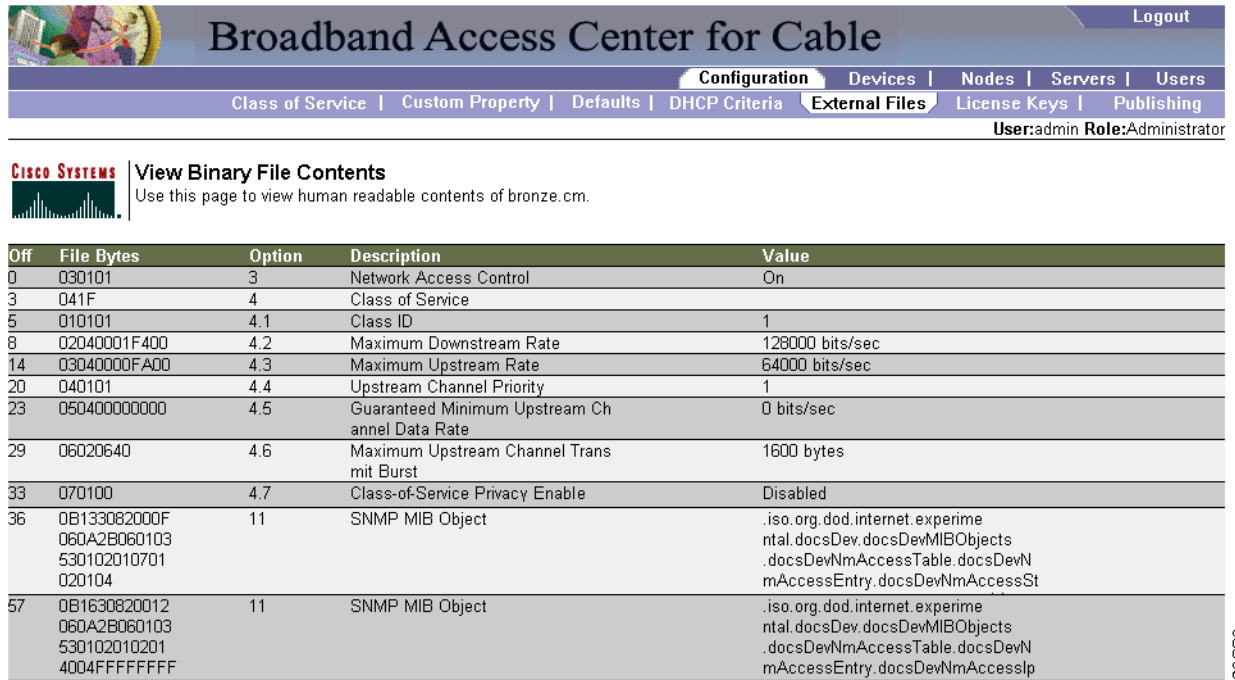
## Viewing External Files

To view the contents of a DOCSIS or PacketCable voice technology external file:

- 
- Step 1** Choose **Configuration** on the Primary Navigation bar.
  - Step 2** Choose **External Files** from the Secondary Navigation bar. The View External Files page appears.
  - Step 3** Search for the required file using the search field and appropriate wildcard characters.

- Step 4** Click the **View Details** icon (🔍) corresponding to the DOCSIS, CableHome WAN-MAN, and PacketCable MTA binary configuration files. A View Binary File Contents page appears. [Figure 11-13](#) identifies sample binary file content.

**Figure 11-13** Sample Binary File Content



**Broadband Access Center for Cable** Logout

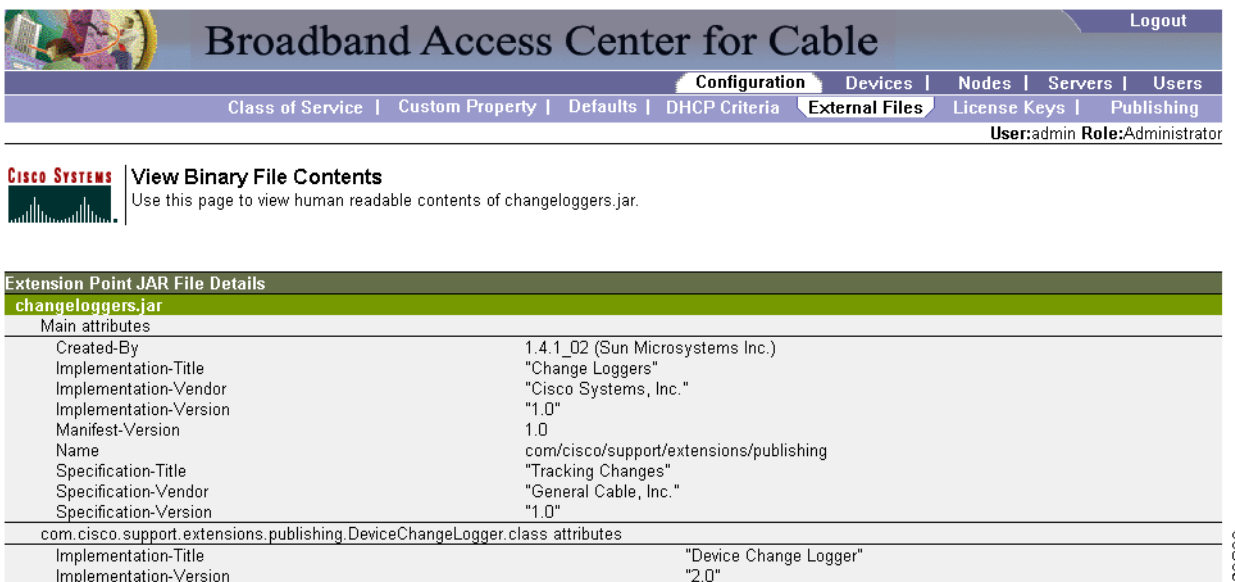
Configuration | Devices | Nodes | Servers | Users  
 Class of Service | Custom Property | Defaults | DHCP Criteria | **External Files** | License Keys | Publishing  
 User:admin Role:Administrator

**CISCO SYSTEMS** View Binary File Contents  
 Use this page to view human readable contents of bronze.cm.

| Off | File Bytes                                       | Option | Description                                   | Value                                                                                                                    |
|-----|--------------------------------------------------|--------|-----------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| 0   | 030101                                           | 3      | Network Access Control                        | On                                                                                                                       |
| 3   | 041F                                             | 4      | Class of Service                              |                                                                                                                          |
| 5   | 010101                                           | 4.1    | Class ID                                      | 1                                                                                                                        |
| 8   | 02040001F400                                     | 4.2    | Maximum Downstream Rate                       | 128000 bits/sec                                                                                                          |
| 14  | 03040000FA00                                     | 4.3    | Maximum Upstream Rate                         | 64000 bits/sec                                                                                                           |
| 20  | 040101                                           | 4.4    | Upstream Channel Priority                     | 1                                                                                                                        |
| 23  | 050400000000                                     | 4.5    | Guaranteed Minimum Upstream Channel Data Rate | 0 bits/sec                                                                                                               |
| 29  | 06020640                                         | 4.6    | Maximum Upstream Channel Transmit Burst       | 1600 bytes                                                                                                               |
| 33  | 070100                                           | 4.7    | Class-of-Service Privacy Enable               | Disabled                                                                                                                 |
| 36  | 0B133082000F060A2B060103530102010701020104       | 11     | SNMP MIB Object                               | .iso.org.dod.internet.experimental.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNmAccessEntry.docsDevNmAccessSt |
| 57  | 0B1630820012060A2B0601035301020102014004FFFFFFFF | 11     | SNMP MIB Object                               | .iso.org.dod.internet.experimental.docsDev.docsDevMIBObjects.docsDevNmAccessTable.docsDevNmAccessEntry.docsDevNmAccessIp |

[Figure 11-14](#) identifies sample Jar file content.

**Figure 11-14** Sample Jar File Content



**Broadband Access Center for Cable** Logout

Configuration | Devices | Nodes | Servers | Users  
 Class of Service | Custom Property | Defaults | DHCP Criteria | **External Files** | License Keys | Publishing  
 User:admin Role:Administrator

**CISCO SYSTEMS** View Binary File Contents  
 Use this page to view human readable contents of changeloggers.jar.

**Extension Point JAR File Details**

**changeloggers.jar**

Main attributes

|                        |                                         |
|------------------------|-----------------------------------------|
| Created-By             | 1.4.1_02 (Sun Microsystems Inc.)        |
| Implementation-Title   | "Change Loggers"                        |
| Implementation-Vendor  | "Cisco Systems, Inc."                   |
| Implementation-Version | "1.0"                                   |
| Manifest-Version       | 1.0                                     |
| Name                   | com/cisco/support/extensions/publishing |
| Specification-Title    | "Tracking Changes"                      |
| Specification-Vendor   | "General Cable, Inc."                   |
| Specification-Version  | "1.0"                                   |

com.cisco.support.extensions.publishing.DeviceChangeLogger.class attributes

|                        |                        |
|------------------------|------------------------|
| Implementation-Title   | "Device Change Logger" |
| Implementation-Version | "2.0"                  |

## Replacing External Files

To replace an existing external file:

- 
- Step 1** Choose **Configuration** on the Primary Navigation bar.
  - Step 2** Choose **External Files** from the Secondary Navigation bar.
  - Step 3** Select the link that corresponds to the file you want to replace from the search output list. The Replace External Files page appears. Note that the selected filename already appears on this page.
  - Step 4** Enter the path and filename of the source file to be used as a replacement for the displayed external filename.



---

**Note** If you do not know the exact name or location of the source file, use the **Browse** button to navigate to the desired directory and select the file.

---

- Step 5** Click **Submit**. After submitting the replacement file, a confirmation page appears to indicate that, after replacement, BAC will regenerate configurations for the affected devices.
- Step 6** Click **OK** and the View External Files page appears.



---

**Note** All devices using this file through a Class of Service are regenerated after the replacement is finished.

---

## Exporting External Files

You can copy external files to your local hard drive using the export function.




---

**Note** The procedure described below assumes that you are using Internet Explorer. This procedure is different if you are using Netscape Navigator.

---

To export a file:

- 
- Step 1** Choose **Configuration** on the Primary Navigation bar.
  - Step 2** Choose **External Files** from the Secondary Navigation bar.
  - Step 3** Identify the external file that you want to export.
  - Step 4** Click the **Export** icon () and you are prompted to either open the file or save it.
  - Step 5** Return to the BAC user interface.
-

## Deleting External Files

Complete this procedure to delete an existing external file:

- 
- Step 1** Choose **Configuration** on the Primary Navigation bar.
  - Step 2** Choose **External Files** from the Secondary Navigation bar.
  - Step 3** In the **External Files** field, enter the filename of the external file that you want to modify.
  - Step 4** Click **Search**. The appropriate file will appear in the External Files list.
  - Step 5** Choose the appropriate file or files.
  - Step 6** Click **Delete**.

**Caution**

Deleting a template file that is not directly linked to a Class of Service, but is referenced by another template file that is linked to a Class of Service, will cause the configuration regeneration service to fail.

**Note**

You cannot delete a file that has a Class of Service associated with it. You must remove the Class of Service association before proceeding. See [Configuring Class of Service, page 11-1](#), for additional information.

## Managing License Keys

Software licenses are used to activate specific features or to increase the functionality of your installation. Each license is available as either a permanent license or an evaluation license.

- **Permanent**—A permanent license is purchased for use in your network environment and activates the specific features for which it is intended.
- **Evaluation**—An evaluation license enables functionality for a specific amount of time after installation. You can upgrade an evaluation license to a permanent license by entering a new permanent license number.

**Caution**

Do not attempt to deploy into a fully operational network with an evaluation license key installed. Any provisioning done by using an evaluation license is disabled when that evaluation license expires.

When you upgrade from an evaluation license to a permanent license, you do not have to reinstall the software or reconfigure BAC. You simply have to provide the permanent license via the BAC administrator user interface.

The Manage License Keys page displays a list of licenses that have been entered for your implementation. This BAC release supports both evaluation and permanent licenses for high-speed data (DOCSIS cable modems), PacketCable MTAs, ATAs, DPEs, CableHome WAN-MAN and WAN-Data devices, and computers. The status of each available license appears as active or expired (shown by the expiration date).



**Note**

You can upgrade a permanent license to increase the number of authorized devices by adding an additional license. When you reach the limit of your number of licensed devices you cannot provision new devices, but existing devices that are already provisioned continue to receive service.

Figure 11-15 identifies a sample Manage License Keys page.

**Figure 11-15** Manage License Keys Page

| Technology       | License Key               | Version | Type      | Devices   | Status                     |
|------------------|---------------------------|---------|-----------|-----------|----------------------------|
| DPE              | dpePerm32007              | 2.0.0   | Permanent | 20        | Installed on April 3, 2007 |
| cablehomewandata | cablehomewandataPerm32007 | 2.0.0   | Permanent | 100000000 | Installed on April 3, 2007 |
| cablehomewanman  | cablehomewanmanPerm32007  | 2.0.0   | Permanent | 100000000 | Installed on April 3, 2007 |
| computer         | computerPerm32007         | 2.0.0   | Permanent | 100000000 | Installed on April 3, 2007 |
| customcpe        | customcpePerm32007        | 2.0.0   | Permanent | 100000000 | Installed on April 3, 2007 |
| docsis           | docsisPerm32007           | 2.0.0   | Permanent | 100000000 | Installed on April 3, 2007 |
| packetcable      | packetcablePerm32007      | 2.0.0   | Permanent | 100000000 | Installed on April 3, 2007 |
| xgcp             | xgcpPerm32007             | 2.0.0   | Permanent | 100000000 | Installed on April 3, 2007 |

License Key:

210856

## Adding and Modifying a License

To add, modify, or upgrade a license:

- Step 1** Choose **Configuration > License Keys**.
- Step 2** Obtain your new license key from either your Cisco representative or the Cisco Technical Assistance Center (TAC) website. See the Preface in this guide for TAC contact information.
- Step 3** Enter the new license key in the License Key field.
- Step 4** Click **Add/Upgrade** to install the new license key. If you enter a permanent license key, it overwrites the corresponding evaluation key (if that key was installed). If you enter a license key (permanent or evaluation) for a new technology, it will appear in the technology list.

## Deleting a License

To delete a license:

- Step 1** Choose **Configuration > License Keys** from the Navigation bar.  
The Manage License Keys page appears.
- Step 2** Copy the license key corresponding to the technology you want to delete.

**Step 3** Paste the license key in the License Key field. Click **Delete**.

A Confirmation dialog box appears.

**Step 4** To confirm deleting the license key, click **Yes**; otherwise click **No**.

The license key disappears from the Manage License Keys page.



**Note** To confirm if the license has been deleted, verify if the action has been recorded in *audit.log*.

## Managing RDU Extensions

Creating a custom extension point is a programming activity that can, when used with the BAC administrator user interface, allow you to augment BAC behavior or add support for new device technologies.

Before familiarizing yourself with managing extensions, you should know the RDU extension points that BAC requires. At least one disruption extension must be attached to the associated technology's disruption extension point when disrupting devices on behalf of a batch.

[Table 11-11](#) lists the RDU extension points that BAC requires to execute extensions.

**Table 11-11 Required RDU Extension Points**

| Extension Point                 | Description                                                                                                                                                                                                                                                                                                                         | Use      | Specific to Technology? |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-------------------------|
| Common Configuration Generation | Executed to generate a configuration for a device. Extensions attached to this extension point are executed after the technology-specific service-level selection extension and before the technology-specific configuration generation extensions. The default extensions built into this release do not use this extension point. | Optional | No                      |
| Configuration Generation        | Executed to generate a configuration for a device.                                                                                                                                                                                                                                                                                  | Required | Yes                     |
| Device Detection                | Executed to determine a device technology based on information in the DHCP Discover request packet of the device.                                                                                                                                                                                                                   | Required | No                      |
| Disruption                      | Executed to disrupt a device.                                                                                                                                                                                                                                                                                                       | Optional | Yes                     |
| Publishing                      | Executed to publish provisioning data to an external datastore. The default extensions built into BAC do not include any publishing plug-ins.                                                                                                                                                                                       | Optional | No                      |
| Service-Level Selection         | Executed to select the service level to grant to a device. Extensions attached to this extension point are executed before any common configuration generation extensions and the technology-specific configuration generation extensions.                                                                                          | Optional | Yes                     |

Managing extensions includes:

- [Writing a New Class, page 11-33](#)
- [Installing RDU Custom Extension Points, page 11-34](#)
- [Viewing RDU Extensions, page 11-34](#)



**Note**

You can specify multiple extension points by specifying the extension points in a comma-separated list.

## Writing a New Class

This procedure is included to better illustrate the entire custom extension creation process. You can create many different types of extensions; for the purposes of this procedure, a new Publishing Extension Point is used.

To write the new class:

**Step 1** Create a Java source file for the custom publishing extension, and compile it.

**Step 2** Create a manifest file for the Jar file that will contain the extension class.



**Note**

For detailed information on creating a manifest file and using the command-line JAR tool, refer to Java documentation.

For example:

```
Name: com/cisco/support/extensions/configgeneration
Specification-Title: "DOCSIS TOD synchronization"
Specification-Version: "1.0"
Specification-Vendor: "General Cable, Inc."
Implementation-Title: "Remove the time-servers DHCP option"
Implementation-Version: "1.0"
Implementation-Vendor: "Cisco Systems, Inc."
```



**Note**

Java Jar file manifests contain attributes that are formatted as name-value pairs and support a group of attributes that provide package versioning information. While BAC accepts extension Jar files that do not contain this information, we recommend that you include a manifest with versioning information in the files to track custom RDU extensions.

You can view manifest information from the administrator user interface (via **Servers > RDU > View Regional Distribution Unit Details** page. Detailed information on the installed extension Jar files and the loaded extension class files appears after the Device Statistics section. You can view manifest information from the RDU logs also.

**Step 3** Create the Jar file for the custom extension point.

For example:

```
C:\>jar cm0vf manifest.txt removetimeservers.jar com
added manifest
adding: com/(in = 0) (out= 0)(stored 0%)
adding: com/cisco/(in = 0) (out= 0)(stored 0%)
adding: com/cisco/support/(in = 0) (out= 0)(stored 0%)
```

```

adding: com/cisco/support/extensions/(in = 0) (out= 0) (stored 0%)
adding: com/cisco/support/extensions/configgeneration/(in = 0) (out= 0) (stored 0%)
adding: com/cisco/support/extensions/configgeneration/
RemoveTimeServersExtension.class(in = 4038) (out= 4038) (stored 0%)
C:\>

```




---

**Note** You can give the Jar file any name. The name can be descriptive, but do not duplicate another existing Jar filename.

---

## Installing RDU Custom Extension Points

After a Jar file is created, use the administrator user interface to install it:

---

**Step 1** To add the new Jar file, see [Adding External Files, page 11-27](#).




---

**Note** Select the JAR file type. Use the Browse function to locate the Jar file created in the procedure described in [Writing a New Class, page 11-33](#), and select this file as the Source File. Leaving the External File Name blank assigns the same filename for both source and external files. The external filename is what you will see on the administrator user interface.

---

**Step 2** Click **Submit**.

**Step 3** Return to the RDU Defaults page and note if the newly added Jar file appears in the Extension Point Jar File Search Order field.

**Step 4** Enter the extension class name in the Publishing Extension Point field.




---

**Note** The RDU returns an error if the class name does not exist within the Jar file. This error occurs mostly when replacing a Jar file, if, for example, the class you set up is not found in the replacement Jar file.

---

**Step 5** Click **Submit** to commit the changes to the RDU database.

**Step 6** View the RDU extensions to ensure that the correct extensions are loaded.

---

## Viewing RDU Extensions

You can view the attributes of all RDU extensions directly from the View Regional Distribution Unit Details page. This page displays details on the installed extension Jar files and the loaded extension class files. See [Viewing Regional Distribution Unit Details, page 10-25](#).

# Publishing Provisioning Data

BAC has the capability to publish the provisioning data it tracks to an external datastore in real time. To do this, a publishing plug-in must be developed to write the data to the desired datastore. The Manage Publishing page, shown in [Figure 11-16](#), identifies information such as the plug-in name, its current status (whether it is enabled or disabled), and switch to enable or disable it.

You can enable as many plug-ins as required by your implementation, but remember that the use of publishing plug-ins can decrease system performance.



## Note

BAC does not ship with any publishing plug-ins. You must create your own plug-ins and load them into BAC in the same way as Jar files are (see [Adding External Files](#), page 11-27). Then, manage the plug-ins from the Manage Publishing page. The plug-ins shown in [Figure 11-16](#) are for illustration only.

**Figure 11-16** Manage Publishing Page

| Plug-In                       | Current Status | Enable/Disable Plug-in            |
|-------------------------------|----------------|-----------------------------------|
| <a href="#">TestPublisher</a> | Enabled        | <a href="#">[Disable plug-in]</a> |
| <a href="#">TestPublisher</a> | Disabled       | <a href="#">[Enable plug-in]</a>  |
| <a href="#">TestPublisher</a> | Enabled        | <a href="#">[Disable plug-in]</a> |
| <a href="#">TestPublisher</a> | Enabled        | <a href="#">[Disable plug-in]</a> |
| <a href="#">TestPublisher</a> | Enabled        | <a href="#">[Disable plug-in]</a> |
| <a href="#">TestPublisher</a> | Enabled        | <a href="#">[Disable plug-in]</a> |

129891

## Publishing Datastore Changes

To enable or disable a publishing plug-in:

- Step 1** Choose **Configuration** on the Primary Navigation bar.
- Step 2** Choose **Publishing** on the Secondary Navigation bar.  
The Manage Publishing page appears. This page displays a list of all available database plug-ins and identifies the current status of each.
- Step 3** Click on the appropriate status indicator to enable or disable the required plug-in. Note that as you click the status, it toggles between the two states. See [Figure 11-16](#).

## Modifying Publishing Plug-In Settings

These settings are a convenient way for plug-in writers to store plug-in settings in the RDU for their respective datastore. To modify the publishing plug-in settings:

- Step 1** Choose **Configuration** on the Primary Navigation bar.
- Step 2** Choose **Publishing** on the Secondary Navigation bar, and the Manage Publishing page appears.
- Step 3** Click the link corresponding to the plug-in you want to modify. The Modify Publishing Plug-Ins page appears.

Table 11-12 identifies the fields shown in the Modify Publishing Plug-Ins page.

**Table 11-12** *Modify Publishing Plug-Ins Page*

| Field or Button  | Description                                                                                                                                 |
|------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| Plug-In          | Identifies the publishing plug-in name.                                                                                                     |
| Server           | Identifies the server name on which the datastore resides.                                                                                  |
| Port             | Identifies the port number on which the datastore resides.                                                                                  |
| IP Address       | Identifies the IP address of the server on which the datastore resides. This address is usually specified when the server name is not used. |
| User             | Identifies the user to allow access to the data stored.                                                                                     |
| Password         | Identifies the user's password, which allows access to the data stored.                                                                     |
| Confirm Password | Confirms the password entered above.                                                                                                        |

- Step 4** Enter the required values in the Server, Port, IP Address, User, Password, and Confirm Password fields. These are all required fields and you must supply this information before proceeding.
- Step 5** Click **Submit** to make the changes to the selected plug-in, or click **Reset** to clear all fields on this page.

## Configuring SRV Records in the Network Registrar DNS Server

You must configure the Network Registrar DNS server to operate with the KDC. To set up this configuration, refer to your Network Registrar documentation and these instructions.



### Note

It is recommended that you create a zone name that matches the desired realm name, and that the only DNS record in this special zone (other than the records required by the DNS server to maintain the zone) should be the SRV record for the realm. This example assumes that the desired Kerberos realm is voice.acme.com, and that all other KDC, Network Registrar, and DPE configuration has been performed. The FQDN of the KDC is assumed to be kdc.acme.com.

- Step 1** Start the **nrcmd** CLI (which resides, by default, in the `/opt/nwreg2/local/usrbin` directory), and enter your username and password.

**Step 2** Enter this command to create a zone for the Kerberos realm:

```
nrcmd> zone voice.acme.com create primary <address of nameserver> hostmaster
```

**Step 3** Enter this command to add the SRV record to the new zone:

```
nrcmd> zone voice.acme.com. addRR _kerberos._udp. srv 0 0 88 <address of KDC>
```

**Step 4** Enter these commands to save and reload the DNS server:

```
nrcmd> save
nrcmd> dns reload
```

## Configuring SNMPv3 Cloning on the RDU and DPE for Secure Communication with PacketCable MTAs

BAC lets you enable an external network manager for SNMPv3 access to MTA devices. Additionally, the RDU is capable of performing SNMPV3 operations in a specific MTA.

To enable this capability, set the security key material at the DPEs and RDU. After the key material has been set, the BAC API calls that are used to create cloned SNMPv3 entries are enabled.



**Note**

Enabling this capability impacts provisioning performance.

## Creating the Key Material and Generating the Key

Creating the key material is a two-step process:

1. Run a script command on the RDU.
2. Run a CLI command on the DPE.



**Note**

This shared secret is not the same shared secret as the CMTS or the BAC shared secrets.

To create the key material:

**Step 1** From the *BPR\_HOME/rdu/bin* directory, run this script on the RDU:

```
generateSharedSecret.sh password
```

where *password* is any password, from 6 to 20 characters, that you create. This password is then used to generate a 46-byte key. This key is stored in a file, called *keymaterial.txt*, that resides in the *BPR\_HOME/rdu/conf* directory.

**Step 2** Run the **packetcable snmp key-material** DPE CLI command, with the *password* used in Step 1 to generate that key, on all DPEs for which this voice technology is enabled. This command generates the same 46-byte key on the DPE and ensures that the RDU and DPEs are synchronized and can communicate with the MTA securely.

# Automatic FQDN Generation

When configuring the PacketCable voice technology, a fully qualified domain name (FQDN) must reside in the BAC database for each voice device, because the KDC queries the registration server for that FQDN. The BAC automatic FQDN generation feature is not limited to use by any single voice technology; it can be used by any BAC technology.

## Automatically Generated FQDN Format

An automatically generated FQDN in BAC follows this format:

```
prefixhtype-hlen-aa-bb-cc-dd-ee-ffsuffix.domain
```

- *prefix*, *suffix*, and *domain*—Identify the information that you set from the BAC administrator user interface or the provisioning API.



**Note** In the sample FQDN used here, *prefix1,6,aa-bb-cc-dd-ee-ffsuffix* is the generated hostname and *domain* is the domain name.

- 1,6,aa-bb-cc-dd-ee-ff—Identifies the device MAC address.

The entry of a prefix and suffix property is optional. If you do not specify these properties, and a hostname is not specified during PacketCable MTA provisioning and, if neither the prefix nor suffix property is defined in the BAC property hierarchy, the device MAC address followed by the domain name are used as the generated FQDN.

### For example:

A device with the MAC address **1,6,aa:bb:cc:dd:ee:ff** will have this FQDN generated:

```
1-6-aa-bb-cc-dd-ee-ff.domain
```

When configuring for PacketCable and other technologies, the domain name property must also be configured. If you do not specify a domain name while provisioning a PacketCable MTA, the BAC property hierarchy is searched and, if it is not found, the MTA is not provisioned. If you do specify the domain name during MTA provisioning, that domain name is used regardless of the domain name property that is specified in the BAC property hierarchy.

## Properties for Automatically Generated FQDNs

Properties can be defined at any acceptable point in the BAC property hierarchy. You can use the System Defaults, Technology Defaults, DHCP Criteria, or Class of Service to accomplish this, and you can also do this at the device level.



## FQDN Validation

There are a few things to consider when entering the information that is used to generate an FQDN. These include:

- Use only valid alphanumeric characters in the generated FQDN.
- Keep the length of each label (characters between the dots in the generated FQDN) to fewer than 63 characters.
- Do not allow the overall length of the generated FQDN to exceed 254 characters.

**Note**

---

The FQDN supports host and domain names as per RFC1035.

---

## Sample Automatic FQDN Generation

This section provides an example of creating an automatically generated FQDN.

- 
- Step 1** Choose the appropriate Class of Service, and set the `/fqdn/domain` property value to the DNS domain for all devices using this Class of Service. For the purposes of this example, assume that the domain in use is **pctest.com**, and that you want to provision a set of PacketCable devices into that domain.

**Note**

---

If a domain is not specified, devices in the Class of Service will not receive a DHCP configuration from BAC.

---

- Step 2** Click **Submit**.

In this example, a device with MAC address 1,6,aa:bb:cc:dd:ee:ff will yield an automatically generated FQDN of 1-6-aa-bb-cc-dd-ee-ff.pctest.com. Additionally, the Automatic FQDN Generation field should be enabled in the device's default configuration.

---





## CHAPTER 12

# Configuring and Using the Sample User Interface

---

Broadband Access Center (BAC) supports a sample user interface (SUI) that you can use for self-provisioning and preprovisioning. The SUI demonstrates the basic functionality of BAC 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 to be used as a deployment vehicle. It is for demonstration only.

---

## What is the Sample User Interface?

BAC provides a sample workflow application that manages the automated provisioning of devices on the network, and an administrator interface that provides administrators with the basic functions to manage the accounts that are maintained in BAC.

In the SUI, data is managed in two distinct ways: registration of devices on the network and the accounting of results from those registrations. For example, pages that permit the complete self-provisioning of new cable modems by using credit card information must be capable of handling the automated billing for services together with device tracking. While the SUI does not track accounting information, it allows each device to be associated with an owner identifier (Owner ID). For example, the association with an Owner ID allows objects stored in BAC to be related to external objects, such as billing account systems. In this sample workflow, the Owner ID is used as the account number, but it is not actually related to any external data. The Owner ID associated on device objects in BAC can be any external string used to group devices.

The SUI uses shortcuts to provide an interface that can support functions such as accounting, without actually needing an external accounting entity. Accounting information is stored on the modem object as custom properties. When viewing an account, the modem is found using the Owner ID and then the account data is retrieved from the modem.

BAC supports two distinct methods of managing devices: the Standard mode and the Promiscuous mode.

- In the Standard mode, modems and computers are tracked individually.
- In the Promiscuous mode, only the modems are tracked regardless of how many computers exist on the other side of the modem. When the Promiscuous mode is enabled, computers receive access only if they are behind a provisioned modem.

## Accessing the Sample User Interface

The SUI includes subscriber and administrator interfaces. You can change the subscriber interface flow to support preprovisioning or self-provisioning, tracking of customers, and tracking of devices being given access.

To access the SUI, enter:

```
http://machine_name:port_number/sampleui
```

- *machine\_name*—Identifies the computer on which the RDU is running.



### Note

To access the SUI via HTTP over SSL, also known as HTTPS, enter:  
`https://machine_name:port_number/`

- *port number*—Identifies the computer port on which the server side of the administrator application runs. The default port number is:
  - 8100 for HTTP over TCP
  - 8443 for HTTP over SSL

Before you use the SUI, examine the `sampleui.properties` file. This file contains a variety of controls that specify the behavior of the interface. To view the default `sampleui.properties` file, refer to [Sample sampleui.properties File, page 12-9](#). You can open this file, and change its content to perform different functions, using any text editor. After you save the changes, restart the SUI so that all changes take effect.

## Sample User Interface Configuration Options

You can configure the SUI using the options described in this section. Modifying these options forces the SUI to behave in different flows. The intention of these options is to represent the majority of your requirements. These options are controlled by settings that exist in BAC or are defined in the `sampleui.properties` file. For additional information, see [Sample sampleui.properties File, page 12-9](#).

### Class of Service

A Class of Service is defined in the interface configuration file and also in the normal service definition within BAC. The Class of Service within the SUI also references the intended DHCP criteria to be used for the devices and has a description for presentation in the interface. For example, if you chose a Class of Service called Blue, the SUI could translate that into a BAC Class of Service called Gold and a DHCP criteria called residential-provisioned. When you launch the SUI, it attempts to verify if the referenced Class of Service is already defined.

### Promiscuous Mode

The Promiscuous mode is the behavior involving the tracking of computers. When the Promiscuous mode is enabled, a computer automatically receives a provisioned configuration when it is plugged in behind a provisioned cable modem. Further, in the Promiscuous mode, computers are not asked for registration information. However, when this mode is disabled (a situation known as the Standard mode),

the SUI lets users register their computers. This mode includes the optional selection of an Internet service provider (ISP) for each computer. This information is maintained within BAC; you can access it from the RDU Defaults page in the administrator user interface.

**Note**

You must restart the SUI after you change the Promiscuous mode.

## Selecting an Internet Service Provider

You can select an ISP individually, for each computer registered with an account, whenever the Standard mode PC registration (non-Promiscuous) mode is used. Selecting an ISP has the same effect as choosing the DHCP criteria assigned to the computer. This setting is configured within the interface's configuration file. If there is a single ISP, the ISP-selection controls are bypassed when moving through the subscription interface.

## Using the Technician Login

You can use the technician login to demonstrate a provisioning flow whereby a technician brings a cable modem to a customer's home and plugs it in. The Technician Login page appears, using which you authenticate the technician in the system before proceeding with the provisioning of the cable modem on the network.

If authentication is disabled, but technician provisioning is enabled, the demonstration is for a self-provisioning flow.

## Administrative Access Levels

Administrators that can access the SUI administrator interface are configured in the interface configuration file. You can use four types of administrators within the SUI:

- full—Has complete access to create and delete accounts and manage devices in the interface.
- createonly—Has access only to create new accounts using the interface.
- readonly—Has access only to view accounts that have been created in the system.
- tech—Has access only to log in through the technician interface. This is for autoprovisioning devices from the customer premises.

## Subscriber Provisioning Examples

This section describes various workflows that are presented while using the SUI. Having the Promiscuous mode enabled or disabled has a significant effect on the behavior of the SUI. Consequently, the flows in this section are identified separately.

## Standard Customer Premise Equipment Registration

This section describes provisioning activities when the system is in the standard and non-Promiscuous modes of operation. These provisioning activities are discussed in:

- [Provisioning a New Cable Modem and a New Computer, page 12-4](#)
- [Provisioning a New Computer with an Existing Cable Modem, page 12-4](#)
- [Altering an Existing Computer ISP, page 12-5](#)

### Provisioning a New Cable Modem and a New Computer

When a new modem and new computer are connected to a network, and you bring a web browser online, you are redirected to the provisioning interface.

- 
- Step 1** The SUI checks the `sampleui.properties` file to determine if technician provisioning is enabled:
- If this feature is enabled, the SUI continues with the next step.
  - If it is disabled, an error page appears, stating that the modem is not registered and the customer should contact the MSO to register their cable modem with the system.
- Step 2** The SUI checks the `sampleui.properties` file to determine if technician authentication is required:
- If this feature is not required, the modem registration screen appears, and you can enter account details to be registered with the system.
  - If the feature is required, the modem registration page appears, and you can enter your technician username and password, and the account details to be registered with the system.
- Step 3** A computer registration page appears. You can use this page to register the computer at the same time the modem is registered. This page also identifies that the modem registration has been successful.
- Step 4** The SUI checks the `sampleui.properties` file to determine if the optional ISP selection is enabled.
- If it is enabled, a drop-down list with available ISPs appears for you to select the appropriate ISP.
  - If it is disabled, no ISP is available for selection.
- Step 5** Click **Register This Computer** and a message appears, stating that the computer is successfully registered with the network.
- 

### Provisioning a New Computer with an Existing Cable Modem

When an existing modem and new computer are connected to the network and you bring a web browser online, you are redirected to the provisioning interface.

- 
- Step 1** The SUI displays the computer registration page. From here, you can register the computer on the network.
- Step 2** The SUI checks the `sampleui.properties` file to determine if the optional ISP selection is enabled.
- If it is enabled, a drop-down list with available ISPs appears for you to select the appropriate ISP option
  - If it is disabled, no ISP is available for selection.

- Step 3** Click **Register This Computer** and a message appears, stating that the computer is successfully registered with the network.
- 

## Altering an Existing Computer ISP

When an existing modem and existing computer are connected to the network and you bring a web browser online, you can browse the network. They then direct the browser to the provisioning interface.

- Step 1** The SUI determines whether or not the optional ISP selection feature is enabled.
- If it is enabled, a drop-down list with available ISPs appears for you to select the appropriate ISP option.
  - If it is disabled, no ISP is available for selection and a message appears, stating this computer is already registered on the system.
- Step 2** Click **Register This Computer** and a message appears, stating that the computer is successfully registered with the network.
- 

## Promiscuous Customer Premises Equipment Registration

This section describes equipment registration using the SUI. These provisioning activities are discussed in these sections:

- [Provisioning a New Cable Modem and a New Computer, page 12-5](#)
- [Provisioning an Existing Cable Modem and a New Computer, page 12-6](#)

## Provisioning a New Cable Modem and a New Computer

When a new modem and new computer are connected to the network and then you bring a web browser online, you are redirected to the provisioning interface.

- Step 1** The SUI checks the `sampleui.properties` file to determine if technician provisioning is enabled:
- If this feature is enabled, the SUI continues with the next step.
  - If it is disabled, an error page appears, stating that the modem is not registered and that the customer should contact their MSO to register their cable modem with the system.
- Step 2** The SUI checks the `sampleui.properties` file to determine if technician authentication is required:
- If this feature is not required, the modem registration screen appears for you to enter account details to be registered with the system.
  - If the feature is needed, the modem registration page appears for you to enter your technician username and password, and the account details to be registered with the system.
- Step 3** A message appears, stating that the modem and computer are successfully registered with the network.
-

## Provisioning an Existing Cable Modem and a New Computer

When an existing modem and new computer are connected to the network and then you bring a web browser online, you can browse the network. Once you are able to browse the network, the existing modem and new computer must direct the browser to the provisioning interface. After being directed, a message appears to indicate that the cable modem and computer are registered on the system.

# Administrator Provisioning Examples

This section identifies some examples that illustrate the use of the SUI in performing account maintenance and account searches. The components of each SUI page that appear only when certain permissions have been assigned are identified with an *if applicable* note appended to the end of the component name.

## Searching for Accounts

This section explains how to perform account searches using the SUI. These search activities are discussed:

- [Searching by Account Number, page 12-6](#)
- [Searching by IP Address, page 12-6](#)
- [Searching by MAC Address, page 12-6](#)

### Searching by Account Number

You can search for an account, using an account number, after logging in. You specify the account number to search for and, if found, the account details appear. If the account is not found, an error message appears, stating that the account number does not exist in the system.

### Searching by IP Address

You can search for an account, using an IP address, after logging in. You specify the IP address of a computer or modem currently provisioned by BAC and, if found, the owner is checked for the device to determine what account should appear.

If a valid account is found, full account details for the device appear. If a valid device is found, but not a valid account, the current MAC address and IP address of the device appear at the bottom of the search page.

If the device cannot be found, an error appears, indicating the search did not find a matching device.

### Searching by MAC Address

You can search for an account, using a MAC address, after logging in. You specify the MAC address of a computer or modem currently provisioned by BAC and, if found, the owner is checked for the device to see what account should appear.



If a valid account is found, full account details for the device appear. If a valid device is found, but not a valid account, the current MAC address and IP address of the device appear at the bottom of the search page. If the device cannot be found, an error appears, indicating that the search did not find a matching device.

## Maintaining Accounts

This section explains how to maintain accounts using the SUI. These maintenance activities are discussed in these sections:

- [Registering a New Account, page 12-7](#)
- [Managing Class of Service, page 12-7](#)
- [Managing Cable Modems, page 12-8](#)
- [Managing Computers, page 12-8](#)

### Registering a New Account

You can use this workflow to register a new account, and modem, with the SUI. First, log in to the system and then:

- 
- Step 1** The SUI displays a page that shows the search options (if applicable) to choose from. Click **Create a New Account**.
  - Step 2** The SUI displays a page that lets you enter the account number, the MAC address of the cable modem, and the Class of Service applicable to the account. Once you enter and submit the information, the account is created.
  - Step 3** Once the account is created, a new account creation page appears. Using this page, you can enter the details required to register multiple accounts with the system.
- 

### Managing Class of Service

You can use this workflow to change the Class of Service on an account and disable the cable modem. Log in to the system and then:

- 
- Step 1** The SUI displays a page that shows the search options (if applicable) to choose from. You can search for the account by using the account number, IP address, or MAC address as the search criteria.
  - Step 2** The SUI displays a page that contains all the information on the account, including the currently selected Class of Service, whether the cable modem is enabled, the account owner information, and the list of registered computers. Select the appropriate Class of Service from the drop-down list.
  - Step 3** Click **Update**.
- The same page refreshes with the account information; note the change in the Class of Service.
-

## Managing Cable Modems

You can use this workflow to change the modem that is currently associated with an account. You can also update account details, such as username, using this workflow. Log in to the system and then:

- 
- Step 1** The SUI displays a page that shows the search options (if applicable) to choose from. You can search for the account by using the account number, IP address, or MAC address as the search criteria.
  - Step 2** The SUI displays a page that contains all the information on the account, including the currently selected Class of Service, whether the cable modem is enabled, the account owner information, and the list of registered computers. Enter the new MAC address for the account's cable modem.
  - Step 3** Click **Update**. The page refreshes with the account information; note the change in the MAC address.
- 

## Managing Computers

You can use this workflow to unregister computers that were previously registered using the Subscriber portion of the SUI. This workflow applies only when Standard mode PC registration is used. Log in to the system and then:

- 
- Step 1** The SUI displays a page that shows the search options (if applicable) to choose from. You can search for the account by using the account number, IP address, or MAC address as the search criteria.
  - Step 2** The SUI displays a page that contains all the information on the account, including the currently selected Class of Service, whether the cable modem is enabled, the account owner information, and the list of registered computers. Determine which computer you want to unregister and click the corresponding **Delete** button.

The page refreshes with the most current account information; note that the selected computer is removed from the list.

---

## Deleting an Account

You can use this workflow to delete an account that was registered using the SUI. Log in to the system and then:

- 
- Step 1** The SUI displays a page that shows the search options (if applicable) to choose from. You can search for the account by using the account number, IP address, or MAC address as the search criteria.
  - Step 2** The SUI displays a page that contains all the information on the account, including the currently selected Class of Service, whether the cable modem is enabled, the account owner information, and the list of registered computers. Click **X**, at the Delete account field from the system prompt.
  - Step 3** The SUI displays the same page containing all the account information. A prompt appears, next to the Delete button, asking for confirmation before proceeding.
  - Step 4** Click **X**, at the Delete account field from the system prompt, again to confirm deletion of the account.
  - Step 5** The SUI displays the original search page, showing that the account is deleted.
-

# Sample sampleui.properties File

This section identifies the contents of a sampleui.properties file. This file resides in the *BAC\_home/rdu/tomcat/webapps/sampleui/WEB-INF/classes* directory.

```
(C) Copyright 2001-2007 by Cisco Systems, Inc.
This program contains proprietary and confidential information.
All rights reserved. This software shall not be used by an party
except by prior written consent of Cisco Systems.
#
DO NOT CHANGE. This is the version of the properties file which
is used during execution and for system updates.
#
version=1.2
#####
System connection information.
#####
#
BPR RDU connection information
#
adminuser=admin
adminpass=admin
host=localhost
port=49187
#####
Provisioning configuration parameters
#####
#
Administrator access credentials
#
3 levels of access: full, createonly, and readonly
full -- can create and read accounts
createonly -- can only create new accounts
readonly -- can only read accounts
tech -- technician access for autoprovisioning modems
(The default access level is 'full')
#
The user.number must equal the number of user accounts
being tracked. If the number is 4, there must exist
entries for users 1-4.
#
user.number=4
user.1.name=admin
user.1.password=changeme
```

```

user.1.access=full
user.2.name=config
user.2.password=changeme
user.2.access=createonly
user.3.name=monitor
user.3.password=changeme
user.3.access=readonly
user.4.name=tech
user.4.password=changeme
user.4.access=tech
#
Indicates if promiscuous-mode is enabled
#
promiscuous-mode is a special mode of BPR in which computers are not
tracked by the provisioning system. (true/false)
#
This is controlled by setting the RDU Defaults option:
ModemKeys.PROMISCUOUS_MODE_ENABLED
#
Indicates if technician autoprovisioning is enabled
#
This mode allows a technician to provision a new account in the field
without requiring the MAC address to be pre-registered.
#
techprovisioning.enabled=true
#
Indicates if technician username/password is required for modem
registration
#
If technician username/password is not required, then this
demo can be used to simulate modem self-registration
#
techprovisioning.authentication=false
#
Unprovisioned configuration (for disabled modems)
#
The client class should match the unprovisioned client class configured
in CNR. The service must be unique (i.e. NOT have the same name of
any of the services specified below).
#
These values are controlled by setting the DOCSIS Defaults options, use:
TechnologyDefaultsKeys.DOCSIS_DEFAULT_CLASS_OF_SERVICE
TechnologyDefaultsKeys.DOCSIS_DEFAULT_DHCP_CRITERIA
TechnologyDefaultsKeys.COMPUTER_DEFAULT_DHCP_CRITERIA

```

```
#
Classes of Service for Modems
#
The DHCP criterias specified here must match valid
DHCP criterias specified in the RDU. If promiscuous
mode is enabled, you must specify CPE DHCP criterias.
#
The service.number must equal the number of services
being tracked. If the number is 3, there must exist
entries for services 1-3.
#
service.number=3
service.1.name=gold
service.1.title=1.5Mb/s Lightning Fast!
service.1.dhcpcriteria=ccProvisionedDOCSISModem
service.1.cpedhcpcriteria=provcpetagProvisionedPromiscuousCpe
service.2.name=silver
service.2.title=512Kb/s Power User
service.2.dhcpcriteria=ccProvisionedDOCSISModem
service.2.cpedhcpcriteria=provcpetagProvisionedPromiscuousCpe
service.3.name=bronze
service.3.title=64kb/s Economy Service
service.3.dhcpcriteria=ccProvisionedDOCSISModem
service.3.cpedhcpcriteria=provcpetagProvisionedPromiscuousCpe
#
ISPs for Computers
#
The computerisp.number must equal the number of ISPs
available. If the number is 1, there must exist
entries for ISPs 1-1.
#
computerisp.number=1
computerisp.1.name=msonet
computerisp.1.title=MSO.net Services
computerisp.1.dhcpcriteria=ccProvisionedComputer
#
Default COS, DHCPCriteria and CPE DHCPCriteria
that modem and computers are placed in when modem's
access is disabled through administrator UI
#
Appropriate DOCSISClassOfService and DHCPCriteria objects have
to be pre-created in the RDU
#
disabled.modem.cpedhcpcriteria=disabled-computer -- defined a
```

## ■ Sample sampleui.properties File

```
DHCPCriteria which computer's behind the modem get when modem's
access is disabled
#
disabled.modem.cos=disabled
disabled.modem.dhcpcriteria=disabled-modem
disabled.modem.cpedhcpcriteria=disabled-computer
```



# CHAPTER 13

## Support Tools and Advanced Concepts

This chapter contains information on, and explains the use of, tools that help you maintain Broadband Access Center (BAC) as well as speed and improve the installation, deployment, and use of this product.

This chapter discusses:

- [Using the RDU Log Level Tool, page 13-2](#)
- [Using the PKCert.sh Tool, page 13-5](#)
- [Using the KeyGen Tool, page 13-11](#)
- [Using the changeNRProperties.sh Tool, page 13-13](#)
- [Using the snmpAgentCfgUtil.sh Tool, page 13-15](#)
- [Using the disk\\_monitor.sh Tool, page 13-21](#)
- [Troubleshooting Devices by MAC Address, page 13-21](#)



### Note

This section contains many examples of tool use. In many cases, the tool filenames include a path specified as *BAC\_home*. This indicates the default directory location.

## BAC Tools

BAC provides automated tools that you use to perform certain functions more efficiently. [Table 13-1](#) lists the various tools that this BAC release supports.

**Table 13-1**     *BAC Tools*

| Tool                       | Description                                                                                                           | Refer..                                                                         |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|
| Configuration File Utility | Used to test, validate, and view BAC template and configuration files.                                                | <a href="#">Using the Configuration File Utility, page 8-27</a>                 |
| BAC Process Watchdog       | Interacts with the BAC watchdog daemon to observe the status of the BAC system components, and stop or start servers. | <a href="#">Using the BAC Process Watchdog from the Command Line, page 2-14</a> |

**Table 13-1** BAC Tools (continued)

| Tool                                       | Description                                                                                                                                              | Refer...                                                         |
|--------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------|
| RDU Log Level Tool                         | Sets the log level of the RDU, and enables or disables debugging log output.                                                                             | <a href="#">Using the RDU Log Level Tool, page 13-2</a>          |
| PacketCable Certificates Tool              | Installs, and manages, the KDC certificates that are required by the KDC for its operation.                                                              | <a href="#">Using the PKCert.sh Tool, page 13-5</a>              |
| KeyGen Tool                                | Generate PacketCable service keys.                                                                                                                       | <a href="#">Using the KeyGen Tool, page 13-11</a>                |
| Changing Network Registrar Properties Tool | Used to change key configuration properties used by BAC extensions that are incorporated into the Network Registrar DHCP server.                         | <a href="#">Using the changeNRProperties.sh Tool, page 13-13</a> |
| SNMP Agent Configuration Tool              | Manages the SNMP agent.                                                                                                                                  | <a href="#">Using the snmpAgentCfgUtil.sh Tool, page 13-15</a>   |
| Disk Space Monitoring Tool                 | Sets threshold values for one or more file systems. When these thresholds are surpassed, an alert is generated until additional disk space is available. | <a href="#">Using the disk_monitor.sh Tool, page 13-21</a>       |

## Using the RDU Log Level Tool

Use the RDU log level tool to change the current log level of the RDU from the command line, using the **setLogLevel.sh** command. This tool resides in the *BAC\_home/rdu/bin* directory. [Table 13-2](#) identifies the available severity levels and the types of messages written to the log file when enabled.

**Table 13-2** Logging 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.                     |



**Table 13-2** Logging Levels (continued)

| Log Level      | Description                                                                                                                                         |
|----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 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.                                                           |
| <b>Note</b>    | 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. |

We recommend that you keep the RDU severity level at the Warning level to help maintain a steady operations state. The Information level is recommended to be used with caution if you need to maintain steady state performance during debug operations. You should exercise caution when running with the Information level because this creates a great number of log entries, which in itself can adversely impact performance.

**Note**

The RDU process has to be up to execute the log level tool. Also, you must be a privileged user to run this tool by using the **setLogLevel.sh** command.

**Syntax Description**

```
setLogLevel.sh -[0..6] [-help] [-show] [-default] [-debug]
```

- **-[0..6]**—Identifies the severity level to be used. For a list of available levels, see [Table 13-2](#).
- **-help**—Displays help for the tool.
- **-show**—Displays the current severity level set for the RDU server.
- **-default**—Sets the RDU to the installation default level 5 (notification).
- **-debug**—Sets an interactive mode to enable or disable tracing categories for the RDU server.

**Note**

You should only enable the debug settings that the Cisco support staff recommends.

You can also use this tool to perform these functions:

- [Setting the RDU Log Level, page 13-3](#)
- [Viewing the Current Log Level of RDU, page 13-4](#)

## Setting the RDU Log Level

You can use this tool to change the logging level from one value to another value. The following example illustrates how to set the RDU logging level to the warning level, as indicated by the number 4 in the **setLogLevel.sh** command. The actual log level set is not important for the procedure, it can be interchanged as required.

The example described in this section assumes that the RDU server is up, the username for the RDU is **admin**, and the password is **changeme**.

To set the RDU logging level:

---

**Step 1** Change directory to *BAC\_home/rdu/bin*.

**Step 2** Run the RDU log level tool using this command:

```
setLogLevel.sh 4
```

This prompt appears:

Please type RDU username:

**Step 3** Enter the RDU username. In this example, the default username (**admin**) is used.

Please type RDU username: **admin**

This prompt appears:

Please type RDU password:

**Step 4** Enter the RDU password for the RDU. In this example, the default password (**changeme**) is used.

Please type RDU password: **changeme**

This message appears to notify you that the log level has been changed. In this example, the level was 5, for notification, and is now 4, for warning.

```
RDU Log level was changed from 5 (notification) to 4 (warning).
```

---

## Viewing the Current Log Level of RDU

You can use this tool to view the RDU log and determine which logging level is configured before attempting to change the value.

The example described in this section assumes that the RDU server is up, the username for the RDU is **admin**, and the password is **changeme**.

To view the current logging level of the RDU:

---

**Step 1** Change directory to *BAC\_home/rdu/bin*.

**Step 2** Run this command:

```
setLogLevel.sh -show
```

This prompt appears:

Please type RDU username:

**Step 3** Enter the RDU username (**admin**) and press **Enter**.

Please type RDU username: **admin**

This prompt appears:

Please type RDU password:

**Step 4** Enter the RDU password (**changeme**) and press **Enter**.

Please type RDU password: changeme

This message appears:

```
The logging is currently set at level: 4 (warning)
```

```
All tracing is currently disabled.
```

## Using the PKCert.sh Tool

The PKCert tool creates the certificate and the corresponding private key that the KDC requires. It also installs the CableLabs service provider certificates as certificate files, similar to those shown below.

- Cablelabs\_Service\_Provider\_Root.cer
- Service\_Provider.cer
- Local\_System.cer
- KDC.cer

This tool also allows you to verify certificate chains and copy and rename a certificate chain to the names required by the KDC.



**Note**

This tool is available only when the KDC component is installed.

## Running the PKCert Tool

Run the PKCert tool by executing the PKCert.sh command, which resides by default in the `BAC_home/kdc` directory.

### Syntax Description

**PKCert.sh** *function option*

- *function*—Identifies the function to be performed. You can choose:
  - **-c**—Creates a KDC certificate. See [Creating a KDC Certificate, page 13-6](#).
  - **-v**—Verifies and normalizes the PacketCable certificate set. See [Validating the KDC Certificates, page 13-7](#).
  - **-z**—Sets the log level for debug output that is stored in the `pkcert.log` file. See [Setting the Log Level for Debug Output, page 13-8](#).



**Note**

If you have trouble in using these options, specify `-?` to display available help information.

- *option*—Implements optional functions, depending on the function you selected.

## Creating a KDC Certificate

To create the KDC certificate:

**Step 1** Change directory to `/opt/CSCObpr/kdc`.

**Step 2** Run the PKCert.sh tool using this syntax:

```
PKCert.sh -s dir -d dir -c cert -e -r realm -a name -k keyFile [-n serial#] [-o]
```

- **-s dir**—Specifies the source directory
- **-d dir**—Specifies the destination directory
- **-c cert**—Uses the service provider certificate (DER encoded)
- **-e**—Identifies the certificate as a Euro PacketCable certificate
- **-r realm**—Specifies the Kerberos realm for the KDC certificate
- **-a name**—Specifies the DNS name of the KDC
- **-k keyFile**—Uses the service provider private key (DER encoded)
- **-n serial#**—Sets the certificate serial number
- **-o**—Overwrites existing files

When a new certificate is created and installed, the new certificate identifies the realm in the subject alternate name field. The new certificate is unique to its current environment in that it contains the:

- KDC realm.
- DNS name associated with this KDC that the Media Terminal Adapter (MTA) will use.

### Examples

```
./PKCert.sh -c "-s . \
> -d /opt/CSCObpr/kdc/solaris/packetcable/certificates \
> -k CLCerts/Test_LSCA_privkey.der \
> -c CLCerts/Test_LSCA.cer \
> -r PCTEST.CISCO.COM \
> -n 100 \
> -a kdc.pctest.cisco.com \
> -o"
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: .
Destination Directory: /opt/CSCObpr/kdc/solaris/packetcable/certificates
Private Key File: CLCerts/Test_LSCA_privkey.der
Certificate File: CLCerts/Test_LSCA.cer
Realm: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC_private_key.pkcs8
File written:
/opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC_private_key_proprietary.
File written: /opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC_PublicKey.der
File written: /opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC.cer
```

```
KDC Certificate Successfully Created at
/opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC.cer
```

Using this command creates the files `/opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC.cer` and `/opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC_private_key.pkcs8`. The KDC certificate will have a realm set to `PCTEST.CISCO.COM`, a serial number set to 100, and the fully qualified domain name (FQDN) of the KDC server set to `kdc.pctest.cisco.com`.

## Validating the KDC Certificates

This command examines all files in the source directory specified and attempts to identify them as X.509 certificates. If legitimate X.509 certificates are found, the files are properly renamed and copied to the destination directory. An error is generated when more than one legitimate chain of certificates for a particular purpose (service provider or device) is identified. If this occurs, you must remove the extra certificate from the source directory and run the command again.



### Note

When you enter the **PKCert.sh -v -?** command, usage instructions for validating KDC certificates by using the PKCert tool appear.

To validate the KDC certificate:

**Step 1** Change directory to `/opt/CSCObpr/kdc`.

**Step 2** Run the PKCert.sh tool using this syntax:

```
PKCert.sh -v -s dir -d dir -r dir
```

- **-s dir**—Specifies the source directory
- **-d dir**—Specifies the destination directory
- **-o**—Overwrites any existing files
- **-r dir**—Specifies the reference certificate directory

Verification is performed against reference certificates built into this package. If you specify the **-d** option, the certificates are installed in the target directory with name normalization.

### Examples

```
./PKCert.sh -v \
> "-s /opt/CSCObpr/kdc/TestCerts
> -d /opt/CSCObpr/kdc/solaris/packetcable/certificates \
> -o"
Pkcrt Version 1.0
Logging to pkcert.log
Output files will overwrite existing files in destination directory

Cert Chain(0) Chain Type: Service Provider
[Local File] [Certificate Label] [PacketCable
Name]
CableLabs_Service_Provider_Root.cer CableLabs_Service_Provider_Root.cer
Service_Provider.cer Service_Provider.cer
Local_System.cer Local_System.cer
KDC.cer KDC.cer
```

```

Cert Chain(1) Chain Type: Device
[Local File] [Certificate Label] [PacketCable
Name]
MTA_Root.cer MTA_Root.cer
File written:
/opt/CSCObpr/kdc/solaris/packetcable/certificates/CableLabs_Service_Provider_Root.cer
File written: /opt/CSCObpr/kdc/solaris/packetcable/certificates/Service_Provider.cer
File written: /opt/CSCObpr/kdc/solaris/packetcable/certificates/Local_System.cer
File written: /opt/CSCObpr/kdc/solaris/packetcable/certificates/KDC.cer

Service Provider Certificate Chain Written to Destination Directory
/opt/CSCObpr/kdc/solaris/packetcable/certificates

File written: /opt/CSCObpr/kdc/solaris/packetcable/certificates/MTA_Root.cer

Device Certificate Chain Written to Destination Directory
/opt/CSCObpr/kdc/solaris/packetcable/certificates

```

## Setting the Log Level for Debug Output

This command enables you to set the log level for debug output that is logged in `pkcert.log`, which resides in `BAC_home/kdc`. You can use the data in the log file to troubleshoot any problems that may have occurred while performing the requested tasks.

To set the log level for debug output:

---

**Step 1** Change directory to `/opt/CSCObpr/kdc`.

**Step 2** Run the PKCert.sh tool using this syntax:

```
PKCert.sh -s dir -d dir -k keyFile -c cert -r realm -a name -n serial# -o {-z error | info | debug}
```

- **-s dir**—Specifies the source directory
  - **-d dir**—Specifies the destination directory
  - **-k keyFile**—Uses the service provider private key (DER encoded)
  - **-c cert**—Uses the service provider certificate (DER encoded)
  - **-r realm**—Specifies the Kerberos realm for the KDC certificate
  - **-a name**—Specifies the DNS name of the KDC
  - **-n serial#**—Sets the certificate serial number
  - **-o**—Overwrites existing files
  - **-z**—Sets the log level for debug output that is stored in the `pkcert.log` file. The values you can choose are:
    - **error**—Specifies the logging of error messages.
    - **info**—Specifies the logging of informational messages.
    - **debug**—Specifies the logging of debug messages. This is the default setting.
-

**Examples****Example 1**

In this example, the log level is set for collecting error messages.

```
./PKCert.sh -c "-s /var/certsInput
> -d /var/certsOutput
> -k /var/certsInput/Local_System.der
> -c /var/certsInput/Local_System.cer
> -r PCTEST.CISCO.COM
> -n 100
> -a kdc.pctest.cisco.com
> -o -z error"
Pkcrt Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
Setting debug to error
WARNING - Certificate File will be overwritten
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e.
/opt/CSCObpr/kdc/solaris/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObpr/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObpr/kdc/solaris)
```

**Example 2**

In this example, the log level is set for collecting information messages.

```
./PKCert.sh -c "-s /var/certsInput
> -d /var/certsOutput
> -k /var/certsInput/Local_System.der
> -c /var/certsInput/Local_System.cer
> -r PCTEST.CISCO.COM
> -n 100
> -a kdc.pctest.cisco.com
> -o -z info"
INFO [main] 2007-05-02 06:32:26,280 (PKCert.java:97) - Pkcrt Version 1.0
Pkcrt Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: PCTEST.CISCO.COM
Serial Number: 100
DNS Name of KDC: kdc.pctest.cisco.com
Setting debug to info
INFO [main] 2007-05-02 06:32:26,289 (PKCCreate.java:69) - PKCCreate startup
WARNING - Certificate File will be overwritten
INFO [main] 2007-05-02 06:32:26,291 (PKCCreate.java:341) - WARNING - Certificate File
will be overwritten
```

```

SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

```

```

Copy KDC.cer to the KDC certificate directory (i.e.
/opt/CSCObpr/kdc/solaris/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObpr/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObpr/kdc/solaris)

```

### Example 3

In this example, the log level is set for debugging.



**Note** The sample output has been trimmed for demonstration purposes.

```

./PKCert.sh -c "-s /var/certsInput
> -d /var/certsOutput
> -k /var/certsInput/Local_System.der
> -c /var/certsInput/Local_System.cer
> -r PCTEST.CISCO.COM
> -n 100
> -a kdc.pctest.cisco.com
> -o -z debug"
INFO [main] 2007-05-02 06:32:06,029 (PKCert.java:97) - Pkcert Version 1.0
Pkcert Version 1.0
Logging to pkcert.log
Source Directory: /var/certsInput
Destination Directory: /var/certsOutput
Private Key File: /var/certsInput/Local_System.der
Certificate File: /var/certsInput/Local_System.cer
Realm: IPFONIX.COM
Serial Number: 100
DNS Name of KDC: bacdev3-dpe-4.cisco.com
Setting debug to debug
INFO [main] 2007-05-02 06:32:06,038 (PKCCreate.java:69) - PKCCreate startup
WARNING - Certificate File will be overwritten
INFO [main] 2007-05-02 06:32:06,039 (PKCCreate.java:341) - WARNING - Certificate File
will be overwritten
DEBUG [main] 2007-05-02 06:32:06,054 (PKCert.java:553) - Characters Read: 1218
DEBUG [main] 2007-05-02 06:32:06,056 (PKCert.java:583) - Binary File:
/var/certsInput/Local_System.der Read. Length: 1218
DEBUG [main] 2007-05-02 06:32:06,062 (PKCert.java:553) - Characters Read: 943
DEBUG [main] 2007-05-02 06:32:06,063 (PKCert.java:583) - Binary File:
/var/certsInput/Local_System.cer Read. Length: 943
DEBUG [main] 2007-05-02 06:32:06,064 (PKCert.java:455) - Jar File Path:
/opt/CSCObpr/lib/pkcerts.jar
DEBUG [main] 2007-05-02 06:32:06,065 (PKCert.java:456) - Opened jar file:
/opt/CSCObpr/lib/pkcerts.jar
DEBUG [main] 2007-05-02 06:32:06,067 (PKCert.java:460) - Jar entry unfiltered:
Tag_Packetcable_Tag/
DEBUG [main] 2007-05-02 06:32:06,068 (PKCert.java:460) - Jar entry unfiltered:
Tag_Packetcable_Tag/CableLabs_Service_Provider_Root.cer
...
DEBUG [main] 2007-05-02 06:32:06,115 (PKCert.java:472) - File:
Tag_Packetcable_Tag/Manu.cer
DEBUG [main] 2007-05-02 06:32:06,116 (PKCert.java:472) - File:
Tag_Packetcable_Tag/Service_Provider.cer

```



```

DEBUG [main] 2007-05-02 06:32:06,121 (PKCCreate.java:91) - Found 7 files in jar.
DEBUG [main] 2007-05-02 06:32:06,827 (KDCCert.java:98) - SP Cert subject name:
C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs Local System CA
SP Cert subject name: C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs
Local System CA
DEBUG [main] 2007-05-02 06:32:07,687 (KDCCert.java:293) - Setting issuer to:
C=US,O=CableLabs\, Inc.,OU=ABC Cable Company,CN=Shared-01 CableLabs Local System CA
DEBUG [main] 2007-05-02 06:32:07,699 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@bd0b4ea6

DEBUG [main] 2007-05-02 06:32:07,700 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@5035bc0

DEBUG [main] 2007-05-02 06:32:07,701 (KDCCert.java:231) - DERVisibleToGeneral
org.bouncycastle.asn1.DERGeneralString@5035bc0

DEBUG [main] 2007-05-02 06:32:07,703 (KDCCert.java:210) - DERCombineTagged [0] IMPLICIT
DER ConstructedSequence
 ObjectIdentifier(1.3.6.1.5.2.2)
 Tagged [0]
 DER ConstructedSequence
 Tagged [0]
 org.bouncycastle.asn1.DERGeneralString@5035bc0
 Tagged [1]
 DER ConstructedSequence
 Tagged [0]
 Integer(2)
 Tagged [1]
 DER ConstructedSequence
 org.bouncycastle.asn1.DERGeneralString@bd0b4ea6
 org.bouncycastle.asn1.DERGeneralString@5035bc0

File written: /var/certsOutput/KDC_private_key.pkcs8
File written: /var/certsOutput/KDC_private_key_proprietary.
File written: /var/certsOutput/KDC_PublicKey.der
File written: /var/certsOutput/KDC.cer
KDC Certificate Successfully Created at /var/certsOutput/KDC.cer

Copy KDC.cer to the KDC certificate directory (i.e.
/opt/CSCObpr/kdc/solaris/packetcable/certificates)
Copy KDC_private_key.pkcs8 to the KDC platform directory (i.e. /opt/CSCObpr/kdc/solaris)
Copy KDC_private_key_proprietary. to the KDC platform directory (i.e.
/opt/CSCObpr/kdc/solaris)

```

## Using the KeyGen Tool

The KeyGen tool is used to generate PacketCable service keys. The service keys are symmetric triple data encryption standard (triple DES or 3DES) keys (shared secret) required for KDC communication. The KDC server requires service keys for each of the provisioning FQDNs of the DPE. Any changes made to the DPE provisioning FQDN from the DPE CLI requires a corresponding change to the KDC service key filename. This change is necessary since the KDC service key uses the DPE provisioning FQDN as part of its filename.

The KeyGen tool, which resides in the *BAC\_home/kdc* directory, uses command-line arguments for the DPE provisioning FQDN, realm name, and a password, and generates the service key files.

**Note**

When running this tool, remember to enter the same password that you used to generate the service key on the DPE (by using the **packetCable registration kdc-service-key** command from the DPE CLI). For information on setting this password, refer to the *Cisco Broadband Access Center DPE CLI Reference, 2.7.1*.

The KDC server reads the service keys on startup. Any modification to the service keys requires that you restart the KDC server.

**Syntax Description**

**keygen options fqdn realm password**

- *options* are:
  - **-?**—Displays this usage message and exits the command.
  - **-v** or **-version**—Displays the version of this tool and exits the command.
  - **-q** or **-quiet**—Implements a quiet mode whereby no output is created.
  - **-c** or **-cms**—Creates a service key for the CMS system.
- *fqdn*—Identifies the FQDN of the DPE and is a required entry.
- *realm*—Identifies the Kerberos realm and is a required entry.
- *password*—Specifies the password to be used. This is also a required field. The password must be from 6 to 20 characters.

Three service key files are written in the KDC keys directory using this filename syntax:

*mtafqdnmap,fqdn@REALM*

*mtaprovsrvr,fqdn@REALM*

*krbtgt,REALM@REALM*

- *fqdn*—Identifies the FQDN of the DPE.
- *REALM*—Identifies the Kerberos realm.

The service key file always contains a version field of 0x0000.

**Examples**

```
keygen dpe.cisco.com CISCO.COM changeme
```

When this command is implemented, these KDC service keys are written to the *BAC\_home/kdc/solaris/keys* directory:

```
mtafqdnmap,dpe.cisco.com@CISCO.COM
mtaprovsrvr,dpe.cisco.com@CISCO.COM
krbtgt,CISCO.COM@CISCO.COM
```

Restart the KDC, so that the new keys are recognized. Use this BAC process watchdog command to restart the KDC:

```
/etc/init.d/bprAgent restart kdc
```

This example illustrates the generation of a CMS service key:

```
keygen -c cms-fqdn.com CMS-REALM-NAME changeme
```

When this command is implemented, this CMS service key is written to the *BAC\_home/kdc/solaris/keys* directory.

```
cms , cms-fqdn.com@CMS-REALM-NAME
```

### Verifying the KDC Service Keys

Once you generate the service keys on the KDC and the DPE, verify if the service keys match on both components.

The KeyGen tool requires you to enter the same password that you used to generate the service key on the DPE using the **packetCable registration kdc-service-key** command. Once you set this password on the DPE, you can view the service key from the *dpe.properties* file, which resides in the *BAC\_home/dpe/conf* directory. Look for the value against the */pktcbl/regsvr/KDCServiceKey=* property.

For example:

```
more dpe.properties
/pktcbl/regsvr/KDCServiceKey=2e:d5:ef:e9:5a:4e:d7:06:67:dc:65:ac:bb:89:e3:2c:bb:
71:5f:22:bf:94:cf:2c
```



**Note** The output of this example has been trimmed for demonstration purposes.

To view the service key generated on the KDC, run the following command from the *BAC\_home/kdc/solaris/keys* directory:

```
od -Ax -tx1 mtaprovsrvr,fqdn@REALM
```

- *fqdn*—Identifies the FQDN of the DPE.
- *REALM*—Identifies the Kerberos realm.

The output that this command generates should match the value of the */pktcbl/regsvr/KDCServiceKey=* property in the *dpe.properties* file.

For example:

```
od -Ax -tx1 mtaprovsrvr,dpe.cisco.com@CISCO.COM
0000000 00 00 2e d5 ef e9 5a 4e d7 06 67 dc 65 ac bb 89
0000010 e3 2c bb 71 5f 22 bf 94 cf 2c
000001a
```

In the examples shown here, note that the service key generated at the KDC matches the service key on the DPE.

## Using the changeNRProperties.sh Tool

The BAC installation program establishes values for configuration properties used by BAC extensions that are incorporated into the Network Registrar DHCP server. You use the **changeNRProperties.sh** command, which is found in the *BAC\_home/cnr\_ep/bin* directory, to change key configuration properties.

Invoking the script without any parameters displays a help message listing the properties that can be set.

To run this command:

**Step 1** Change directory to `BAC_home/cnr_ep/bin`.

**Step 2** Run the `changeNRProperties.sh` command using this syntax:

`changeNRProperties.sh options`

Where *options* are:

- **-help**—Displays this help message. The `-help` option must be used exclusively. Do not use this in conjunction with any other option.
- **-e enabled | disabled**—Sets the PacketCable enable property. Enter **-e enabled** to enable the property, and **-e disabled** to disable it.
- **-d**—Displays the current properties. The `-d` option must be used exclusively. Do not use this in conjunction with any other option.
- **-s secret**—Identifies the BAC shared secret. For example, enter **-s secret**, if the shared secret is the word `secret`.
- **-f fqdn**—Identifies the RDU FQDN. For example, enter **-f rdu.cisco.com**, if you use `rdu.cisco.com` as the fully qualified domain name.
- **-p port**—Identifies the RDU port you want to use. For example, enter **-p 49187**, if you wanted to use port number 49187.
- **-r realm**—Identifies the PacketCable realm. For example, enter **-r CISCO.COM**, if your PacketCable realm is `CISCO.COM`.



**Note** The realm must be entered in uppercase letters.

- **-g prov\_group**—Identifies the provisioning group. For example, enter **-g group1**, if you are using provisioning group called `group1`.
- **-t 00 | 01**—Identifies whether or not the PacketCable TGT is set to off or on. For example, enter **-t 00** to set this to off, or **-t 01** to set it to on.
- **-a ip**—Identifies the PacketCable primary DHCP server address. For example, enter **-a 10.10.10.2** if the IP address of your primary DHCP server is `10.10.10.2`.
- **-b ip**—Identifies the PacketCable secondary DHCP server address. For example, enter **-b 10.10.10.4**, if the IP address of your secondary DHCP server is `10.10.10.4`. You can also enter **-b null** to set a null value, if appropriate.
- **-y ip**—Identifies the PacketCable primary DNS server address. For example, enter **-y 10.10.10.6**, if the IP address of the PacketCable primary DNS server is `10.10.10.6`.
- **-z ip**—Identifies the PacketCable secondary DNS server address. For example, enter **-z 10.10.10.8**, if the IP address of your secondary DNS server is `10.10.10.8`. You can also enter **-z null** to set a null value, if appropriate.
- **-o prov\_ip man\_ip**—Sets the management address to use for communication with the DPE identified by the given provisioning address. For example, **-o 10.10.10.7 10.14.0.4**, if the IP address of your provisioning group is `10.10.10.7`. You can also enter a null value, if appropriate; for example, **-o 10.10.10.7 null**.

**Step 3** Restart the DHCP server.

## Examples

This is an example of changing the Network Registrar extensions by using the NR Extensions Properties tool:

```
/opt/CSCOopr/cnr_ep_bin/changeNRProperties.sh -g primary1
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.acme.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
PacketCable Enable: enabled
CableLabs client TGT: 01
CableLabs client Realm: ACME.COM
CableLabs client Primary DHCP Server: 10.10.1.2
CableLabs client Secondary DHCP Server: NOT SET
CableLabs client Primary DNS Server: 10.10.1.2
CableLabs client Secondary DNS Server: NOT SET
```



### Note

You must restart your NR DHCP server for the changes to take effect.

This is an example of viewing the current properties:

```
/opt/CSCOopr/cnr_ep_bin/changeNRProperties.sh -d
Current NR Properties:
RDU Port: 49187
RDU FQDN: rdu.acme.com
Provisioning Group: primary1
Shared Secret: fggTaLg0XwKRs
PacketCable Enable: enabled
CableLabs client TGT: 01
CableLabs client Realm: ACME.COM
CableLabs client Primary DHCP Server: 10.10.1.2
CableLabs client Secondary DHCP Server: NOT SET
CableLabs client Primary DNS Server: 10.10.1.2
CableLabs client Secondary DNS Server: NOT SET
```

## Using the snmpAgentCfgUtil.sh Tool

You can use the **snmpAgentCfgUtil.sh** tool to manage the SNMP agent installed on a Solaris computer. Using this tool, which resides in the *BAC\_home/snmp/bin* directory, you can add (or remove) your host to a list of other hosts that receive SNMP notifications, and start and stop the SNMP agent process. This tool should be run from the local directory.



### Note

The default port number of an SNMP agent running on a Solaris computer is 8001.

You can use the RDU SNMP agent to:

- [Adding a Host, page 13-16](#)
- [Deleting a Host, page 13-16](#)
- [Adding an SNMP Agent Community, page 13-17](#)
- [Deleting an SNMP Agent Community, page 13-17](#)
- [Starting the SNMP Agent, page 13-18](#)
- [Stopping the SNMP Agent, page 13-18](#)
- [Changing the SNMP Agent Location, page 13-19](#)

- [Setting Up SNMP Contacts, page 13-19](#)
- [Displaying SNMP Agent Settings, page 13-20](#)

## Adding a Host

You use this command to add the host address to the list of hosts that receive SNMP notifications from the SNMP agent.

### Syntax Description

**snmpAgentCfgUtil.sh add host** *ip-addr* **community** *community* [**udp-port** *port*]

- *ip-addr*—Specifies the IP address of the host to which notifications are sent.
- *community*—Specifies the community (read or write) to be used while sending SNMP notifications.
- *port*—Identifies the UDP port used for sending the SNMP notifications.

### Examples

```
./snmpAgentCfgUtil.sh add host 10.10.10.5 community trapCommunity udp-port 162
OK
Please restart [stop and start] SNMP agent.
```



#### Note

The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

## Deleting a Host

You use this command to remove a host from the list of those receiving SNMP notifications from the SNMP agent.

### Syntax Description

**snmpAgentCfgUtil.sh delete host** *ip-addr*

*ip-addr*—Specifies the IP address of the host that you want to delete from the list of hosts.

### Examples

```
./snmpAgentCfgUtil.sh delete host 10.10.10.5
OK
Please restart [stop and start] SNMP agent.
```



#### Note

The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

## Adding an SNMP Agent Community

You use this command to add an SNMP community string to allow access to the SNMP agent.

### Syntax Description

**snmpAgentCfgUtil.sh add community *string* [ro | rw]**

- *string*—Identifies the SNMP community.
- **ro**—Assigns a read-only (**ro**) community string. Only **get** requests (queries) can be performed. The **ro** community string allows **get** requests, but no **set** operations. The NMS and the managed device must reference the same community string.
- **rw**—Assigns a read-write (**rw**) community string. SNMP applications require read-write access for **set** operations. The **rw** community string enables write access to OID values.



**Note** The default **ro** and **rw** community strings are `baccread` and `baccwrite`, respectively. We recommend that you change these values before deploying BAC.

### Examples

```
./snmpAgentCfgUtil.sh add community fsda54 ro
OK
Please restart [stop and start] SNMP agent.
```



**Note** The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

## Deleting an SNMP Agent Community

You use this command to delete an SNMP community string to prevent access to the SNMP agent.

### Syntax Description

**snmpAgentCfgUtil.sh delete community *string* [ro | rw]**

- *string*—identifies the SNMP community
- **ro**—assigns a read only (ro) community string
- **rw**—assigns a read write (rw) community string



### Note

See [Adding an SNMP Agent Community, page 13-17](#), for additional information on the **ro** and **rw** community strings.

### Examples

```
./snmpAgentCfgUtil.sh delete community fsda54 ro
OK
Please restart [stop and start] SNMP agent.
```

**Note**

The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

## Starting the SNMP Agent

You use this command to start the SNMP agent process on a Solaris computer on which BAC is installed.

**Note**

You can also start the SNMP agent by invoking the BAC process watchdog using the `/etc/init.d/bprAgent start snmpAgent` command. For more information, see [Using the BAC Process Watchdog from the Command Line, page 2-14](#).

### Examples

```
./snmpAgentCfgUtil.sh start
Process snmpAgent has been started
```

## Stopping the SNMP Agent

You use this command to stop the SNMP agent process on a Solaris computer on which BAC is installed.

**Note**

You can also start the SNMP agent by invoking the BAC process watchdog using the `/etc/init.d/bprAgent stop snmpAgent` command. For more information, see [Using the BAC Process Watchdog from the Command Line, page 2-14](#).

### Examples

```
./snmpAgentCfgUtil.sh stop
Process snmpAgent has stopped
```

## Configuring an SNMP Agent Listening Port

You use this command to specify the port number that the SNMP agent will listen to. The default port number used by RDU SNMP agent is 8001.

### Syntax Description

```
snmpAgentCfgUtil.sh udp-port port
```

*port* identifies the port number that the SNMP agent will listen to.

### Examples

```
./snmpAgentCfgUtil.sh udp-port 8001
OK
Please restart [stop and start] SNMP agent.
```



**Note**

The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

## Changing the SNMP Agent Location

You use this command to enter a string of text that indicates the location of the device running the SNMP agent. This could, for example, be used to identify the physical location of the device. You can enter any character string that is fewer than 255 characters.

### Syntax Description

`snmpAgentCfgUtil.sh location location`

*location* is the character string identifying the agent's location.

### Examples

In this example, the physical location of the SNMP agent is in an equipment rack identified as rack 5D:

```
./snmpAgentCfgUtil.sh location "equipmentrack5D"
OK
Please restart [stop and start] SNMP agent.
```

**Note**

The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

## Setting Up SNMP Contacts

You can use this command to enter a string of text that identifies the contact person for the SNMP agent, together with information on how to contact this person. This could, for example, be used to identify a specific person including that person's telephone number. You can enter any character string that is fewer than 255 characters.

### Syntax Description

`snmpAgentCfgUtil.sh contact contact-info`

*contact-info* is the character string identifying the individual to contact concerning the SNMP agent.

### Examples

In this example, the contact name is Terry and the telephone extension is 1234:

```
./snmpAgentCfgUtil.sh contact "Terry-ext1234"
OK
Please restart [stop and start] SNMP agent.
```

**Note**

The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

## Displaying SNMP Agent Settings

You use this command to display all current SNMP settings.

### Examples

```
./snmpAgentCfgUtil.sh show
Location : equipmentrack5D
Contact : Terry-ext1234
Port Number : 8001
Notification Type : trap
Notification Recipient Table :
 [Host IP address, Community, UDP Port]
 [10.10.10.5 , trapCommunity , 162]
Access Control Table :
 Read Only Communities
 baccread
 Read Write Communities
 baccwrite
```

## Specifying SNMP Notification Types

You use this command to specify the types of notifications (traps or informs) that will be sent from the SNMP agent. By default, traps are sent, though you can set this to send SNMP informs instead.

### Syntax Description

**snmpAgentCfgUtil.sh inform [retries timeout] | trap**

Where the parameter is the backoff timeout between retries.

### Examples

```
./snmpAgentCfgUtil.sh inform retries 3 timeout 1000
OK
Please restart [stop and start] SNMP agent.
```



**Note** The changes that you introduce through this command do not take effect until you restart the SNMP agent by using the `/etc/init.d/bprAgent restart snmpAgent` command. For detailed information, see [BAC Process Watchdog, page 2-14](#).

Use the `snmpAgentCfgUtil.sh show` command to verify your configuration settings.

```
./snmpAgentCfgUtil.sh show
Location : equipmentrack5D
Contact : Terry-ext1234
Port Number : 8001
Notification Type : inform
Notification Retries : 3
Notification Timeout : 1000
Notification Recipient Table :
 [Host IP address, Community, UDP Port]
 [10.10.10.5 , trapCommunity , 162]
Access Control Table :
 Read Only Communities
 baccread
 Read Write Communities
```

baccwrite

## Using the `disk_monitor.sh` Tool

Monitoring available disk space is an important system administration task. You can use a number of custom written scripts or commercially available tools to do so.

The `disk_monitor.sh` command, which resides in the `BAC_home/rdu/samples/tools` directory, sets threshold values for one or more file systems. When these thresholds are surpassed, an alert is generated through the Solaris syslog facility, at 60-second intervals, until additional disk space is available.



### Note

We recommend that, at a minimum, you use the `disk_monitor.sh` script to monitor the `BAC_data` and `BAC_dblog` directories.

### Syntax Description

`disk_monitor.sh filesystem-directory x [filesystem-directory* x*]`

- `filesystem-directory`—Identifies any directory in a file system to monitor.
- `x`—Identifies the percentage threshold applied to the specified file system.
- `filesystem-directory*`—Identifies multiple file systems.
- `x*`—Specifies percentage thresholds to be applied to multiple file systems.

### Examples

#### Example 1

This example specifies that a notification be sent out when the `/var/CSCObpr` file system reaches 80 percent of its capacity.

```
./disk_monitor.sh /var/CSCObpr 80
```

When the database logs disk space reaches 80-percent capacity, an alert similar to the following one is sent to the syslog file:

```
Dec 7 8:16:06 perf-u80-1 BPR: [ID 702911 local6.warning] File system /var/bpr usage is 81%
(threshold is 80%)
```

#### Example 2

This example describes how you can run the `disk_monitor.sh` tool as a background process. Specifying an ampersand (&) at the end of the command immediately returns output while running the process in the background.

```
./disk_monitor.sh /var/CSCObpr 80 &
1020
```

## Troubleshooting Devices by MAC Address

You can use this feature to collect detailed diagnostics about one or more specific devices.

Troubleshooting information includes all server interactions related to a given device or a group of devices. This information includes administrator user interface operations, RDU API operations, DPE interactions with devices, and interserver DPE-to-RDU interactions.

You can enable or disable troubleshooting via node management for one or more specific devices without turning logging on, and without searching through log files for specific device information.

BAC maintains a list of devices, based on MAC addresses, for which detailed diagnostics are collected. Troubleshooting information is stored centrally at the RDU and is maintained on a per-device basis. Neither DPEs nor Network Registrar extensions store this data. Rather, they forward this information to the RDU, which, upon receiving information, writes it to the appropriate device log file. If the connection from the DPE or Network Registrar extension to the RDU is lost, any new troubleshooting events occurring on the DPE or Network Registrar extension are discarded. The logging of troubleshooting information resumes only after the connection to the RDU is restored.

The DPE maps all device MAC addresses to its IP address mapping for that device, and the Network Registrar extensions send the IP update to the DPE whenever the extensions determine that device troubleshooting is enabled.

Any modifications to the device tracking list, such as the addition of a new device or a group, take place immediately at all servers; you do not have to reboot the RDU or the DPE. The log files on the respective servers list the current list of devices in the troubleshooting mode.

**Caution**

Additional memory and disk space is required whenever the device troubleshooting feature is used. As the number of tracked devices increases, so does the amount of memory and disk space that is required to support the number of logs that are created.

The device troubleshooting feature is disabled until one or more devices are set in troubleshooting mode.

**Note**

To enable diagnostics for a device, the device must be preregistered in the BAC RDU. If the device is not yet preregistered, add the device from the Manage Devices page by clicking the Add button. For information on adding devices, see [Adding Device Records, page 10-14](#).

You can configure a maximum number of devices in diagnostics mode to prevent inadvertently putting too many devices in this mode and thus diminishing server performance. By default, this number is set at 100. To configure the maximum number of devices allowed in troubleshooting mode from the administrator user interface, click the Systems Defaults page via the **Configuration > Defaults** tabs. Enter a value in the Maximum Diagnostics Device Count field.

## Relating a Device to a Node

You can troubleshoot a device by relating it to a specific node. Use the Relate function to associate a device, using its MAC address, to a specific node, which is in turn associated with a specific node type. (See [Relating and Unrelating Devices, page 10-16](#).) Doing so records an extraordinarily large volume of information for a device; you can then use the information to troubleshoot potential problems.

[Table 13-3](#) identifies a possible workflow using the Relate and Unrelate functions.

**Table 13-3** Sample Relate/Unrelate Process

| Step | Action                                                                                             |
|------|----------------------------------------------------------------------------------------------------|
| 1.   | Determine whether or not a problem exists and identify which devices are affected.                 |
| 2.   | Relate the devices to a node.                                                                      |
| 3.   | Wait a few minutes to ensure that device traffic is passing, or perform a hard boot of the device. |

**Table 13-3** Sample Relate/Unrelate Process (continued)

| Step | Action                                                                                                                    |
|------|---------------------------------------------------------------------------------------------------------------------------|
| 4.   | Open the rdu.log file in a word processing application and locate the entries for the MAC address of the specific device. |
| 5.   | Identify, correct, test, and verify the problem.                                                                          |
| 6.   | Unrelate the device from the node.                                                                                        |

## Viewing a List of the Devices in Troubleshooting Mode

When you enable troubleshooting for a device, the device is automatically added to a special device node that contains a list of devices in troubleshooting mode. The node type is **system** and the node name is **diagnostics**. You can access the list of devices in this group from the API or the administrator user interface.

To view a list of devices currently enabled for troubleshooting:

- 
- Step 1** From the Manage Devices page, click the Search Type drop-down list and select Node Search.
- Step 2** From the Node Name (Node Type) drop-down list, select the diagnostics (system) option to view all the devices in troubleshooting mode.
- Step 3** Click **Search**.



**Note** An alternative way to view the list of devices in troubleshooting mode is to consult the RDU log (rdu.log) and the DPE log (dpe.log) files. The list of devices is logged whenever the server is started and whenever there is a change in the list of devices enabled for diagnostics.

The devices enabled for troubleshooting appear in the log files with the log level of 5-notification. For details on log files, see [Logging, page 2-15](#).

---

### Examples

This example features log output while troubleshooting an MTA:

```

bac-test.cisco.com:2005 03 04 18:38:24 EST:%BPR-DIAGNOSTICS-3-4055:[##MTA-9a Unconfirmed
FQDN Request Received from [/10.10.10.5 ['kdcquery']]. Client with IP Address [10.10.20.2]
and MAC Address [1,6,00:00 :ca:b7:7e:91]]]
bac-test.cisco.com:2005 03 04 18:38:24 EST:%BPR-DIAGNOSTICS-3-4082:[Results of BACC
Lookup. FQDN: [1-6-00-00-ca-b7-7e-91.cisco.com MAC: 1,6,00:00:ca:b7:7e:91. Client with IP
Address [10.10.20.2] and MAC Address [1,6,00:00:ca:b7:7e:91]]]
bac-test.cisco.com:2005 03 04 18:38:24 EST:%BPR-DIAGNOSTICS-3-4070:[##MTA-9b FQDN Reply
Sent to [/10.10.20.2(41142) for MTA 1,6,00:00:ca:b7:7e:91. Client with IP Address
[10.10.20.2] and MAC Address [1,6, 00:00:ca:b7:7e:91]]]
bac-test.cisco.com:2005 03 04 18:38:26 EST:%BPR-DIAGNOSTICS-3-4132:[##MTA-13 Incoming
APREQ received from [/10.10.20.2:1293. Client with IP Address [10.10.20.2] and MAC
Address [1,6,00:00:ca:b7:7e:91]]]
bac-test.cisco.com:2005 03 04 18:38:26 EST:%BPR-DIAGNOSTICS-3-4141:[##MTA-13 APREP sent to
[/10.10.20.2(1293) For MTA 1,6,00:00:ca:b7:7e:91. Client with IP Address [10.10.20.2] and
MAC Address [1,6,00:00: ca:b7:7e:91]]]
bac-test.cisco.com:2005 03 04 18:38:26 EST:%BPR-DIAGNOSTICS-3-0764:[##MTA-15 SNMPv3
INFORM Received From 10.10.20.2. Client with IP Address [10.10.20.2] and MAC Address
[1,6,00:00:ca:b7:7e:91]]]

```

```
bac-test.cisco.com:2005 03 04 18:38:26 EST:%BPR-DIAGNOSTICS-3-0764:[[#MTA-19 SNMPv3 SET
Sent to 10.10.20.2. Client with IP Address [10.10.20.2] and MAC Address
[1,6,00:00:ca:b7:7e:91]]]
bac-test.cisco.com:2005 03 04 18:38:26 EST:%BPR-DIAGNOSTICS-3-1092:[Received a TFTP [read]
request from [10.10.20.2:1271] for [bpr01060000cab77e910002]; Client with MAC Address
[1,6,00:00:ca:b7:7e:91] and IP Address [10.10.20.2]]
bac-test.cisco.com:2005 03 04 18:38:26 EST:%BPR-DIAGNOSTICS-3-1155:[[#MTA-23 Finished
handling [read] request from [10.10.20.2:1271] for [bpr01060000cab77e910002]; Transferred
[236] bytes to Client with MAC Address [1,6,00:00:ca:b7:7e:91] and IP Address
[10.10.20.2]]
bac-test.cisco.com:2005 03 04 18:38:27 EST:%BPR-DIAGNOSTICS-3-0764:[[#MTA-25 SNMP
Provisioning State INFORM Received from 10.10.20.2. Client with IP Address [10.10.20.2]
and MAC Address [1,6,00:00:ca:b7:7e:91]]]
bac-test.cisco.com:2005 03 04 18:38:27 EST:%BPR-DIAGNOSTICS-3-0764:[MTA Configuration
Confirmed, Returned 'pass' as the final MTA provisioning state for 10.10.20.2. Client with
IP Address [10.10.20.2] and MAC Address [1,6,00:00:ca:b7:7e:91]]]
```



## CHAPTER 14

# Database Management

---

This chapter contains information on RDU database management and maintenance. The RDU database is the Broadband Access Center (BAC) central database. The BAC RDU requires virtually no maintenance other than to ensure availability of sufficient disk space. As the administrator, you must understand and be familiar with database backup and recovery procedures.

## Understanding Failure Resiliency

The RDU database uses a technique known as *write-ahead logging* to protect against database corruption that could result from unforeseen problems, such as an application crash, system failure, or power outage.

Write-ahead logging involves writing a description of any database change to a database log file prior to writing the change into the database files. This mechanism allows the repair of any incomplete database writes that can result from system failures.

The RDU server performs an automatic recovery each time it is started. During this recovery process, the database log files are used to synchronize the data with the database files. Database changes that were written into the database log, but not into the database, are written into the database during this automatic recovery.

In this way, write-ahead logging virtually guarantees that the database does not become corrupted when the RDU server crashes because the database is automatically repaired when the RDU server is restarted.

Write-ahead logging requires these conditions to work properly:

- You must set up the file system and physical storage so that they guarantee that the data is flushed to physical storage when requested. For example, a storage system with SDRAM-only write cache, which loses data during system failure, is not appropriate. However, a disk array with battery-backup write cache that guarantees that the data gets persisted, even in the event of a system failure, is acceptable. A system without battery-backed write cache should flush the data disk when requested instead of performing in-memory data caching.
- You must set up the file system with an 8192-byte block size to match the RDU database block size. This setting is usually the default on Solaris unless explicitly adjusted.

## Database Files

The RDU database stores data in binary files using the file system you have mounted on the partition containing the files. It is essential to choose and configure a file system in a way that it is not susceptible to long recovery times after system failures.

Database files are vital to the operation of the RDU. Therefore, take extra precautions to safeguard them against accidental removal or other manual manipulation. Follow standard system administration practices to safeguard these important files. For example, these files should always have permissions that allow only root user access. Additionally, it is a good practice to never log in to your production system as a root user. Instead, log in as a less privileged user and use the **sudo** command to execute tasks requiring root privileges.

## Database Storage File

The RDU server stores its database in a file called *bpr.db*, which resides in the database directory. This directory resides in the *BAC\_data/rdu/db* directory; you can configure this location by specifying the *BAC\_data* parameter during a component installation. See [Changing Database Location, page 14-7](#), for additional information on moving the database.

**Note**

---

The database file is normally accessed in a random fashion. You should, therefore, select a disk with the fastest seek time and rotational access latency to obtain the best database performance.

---

## Database Transaction Log Files

The RDU server stores database transaction logs in 10-MB files that are stored in the database log directory. You configure this directory during installation by specifying the *BAC\_dblog* parameter. The log directory resides in the *BAC\_dblog/rdu/dblog* directory. See [Changing Database Location, page 14-7](#), for additional information on moving the transaction logs to a new directory.

Database log files are named in numeric sequence, starting at log.00000001, log.00000002, and so on.

**Note**

---

The disk on which transaction logs are stored is usually accessed in a sequential manner, with data being appended to the log files. Select a disk that can efficiently handle this access pattern to achieve the best database performance. We recommend that you locate the database transaction logs directory on the fastest disk on the system. Also, ensure that 1 GB of disk space is available.

---

## Automatic Log Management

Database transaction logs files are used to store transaction data until that data is completely written into the database. After that, the transaction log data becomes redundant and the files are then automatically removed from the system.

Under normal circumstances there should be only a few log files in the database transaction log directory. Over time, you will notice that older transaction logs disappear and newer ones are created.



**Note**

Database transaction logs are an integral part of the database. Manual deletion of transaction log files will result in database corruption.

## Miscellaneous Database Files

The database directory contains additional files that are essential to database operation. These files, in addition to the `rdu.db` file, are found in the `BAC_data/rdu/db` directory and are copied as part of the database backup:

- `DB_VERSION`—Identifies the physical and logical version of the database and is used internally by the RDU.
- `history.log`—Used to log activity about essential database management tasks, such as automatic log file deletion, backup, recovery, and restore operations. In addition to providing useful historical information for the administrator, this log file is essential to RDU database operation.

## Disk Space Requirements

The size of a fully populated database depends on a number of factors:

- Device objects that the RDU manages
- Custom properties stored on each object

The approximate estimates for disk space required for each partition are:

- `BAC_data`, approximately 3 to 5 KB per device object
- `BAC_dblog`, at least 500 MB

**Caution**

These numbers are provided as a guideline only and do not eliminate the need for normal system monitoring.

You can use the `disk_monitor.sh` tool to monitor available disk space and alert the administrator. See [Using the `disk\_monitor.sh` Tool, page 13-21](#), for additional information.

## Handling Out of Disk Space Conditions

When the RDU server runs out of disk space, it generates an alert through the syslog facility and the RDU log. The RDU server then tries to restart automatically. When attempting to restart, the RDU server may again encounter the out of disk space error and attempt to restart again.

The RDU server continues trying to restart until free disk space becomes available. Once you free up some disk space on the disk that is operating near a limit, the next time the RDU restarts it will succeed.

If the size of your database grows beyond the capacity of the current disk partition, move the database to a new disk or partition. For more information, see [Changing Database Location, page 14-7](#).

**Note**

It is a good practice to monitor disk space utilization to prevent failure. See [Using the `disk\_monitor.sh` Tool, page 13-21](#), for additional information.

# Backup and Recovery

The RDU server supports a highly efficient backup process that can be performed without stopping the server or suspending any of its activities. Database backup and recovery involves these stages:

- Backup—Takes a snapshot of the RDU database from a live server.
- Recovery—Prepares the database snapshot for re-use.
- Restore—Copies the recovered database snapshot to the RDU server.

**Note**

---

Once migration is complete, you can optionally check for database consistency.

---

Automated tools are provided for each of these steps. You can use these tools in either interactive mode or silent mode, but you must have root privileges to use the tools.

## Database Backup

Backup is the process of copying the database files into a backup directory. The files can then be compressed and placed on tape or other archive.

RDU database backup is highly efficient because it involves just copying files without interrupting server activity. However, because it involves accessing the RDU database disk, backup may adversely affect RDU performance. The opposite is also true. RDU activity happening during backup will adversely affect backup performance. Therefore, you should perform backups during off-peak hours.

Other than concurrent system activity, backup performance also depends on the underlying disk and file system performance. Essentially, backup will perform as fast as database files can be copied from source to target.

Use the **backupDb.sh** tool, in the *BAC\_home/rdu/bin* directory, to perform database backups:

- To use this tool, you must provide the target directory where the backup files will be placed. This directory should be on a disk or partition that has available disk space equivalent to 120% of the current database file size.
- As illustrated in the following example, this tool automatically creates a timestamped subdirectory, under the directory you specify, and places the backups there.

The **backupDb.sh** command also reports progress to the screen and logs its activity in *history.log*.

When using the **backupDb.sh** tool, you can use a **-help** option to obtain usage information. You can also use the optional **-nosubdir** flag to disable, if necessary, the automatic creation of the subdirectory.

---

**Examples**

In this example, */var/backup* identifies the target location for database backup files.

```
backupDb.sh /var/backup

Database backup started
Back up to: /var/backup/rdu-backup-20070316-031028

Copying DB_VERSION. Size: 396 bytes.
DB_VERSION: 100% completed.

Copying bpr.db. Size: 434176 bytes.
bpr.db: 100% completed.
```

```
Copying log.0000000001. Size: 469268 bytes.
log.0000000001: 100% completed.
```

```
Copying history.log. Size: 574 bytes.
history.log: 100% completed.
```

```
Database backup completed
```

In this example, all backup database files are stored in a directory called `/var/backup/rdu-backup-20070316-031028`. The last subdirectory (`rdu-backup-20070316-031028`) is automatically created with a current timestamp.

**Note**

The timestamped subdirectory format is `rdu-backup-yyyyMMdd-HHmss`. In this example, the subdirectory would be `rdu-backup-20070316-031028`, meaning that the directory contains a backup that was started at 3:10:28 a.m. on March 16, 2007.

## Database Recovery

Database recovery is the process of restoring the database to a consistent state. Since backup is performed on a live RDU, the database can be changing while it is being copied. The database log files, however, ensure that the database can be recovered to a consistent state.

Recovery is performed on a snapshot of a database. In other words, this task does not involve touching the database on the live RDU server. The recovery task can be performed either immediately following a backup or prior to restoring the database to the RDU server.

**Note**

We recommend that you perform database recovery immediately after each backup. This way, the backed-up database can be more quickly restored in case of emergency.

The duration of database recovery depends on the number of database log files that were copied as part of the backup, which in turn depends on the level of RDU activity at the time of the backup. The more concurrent activity RDU experiences during the backup, the more transaction log files have to be copied as part of the backup and the longer is the recovery. Generally, recovering a database takes from 10 to 60 seconds per transaction log file.

Use the **recoverDb.sh** tool, located in the `BAC_home/rdu/bin` directory, to perform recovery of the snapshot of a database. When you use this tool, you must provide the location of the backup. This is also the directory in which the recovery will be performed.

When using the **recoverDb.sh** tool, you can use the **-help** option to obtain usage information on the tool.

**Note**

Once migration is complete, you can run the **verifyDb.sh** tool to check the integrity of the database. Verification is an optional task and you can skip it if shorter downtime for migration is critical. Run the **verifyDb.sh** script, which resides in the `BPR_HOME/rdu/internal/db/bin` directory, after migrating the database. You can run the tool against the RDU when the RDU server is down or against a backup snapshot of the database. Use the **-help** option for usage information on the command.

Before running this tool, ensure that you stop the RDU server. Also, remember to verify database consistency on a separate system because the process of verification consumes additional memory and disk space.

**Examples**

```
recoverDb.sh /var/backup/rdu-backup-20070316-031028

*
* Recovery process modifies the backup snapshot of
* the database. You should never do recovery without
* making a copy of the database and log files you
* are using for recovery.
*

To start recovery please type "yes" and enter: yes

Database recovery started
Recovering in: /var/backup/rdu-backup-20070316-031028
This process may take a few minutes.
Database recovery completed
```

In this example, the snapshot located in the `/var/backup/rdu-backup-20070316-031028` directory is recovered to a consistent state. The progress of the recovery operation appears on screen and the activity is recorded in the `history.log` file in the snapshot directory.

## Database Restore

Restoring the database is the process of copying the previously recovered database snapshot to the database location used by the RDU server. Only a database that has been previously recovered can be restored.

Since restoring the database means replacing the current RDU database, it is very important that you first properly remove and archive the old database.

**Caution**

Do not delete the database you are replacing. You might need a copy of an old database to simplify future system diagnostics.

Use the **restoreDb.sh** tool, which resides in the `BAC_home/rdu/bin` directory, to replace the current RDU database with another database. When using this tool, you must specify an input directory. This directory must contain the recovered backup snapshot of the database to be restored to the RDU server.

**Note**

Before running the **restoreDb.sh** tool, you must stop the RDU server by running the `/etc/init.d/bprAgent stop rdu` command. Also, remember to back up the database, then remove the database files from the `rdu/db` and the `rdu/dblog` directories.

When using the **restoreDb.sh** tool, you can use the `-help` option to obtain usage information.

**Examples**

```
restoreDb.sh /var/backup/rdu-backup-20070316-031028

Restoring RDU database...
Restoring from: /var/backup/rdu-backup-20070316-031028

Copying bpr.db. Size: 434176 bytes.
bpr.db: 100% completed.
```

```
Copying log.0000000001. Size: 471261 bytes.
log.0000000001: 100% completed.
```

```
Copying history.log. Size: 1260 bytes.
history.log: 100% completed.
```

```
Copying DB_VERSION. Size: 396 bytes.
DB_VERSION: 100% completed.
```

```
Database was successfully restored
You can now start RDU server.
```

In this example, the database found in the `/var/backup/rdu-backup-20070316-031028` directory is restored to the RDU server.

You must restart the RDU after the restore operation is completed. The RDU log file will contain successful startup messages.

This tool displays progress on the monitor and logs its activity in the `history.log` file.

**Note**

---

Once migration is complete, run the `verifyDb.sh` tool (described in [Database Recovery, page 14-5](#)) to check the integrity of the database. Verification is an optional task and you can skip it if shorter downtime for migration is critical.

---

## Changing Database Location

You can move the database from one partition or disk to another on the same system. You might sometimes want to do this for administrative reasons. This process requires stopping the RDU server and the BAC process watchdog.

The process of changing the database location involves changing system parameters and copying the appropriate files to the new location.

You can adjust one or both of the following parameters:

- *BAC\_data*—This parameter is initially set during installation and points to a directory that is used to store the database, and many other important files, such as logs, configuration files, and so on. This directory also stores log data for the BAC process watchdog, the DPE (if installed on the same system), the RDU, and SNMP agent, among others.
- *BAC\_dblog*—This parameter is initially set during installation and points to the directory that stores database transaction log files.

The values for the above parameters are recorded in a file called `BAC_data/bpr_definitions.sh`. Any change to this file requires that you restart all BAC components running on the system.

To change the location of the database and transaction logs:

- 
- Step 1** Run the `/etc/init.d/bprAgent stop` command to stop the BAC process watchdog and all BAC components.
  - Step 2** Make a backup copy of the `BAC_home/bpr_definitions.sh` file.
  - Step 3** Edit the file and change either or both the `BAC_data` and `BAC_dblog` parameters to new directories.
  - Step 4** Save the file.

- Step 5** Copy or move the directory structure and contents of the original *BAC\_data*, *BAC\_dblog*, or both, directories to the new locations. If you make a copy, make sure that all file and directory permissions are preserved.
  - Step 6** Run the `/etc/init.d/bprAgent start` command to start the BAC process watchdog and all BAC components.
  - Step 7** Monitor the appropriate log files to ensure that all components have successfully started.
- 

## RDU Database Migration

For information on migrating the RDU database, refer to the *Installation and Setup Guide for Cisco Broadband Access Center*, 2.7.1.



# APPENDIX **A**

## Alert and Error Messages

---

This appendix identifies all alert and error messages that Broadband Access Center (BAC) generates, specifically:

- [Syslog Alert Messages, page A-1](#)
  - [RDU Alerts, page A-2](#)
  - [Solaris DPE Alerts, page A-3](#)
  - [Watchdog Alerts, page A-5](#)
  - [Network Registrar Extension Point Alerts, page A-6](#)
- [RDU Error Messages with CCM, page A-7](#)

## Syslog Alert Messages

BAC generates alerts through the Syslog service. Syslog is a client-server protocol that manages the logging of information on Solaris. BAC syslog alerts are not a logging service; they provide a notification that a problem exists, but do not necessarily define the specific cause of the problem. you might find this information in the appropriate BAC log files.

## Message Format

When BAC generates an alert message, the format is:

*XXX-#-####: Message*

- *XXX*—Identifies the facility code, which can include:
  - RDU (Regional Distribution Unit)
  - DPE (Device Provisioning Engine)
  - AGENT (rduSnmpAgent or dpeSnmpAgent)
  - NR\_EP (Network Registrar extension points)
  - KDC

- #—Identifies the severity level in use. [Table A-1](#) describes the different levels.

**Table A-1 Severity Levels for Alert Messages**

| Severity Level | Description                         |
|----------------|-------------------------------------|
| 1              | Identifies an alert                 |
| 2              | Identifies a critical alert         |
| 3              | Identifies an error                 |
| 6              | Identifies an informational message |

- ###—Identifies the numeric error code.
- *Message*—Provides the alert text or message.

## RDU Alerts

[Table A-2](#) identifies the RDU alerts.

**Table A-2 RDU Alerts**

| Alert                                                                                                                             | Description                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RDU-1-101: RDU ran out of disk space                                                                                              | Indicates that the storage partition of the RDU server ran out of space. After encountering this error, the RDU attempts to restart automatically, but will typically encounter the same error again until more storage space is available. You can remove or compress some of the log files.<br><br>See <a href="#">Support Tools and Advanced Concepts, page 13-1</a> , for additional information on upgrading the disk. |
| RDU-1-103: RDU ran out of memory                                                                                                  | Indicates that the RDU ran out of memory. After encountering this error, the RDU server restarts automatically.                                                                                                                                                                                                                                                                                                             |
| RDU-1-111: Evaluation key for technology <i>[technology_name]</i> expired                                                         | Indicates that an evaluation key for the technology specified expired. You must contact Cisco sales or TAC for a new license key.                                                                                                                                                                                                                                                                                           |
| RDU-1-115: You have used <i>[ ]</i> percent of available <i>[technology_name]</i> licenses.                                       | Identifies, in percentage, the quantity of licenses used out of the total number of allowable licenses. Appears when you reach 80 percent of the license capacity.                                                                                                                                                                                                                                                          |
| RDU-1-122: DNS took <i>[ ]</i> seconds for lookup of address <i>[ip/hostname]</i> . Check DNS configuration and health of servers | Indicates that BAC performance may be slow due to delayed response from the DNS. The alert is generated whenever IP address lookup takes more than 60 seconds.                                                                                                                                                                                                                                                              |



**Table A-2** RDU Alerts (continued)

| Alert                                                                                                                                                                                    | Description                                                                                                                                                                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RDU-2-119: Directory [] that contains the RDU database has a filesystem block size of [] bytes that does not match the required size of [] bytes. Corruption may occur.                  | Indicates that the BAC database may not be reliable because the file system that contains the database files is not configured to support an 8-KB or greater block size.<br><br>For details on configuring the file system block size, refer to the <i>Installation and Setup Guide for the Cisco Broadband Access Center</i> , 2.7.1.     |
| RDU-2-200: Directory [] that contains the RDU database transaction logs has a filesystem block size of [] bytes that does not match the required size of [] bytes. Corruption may occur. | Indicates that the BAC database may not be reliable because the file system that contains the database log files is not configured to support an 8-KB or greater block size.<br><br>For details on configuring the file system block size, refer to the <i>Installation and Setup Guide for the Cisco Broadband Access Center</i> , 2.7.1. |
| <b>Note</b> Whenever an RDU syslog alert is sent, additional details (if any) can be found in the log file, <i>BAC_data/rdu/logs/rdu.log</i> .                                           |                                                                                                                                                                                                                                                                                                                                            |

## Solaris DPE Alerts

Whenever a DPE syslog alert is sent, you can find additional details in the DPE logs.

You can use the **show log** command to access the DPE logs. For additional information, refer to the *Cisco Broadband Access Center DPE CLI Reference*, 2.7.1.

Some DPE errors are also propagated to the RDU server log files. You can find these in the *BAC\_data/rdu/logs/rdu.log* file.

[Table A-3](#) identifies the Solaris DPE alerts.

**Table A-3** DPE Alerts

| Alert                                                                                                   | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DPE-1-102: DPE ran out of disk space                                                                    | <p>The storage partition that the DPE server uses ran out of space. You have three options:</p> <ol style="list-style-type: none"> <li>a. Clear out any excess support bundles that may reside on the disk. You can do this by moving those support bundles to another machine and then running the <b>clear bundles</b> command from the DPE CLI.</li> <li>b. Run the <b>clear logs</b> command from the DPE CLI to clear more disk space.</li> <li>c. As a last resort, run the <b>clear cache</b> command from the DPE CLI to remove any cache files and force the DPE to resynchronize with the RDU server.</li> </ol> |
| DPE-1-104: DPE ran out of memory                                                                        | <p>The DPE process ran out of memory. After encountering this error condition, the DPE restarts automatically.</p> <p>Determine how many device configurations are on the DPE; the larger the number of device configurations, the more memory is used. To reduce the number of device configurations, limit the number of devices in the provisioning groups, either primary or secondary, that the DPE serves.</p>                                                                                                                                                                                                       |
| DPE-1-109: Failed to connect to RDU                                                                     | <p>The RDU cannot be contacted. You must:</p> <ol style="list-style-type: none"> <li>a. Verify that the DPE network is configured and connected correctly.</li> <li>b. Check that the DPE is configured to connect to the proper RDU, and that the connecting port is configured properly by using the <b>dpe rdu-server</b> command.</li> <li>c. Check that the RDU process is running on the correct server and listening on the correct port. The DPE attempts to reconnect to the RDU process every few seconds until a connection is established.</li> </ol>                                                          |
| DPE-1-117: DPE license nodes have been exceeded or there is no valid DPE license                        | <p>Indicates that the BAC watchdog agent, which starts the DPE, did not detect a license for the DPE.</p> <p>Enter the license key for the DPE using the administrator user interface. If you do not have a license, contact your Cisco representative.</p>                                                                                                                                                                                                                                                                                                                                                                |
| DPE-1-116: DPE evaluation license has expired. Dropping DPE connections and deleting DPEs from database | <p>Indicates that an evaluation license key for the DPE expired. You must contact Cisco sales or TAC for a new license key.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

**Table A-3 DPE Alerts (continued)**

| Alert                                                                                                                                                                     | Description                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| DPE-2-118: Directory [ ] that contains the DPE's cache has a filesystem block size of [ ] bytes that does not match the required size of [ ] bytes. Corruption may occur. | Indicates that the DPE cache may not be reliable because the file system is not configured to support an 8-KB or greater block size.<br><br>For details on configuring the file system block size, refer to the <i>Installation and Setup Guide for the Cisco Broadband Access Center, 2.7.1</i> . |
| DPE-1-121: Cannot start the server due to an invalid encryption key.                                                                                                      | Indicates that the DPE could not be started because of an invalid encryption key.                                                                                                                                                                                                                  |

## Watchdog Alerts

Whenever the watchdog agent process sends a syslog alert, you can find error details (if any) in the *BPR\_DATA/agent/logs/agent\_console.log* file and the log files corresponding to the specific component mentioned in the alert (if any). For example, if you receive an alert similar to *The rdu unexpectedly terminated*, you would check the RDU server log file (*BPR\_DATA/rdu/logs/rdu.log*) for additional information. [Table A-4](#) identifies the watchdog agent alerts.

**Table A-4 Watchdog Agent Alerts**

| Alert                                                                                                                                                  | Description                                                                                                                         |
|--------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| AGENT-3-9001: Failed to start the [component]                                                                                                          | Indicates that the watchdog has failed to start the specified component.                                                            |
| AGENT-3-9002: The [component] unexpectedly terminated                                                                                                  | Indicates that the specified component, monitored by the agent process, has unexpectedly failed.                                    |
| AGENT-6-9004: The [component] has started                                                                                                              | Generated any time a component is successfully started by the watchdog agent. This message is for informational purposes only.      |
| AGENT-6-9005: The [component] has stopped                                                                                                              | Generated any time a component is successfully stopped through the watchdog agent. This message is for informational purposes only. |
| AGENT-3-9003: Failed to stop the [component]                                                                                                           | Indicates that a component did not stop when the watchdog agent attempted to stop it.                                               |
| AGENT-3-9003:<br>Failed to create listener thread;<br>[error no]Failed to close listen socket;<br>[error no] Failed to cancel listen thread, and so on | Indicates errors that are not defined in other alert messages.                                                                      |

The [component] variable presented in the watchdog agent alerts list shown in [Table A-4](#) represents any of these component values:

- rdu
- dpe
- tomcat
- cli

- snmpAgent
- kdc

## Network Registrar Extension Point Alerts

Whenever a BAC Network Registrar extension point syslog alert is sent, you can find additional details in the Network Registrar log file. [Table A-5](#) identifies the watchdog agent alerts.

**Table A-5** Network Registrar Extension Alerts

| Alert                                           | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NR_EP-1-106: Failed to connect to RDU           | <p>The Network Registrar server cannot connect to the RDU. You should verify that the RDU process is running and, if it is not already running, start the RDU.</p> <p>If the RDU is running, use the Network Registrar computer to ping the RDU. If you are unable to ping the RDU, fix the routing tables or other communication parameters, between the two devices.</p> <p>If this alert is frequently repeated, you may have an unstable connection between the two hosts. Use generally accepted network troubleshooting techniques to improve the connectivity between the two hosts.</p>                                                                                                                                                                                                                                             |
| NR_EP-1-107: Failed to connect to any DPEs      | <p>The Network Registrar extension cannot connect to the DPEs.</p> <p>Check that there are DPEs in the provisioning group for each Network Registrar extension. If not, change the Network Registrar provisioning group to one that has DPEs available. If DPEs are in the provisioning group, ensure that the Network Registrar extension has registered with the RDU, if it has not, it will not recognize any of the DPEs.</p> <p>If, after completing the check, the alert continues, check that there is network connectivity between the Network Registrar extension and the DPEs in the provisioning group.</p> <p>If this alert is frequently repeated, you may have an unstable connection between the two hosts. Use generally accepted network troubleshooting techniques to improve the connectivity between the two hosts.</p> |
| NR_EP-6-108: The BAC NR extensions have started | The Network Registrar extensions have been started.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| NR_EP-6-109: The BAC NR extensions have stopped | The Network Registrar extensions have been stopped.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

**Table A-5 Network Registrar Extension Alerts (continued)**

| Alert                                                        | Description                                                                                                                                                                             |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| NR_EP-6-110: Registered with RDU [ <i>address and port</i> ] | The Network Registrar extensions have been registered with the RDU. The <i>address and port</i> identifies the address of the RDU that has registered the Network Registrar extensions. |
| NR_EP-1-111: Failed to find usable (best) DPEs               | The Network Registrar extensions are unable to find a usable DPE.                                                                                                                       |

## RDU Error Messages with CCM

The RDU error messages described in this section are accompanied by corrective actions that you perform to remedy specified problems. These error messages are recorded in the *rdu.log* file, and are a direct result of the Lease Reservation feature. Consequently, if you are not using the Network Registrar Regional CCM feature, these error messages should not appear.

BAC generates three types of lease reservation-related error messages, including:

- Failure to add a reservation request failure. These messages are preceded by the following information, which identifies several aspects of the error:

```
rdu.cisco.com: 2005 02 09 13:25:11 EST: %BPR-RDU-3-1146: PACE-13: Failed to add
reservation [10.10.10.1] for [1,6,03:03:14:00:00:21] using client-class
[unprovisioned-docsis] and selection-tag [NULL]; [AddReservation: FORWARD_FAILED]
```

- Failure to remove a reservation request failure. These messages are preceded by the following information, which identifies several aspects of the error:

```
rdu.cisco.com: 2005 02 09 13:25:11 EST: %BPR-RDU-3-1147: Failed to remove reservation
[10.10.10.1] for [1,6,03:03:14:00:00:21];
```

- Selection-criteria exclusion tag errors. These messages are preceded by the following information, which identifies several aspects of the error:

```
rdu.cisco.com: 2005 02 09 13:25:11 EST: %BPR-RDU-4-1145: Use of selection-criteria
exclusion tags [black] is not allowed when adding a lease reservation. They will be
ignored.
```

Where, from the preceding examples:

- `rdu.cisco.com:`—Identifies the RDU FQDN
- `2005 02 09 13:25:11 EST:`—Identifies the date and time that the error took place.
- `%BPR-RDU-#-####:`—Identifies the RDU error number (#) and error message ID (####).
- `[10.10.10.1]`—Identifies the IP address being reserved.
- `1,6,03:03:14:00:00:21]`—Identifies the MAC address that is attempting to reserve an IP address. Note that the second example illustrates BAC support for variable length MAC addresses.

Each of these examples contain a generalized description of the error as well as the error message that is returned from Network Registrar.

The following sections describe error messages that BAC generates.

## [OBJECT\_EXISTS]

A user is trying to make reservation [11.100.14.21] for MAC [1,6,03:03:14:00:00:21], but the reservation request failed due to duplicate IP used (IP is already being reserved by another device in the database).

**Corrective Action:**

Use a different IP address or free up the desired IP address.

## [RemoveReservation: NOT\_FOUND]

A user is trying to remove a reservation [11.100.0.103] for MAC device [1,6,01:02:03:04:05:06]; the request failed due to nonexistence of reservation in CCM database. In other words, CCM is unable to find the reservation in the database.

**Corrective Action:**

Verify if the reservation exists in the CCM database.

## [AddReservation: INVALID\_PARENT]

This means that CCM did not find any scope for the IP address of the reservation. CCM local has to find a scope that could contain the reservation. CCM displays this error message when reserved IP is out of scope. You get this error message if you try to add a reservation to a nonexisting subnet or scope.

**Corrective Action:**

Recheck your DHCP server configuration (scope, subnet, and so on).

## [AddReservation: INVALID\_SECOND\_PARENT]

This should be returned only by a local CCM. It means that no scope existing in the configuration could contain the reservation. Recheck your DHCP server configuration (scope, subnet, and so on).

You get this error if you try to add a reservation to a subnet that exists at the Regional Cluster but no such scope/subnet exists at the local cluster(s):

Or CCM cannot find a scope that contains both the reserved IP address and matching clientclass/selection-tags criteria.

**Corrective Action:**

Recheck your DHCP server configuration (scope, subnet, clientclass, selection-tags, and so on).

## [AddReservation: FORWARD\_FAILED]

CCM regional was not able to forward the reservation request to any local CCM. This happens when the CCM cannot find a scope that contains both the reserved IP address and matching clientclass/selection-tags criteria.

The following two use cases illustrate solutions to potential causes of this error:

### Example Case #1

Assume that there are three scopes with these attributes/tags; scope A is red and gray, scope B is blue and red, and scope C is blue and green. If the AddReservation API call specifies the inclusion of red and gray tags, but the scope that contains the IP address requested is scope B.

The outcome of this situation is that the reservation request fails and this error is logged in the *rdu.log* file.

#### Corrective Action

Use the same setup or configuration, but modify the AddReservation API call to specify the inclusion only red tags, rather than red and gray as described above. This should be successful and any scope that contains the IP address, and matches all of the selection tags, is used.



#### Note

---

In this example, it is expected that the device gets the reservation from scope B.

---

### Example Case #2

Assume that there are three scopes with these attributes/tags; scope A is red, scope B is blue, and scope C is green. Also assume that the AddReservation API call specifies a client class of test1, Network Registrar defines test1 with the green selection criteria, and the IP being requested is contained in the scope B.

The outcome of this situation is that the reservation request fails and this error is logged in the *rdu.log* file.

#### Corrective Action

Use the same setup or configuration, but reconfigure the test1 clientclass in Network Registrar with selection criteria blue. This should successfully correct the problem so that any scope containing IP address, and matching all of the selection tags, is used.

## [AddReservation: AX\_ETIME]

This indicates that a reservation request timeout has occurred. This could be due to the CCM being overloaded, or busy with tasks and requests, and was not able to process the reservation request.

#### Corrective Action

Use the administrator user interface to configure a longer timeout value. See [RDU Defaults, page 11-19](#), for information on submitting reservation requests.

## [AddReservation: INVALID\_OBJECT]

The reservation itself is invalid: the MAC address is invalid, the IP address is missing, or it supplied an invalid client-class name or selection tags.

**Corrective Action:**

Check the reservation requests: the MAC address, IP address, client class, selection tags, and so on.

## Selection-criteria exclusion tags will be ignored

The use of selection-criteria exclusion tags [black] is not allowed when adding a lease reservation. They will be ignored.

**Corrective Action:**

You should not configure selection-criteria exclusion tags in the DHCP criteria since the use of selection-criteria exclusion tags is not allowed.

## [AX\_EIO]

This indicates that the connection, or session, between the RDU and the CCM is broken.

**Corrective Action:**

There is no user interaction required when this error occurs. The RDU automatically establishes another connection to CCM.

## [AX\_EPIPE]

This indicates that the connection, or session, between the RDU and the CCM is broken.

**Corrective Action:**

There is no user interaction required when this error occurs. The RDU automatically establishes another connection to CCM.





# APPENDIX **B**

## PacketCable DHCP Options to BAC Properties Mapping

This appendix identifies the mapping of BAC properties to the PacketCable DHCP options used for PacketCable provisioning.

The minimum required set of these properties is configured, during installation, in the *BAC\_home/cnr\_ep/conf/cnr\_ep.properties* file. This file resides on the Network Registrar host. The set of properties defined in *cnr\_ep.properties* is applied to all PacketCable voice technology devices in the provisioning group. Like other BAC properties, you can also set these properties on a device or a Class of Service. Setting them at the RDU, using either the administrator user interface or the API, overrides the corresponding values set in the *cnr\_ep.properties* file.



**Note**

See [Using the KeyGen Tool, page 13-11](#), for information on changing these key configuration properties.

BAC supports both PacketCable DHCP Option 122 (as specified in RFC 3495 and 3594) and the deprecated PacketCable DHCP Option 177. BAC does not ignore DHCP requests when it cannot populate option 122 and/or 177 content. Whatever Option 122/177 content is available is populated and the decision to ignore the option is left to the eMTA.

When BAC receives a DHCP request asking for both option 122 and 177, BAC will ignore the request for Option 177 and populate only Option 122 content.



**Caution**

There should be only one instance of each property in *BAC\_home/cnr\_ep/conf/cnr\_ep.properties*.

## Option 122 and BAC Property Comparison

[Table B-1](#) identifies the BAC properties as they apply to the definition of Option 122 in RFC-3495 and RFC-3594.

**Table B-1** DHCP Option 122 to BAC Property Comparison

| DHCP Option | Type    | BAC Property Name  |
|-------------|---------|--------------------|
| 6           | IP addr | /ccc/dns/primary   |
| 6           | IP addr | /ccc/dns/secondary |
| 122.1       | IP addr | /ccc/dhcp/primary  |

**Table B-1** DHCP Option 122 to BAC Property Comparison (continued)

| DHCP Option | Type    | BAC Property Name                                                                                                           |
|-------------|---------|-----------------------------------------------------------------------------------------------------------------------------|
| 122.2       | IP addr | /ccc/dhcp/secondary                                                                                                         |
| 122.3       | FQDN    | /ccc/prov/fqdn<br><b>Note</b> Option 122.3 is automatically filled by BAC; consequently, do not set this property manually. |
| 122.4       | Integer | /ccc/kerb/auth/backoff/nomTimeout<br>/ccc/kerb/auth/backoff/maxTimeout<br>/ccc/kerb/auth/backoff/maxRetries                 |
| 122.5       | Integer | /ccc/kerb/app/backoff/nomTimeout<br>/ccc/kerb/app/backoff/maxTimeout<br>/ccc/kerb/app/backoff/maxRetries                    |
| 122.6       | String  | /ccc/kerb/realm                                                                                                             |
| 122.7       | Boolean | /ccc/tgt                                                                                                                    |
| 122.8       | Integer | /ccc/prov/timer                                                                                                             |
| 122.9       | Integer | /ccc/security/ticket/invalidation                                                                                           |

**Caution**

If any of /ccc/kerb/auth/backoff/nomTimeout, /ccc/kerb/auth/backoff/maxTimeout, or /ccc/kerb/auth/backoff/maxRetries are defined, they must all be defined. Similarly, if any of /ccc/kerb/app/backoff/nomTimeout, /ccc/kerb/app/backoff/maxTimeout, or /ccc/kerb/app/backoff/maxRetries are defined, they must all be defined.

## Option 177 and BAC Property Comparison

In accordance with PacketCable compliance wave 26, Option 177 is deprecated, and Option 122 is now the preferred MTA provisioning option. For legacy devices that still support Option 177, [Table B-2](#) identifies the BAC properties as they apply to the definition of Option 177.

**Table B-2** DHCP Option 177 to BAC Property Comparison

| Option 177 | Type    | BAC Property Names         |
|------------|---------|----------------------------|
| 177.1      | ip addr | /pktcbl/dhcp/primary       |
| 177.2      | ip addr | /pktcbl/dhcp/secondary     |
| 177.3      | fqdn    | /pktcbl/snmp/entity/fqdn   |
| 177.4      | ip addr | /pktcbl/dns/primary        |
| 177.5      | ip addr | /pktcbl/dns/secondary      |
| 177.6      | string  | /pktcbl/snmp/realm         |
| 177.7      | boolean | /pktcbl/snmp/tgt           |
| 177.8      | integer | /pktcbl/provisioning/timer |

**Table B-2** *DHCP Option 177 to BAC Property Comparison (continued)*

| <b>Option 177</b> | <b>Type</b> | <b>BAC Property Names</b>                                                                                                                                      |
|-------------------|-------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 177.10            | integer     | /pktcbl/kerberos/authentication/backoff/nomTimeout<br>/pktcbl/kerberos/authentication/backoff/maxTimeout<br>/pktcbl/kerberos/authentication/backoff/maxRetries |
| 177.11            | integer     | /pktcbl/kerberos/application/backoff/nomTimeout<br>/pktcbl/kerberos/application/backoff/maxTimeout<br>/pktcbl/kerberos/application/backoff/maxRetries          |
| 177.12            | integer     | /pktcbl/snmp/kerberos/ticket/invalidation                                                                                                                      |





## APPENDIX **C**

# API Use Cases

---

This appendix presents a series of the most common provisioning application programming interface (API) use cases, including pseudo-code segments that can be used to model typical service provider workflows. The pseudo code used in the use cases resembles java though it is not intended for direct compilation. Please refer to the Broadband Access Center (BAC) 2.7.1 API Javadoc for more details and sample Java code segments explaining individual API calls and features.

These use cases are directly related to device (and/or service) provisioning. Many administrative operations, such as managing Class of Service, DHCP criteria, and licenses are not addressed here. It is highly recommended to go through the API javadoc for more details on the related API calls. You can also use the administrator user interface to perform most of these activities.

## Use Cases

This appendix includes these use cases:

- [Self-Provisioned Modem and Computer in Fixed Standard Mode, page C-2](#)
- [Adding a New Computer in Fixed Standard Mode, page C-5](#)
- [Disabling a Subscriber, page C-7](#)
- [Preprovisioning Modems/Self-Provisioned Computers, page C-9](#)
- [Modifying an Existing Modem, page C-11](#)
- [Unregistering and Deleting a Subscriber's Devices, page C-12](#)
- [Self-Provisioning First-Time Activation in Promiscuous Mode, page C-14](#)
- [Bulk Provisioning 100 Modems in Promiscuous Mode, page C-17](#)
- [Preprovisioning First-Time Activation in Promiscuous Mode, page C-19](#)
- [Replacing an Existing Modem, page C-20](#)
- [Adding a Second Computer in Promiscuous Mode, page C-21](#)
- [Self-Provisioning First-Time Activation with NAT, page C-21](#)
- [Adding a New Computer Behind a Modem with NAT, page C-23](#)
- [Move Device to Another DHCP Scope, page C-24](#)
- [Log Device Deletions Using Events, page C-25](#)
- [Monitoring an RDU Connection Using Events, page C-26](#)
- [Logging Batch Completions Using Events, page C-27](#)

- [Getting Detailed Device Information, page C-27](#)
- [Searching Using the Default Class of Service, page C-28](#)
- [Retrieving Devices Matching a Vendor Prefix, page C-30](#)
- [Preprovisioning PacketCable eMTA, page C-32](#)
- [SNMP Cloning on PacketCable eMTA, page C-34](#)
- [Incremental Provisioning of PacketCable eMTA, page C-35](#)
- [Preprovisioning DOCSIS Modems with Dynamic Configuration Files, page C-37](#)
- [Optimistic Locking, page C-39](#)
- [Temporarily Throttling a Subscriber's Bandwidth, page C-40](#)
- [Preprovisioning CableHome WAN-MAN, page C-41](#)
- [CableHome with Firewall Configuration, page C-43](#)
- [Retrieving Device Capabilities for CableHome WAN-MAN, page C-45](#)
- [Self-Provisioning CableHome WAN-MAN, page C-47](#)
- [Lease Reservation Use Cases, page C-49](#)

## Self-Provisioned Modem and Computer in Fixed Standard Mode

The subscriber has a computer installed in a single dwelling unit and has purchased a DOCSIS cable modem. The computer has a web browser installed.

### Desired Outcome

Use this workflow to bring a new unprovisioned DOCSIS cable modem and computer online with the appropriate level of service.

- 
- Step 1** The subscriber purchases and installs a DOCSIS cable modem at home and connects a computer to it.
  - Step 2** The subscriber powers on the modem and the computer, and BAC gives the modem restricted access. The computer and modem are assigned IP addresses from restricted access pools.
  - Step 3** The subscriber starts a web browser, and a spoofing DNS server points the browser to a service provider's registration server (for example, an OSS user interface or a mediator).
  - Step 4** The subscriber uses the service provider's user interface to complete the steps required for registration, including selecting Class of Service.
  - Step 5** The service provider's user interface passes the subscriber's information, such as the selected Class of Service and computer IP address, to BAC, which then registers the subscriber's modem and computer.

```
// First we query the computer's information to find the modem's
// MAC address. We use the computers IP address (the web browser
// received this when the subscriber opened the service providers
// web interface)

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a query
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device

LeaseResults computerLease =
 getAllForIPAddress("10.0.14.38");
 // ipAddress: restricted access.

// Derive the modem MAC address from the computer's network
// information. The 1,6, is a standard prefix for an Ethernet
// device. The fully qualified MAC address is required by BACC

String modemMACAddress = "1,6," +
 computerLease.getSingleLease().get(RELAY_AGENT_REMOTE_ID);

// Now let's provision the modem and the computer in the same
// batch. This can be done because the activation mode of this
// batch is NO_ACTIVATION. More than one device can be operated
// on in a batch if the activation mode does not lead to more
// than one device being reset. NO_ACTIVATION will generate a
// configuration for the devices. However it will not attempt
// to reset the devices. The configuration can be generated
// because the devices have booted. NO_CONFIRMATION is the
// confirmation mode because we are not attempting to reset
// the modem. We do not want to reset the modem here because
// we want to notify the user to reset their computer. If we
// reset the modem in this batch, we will not be able to notify
// the user of anything until the modem has come back online.
// To add a DOCSIS modem:

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

add(
 DeviceType: DOCSIS,
 // deviceType: DOCSIS
 modemMACAddress,
 // macAddress: derived from computer lease
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "Silver",
 // ClassOfService
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
```

```

 null
 // properties: not used
);
 // properties: not used
String computerMACAddress = computerLease.
 getSingleLease().get(DeviceDetailsKeys.MAC_ADDRESS);
// Create a Map for the computer's properties

Map properties;

// Setting the property IPDeviceKeys.MUST_BE_BEHIND_DEVICE
// on the computer ensures that when the computer boots it
// will only receive its provisioned access when it is behind
// the given device. If it is not behind the given device,
// it will receive default access (unprovisioned). This makes
// the computer "fixed" behind the specified modem.

properties.put(IPDeviceKeys.MUST_BE_BEHIND_DEVICE,
 modemMACAddress);

add(
 DeviceType.COMPUTER,
 // deviceType: COMPUTER
 computerMACAddress,
 // macAddress: derived from computer lease
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 null,
 // classOfService : get the default COS
 "provisionedCPE",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 properties
 // properties:
);

```

**Step 6** The user interface prompts the subscriber to reboot the computer.

**Step 7** The provisioning client calls `performOperation(DeviceOperation.RESET, modemMACAddress, null)` to reboot the modem and gives the modem provisioned access.



```
get-new-batch(AUTOMATIC, NO_CONFIRMATION);

// AUTOMATIC is the activation mode because we are attempting
// to reset the modem so that it receives its new class of service.
// NO_CONFIRMATION is the confirmation mode because we don't
// want the batch to fail if we can't reset the modem. The user
// may have power cycled the modem when they rebooted their computer.
// Send a batch to reset the modem now that the user has been
// notified to reboot their computer.

performOperation(
 DeviceOperation.RESET,
 //deviceOperation: Reset operation
 modemMACaddress,
 // macAddress:Modem's MAC address
 null
 // properties: not used
);
```

- Step 8** After rebooting, the computer receives a new IP address, and both cable modem and computer are now provisioned devices. The computer has access to the Internet through the service provider's network.
- 

## Adding a New Computer in Fixed Standard Mode

A multiple system operator (MSO) lets a subscriber have two computers behind a cable modem. The subscriber has one computer already registered and then brings home a laptop from work and wants access. The subscriber installs a hub and connects the laptop to it.

### Desired Outcome

Use this workflow to bring a new unprovisioned computer online with a previously provisioned cable modem so that the new computer has the appropriate level of service.

---

- Step 1** The subscriber powers on the new computer and BAC gives it restricted access.
- Step 2** The subscriber starts a web browser on the new computer and a spoofing DNS server points it to the service provider's registration server (for example, an OSS user interface or a mediator).
- Step 3** The subscriber uses the service provider's user interface to complete the steps required to add a new computer.
- Step 4** The service provider's user interface passes the subscriber's information, such as the selected Class of Service and computer IP address, to BAC, which then registers the subscriber's modem and computer.

```

// First we query the computer's lease information to its
// MAC address. We use the computers IP address (the web browser
// received this when the subscriber opened the service providers
// web interface)

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

/ NO_ACTIVATION is the activation mode because this is a query
/ NO_CONFIRMATION is the confirmation mode because we are not
/ attempting to reset the device.

LeaseResults computerLease =
 getAllForIPAddress("10.0.14.39");
 // ipAddress: restricted access.

String computerMACAddress = computerLease.
 getSingleLease().get(DeviceDetailsKeys.MAC_ADDRESS);

// We have the MAC address now. Let's add the computer to BACC.

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION will generate a configuration for the computer.
// However it will not attempt to reset the computer (this
// can not be done). The configuration can be generated because
// the device has booted. NO_CONFIRMATION is the confirmation
// mode because we are not attempting to reset the modem.

Map properties; // Map containing the properties for the computer

// Setting the property IPDeviceKeys.MUST_BE_BEHIND_DEVICE on
// the computer ensures that when the computer boots, it will
// only receive its provisioned access when it is behind the
// given device. If it is not behind the given device, it
// will receive default access (unprovisioned) and hence the
// fixed mode.

properties.put(IPDeviceKeys.MUST_BE_BEHIND_DEVICE, modemMACAddress);

// The IPDeviceKeys.MUST_BE_IN_PROV_GROUP property ensures that
// the computer will receive its provisioned access only when
// it's brought up in the specified provisioning group. This prevents
// the computer (and/or the modem) from moving from one locality to
// to another locality.

properties.put(IPDeviceKeys.MUST_BE_IN_PROV_GROUP, "bostonProvGroup");

add(
 DeviceType.COMPUTER,
 // deviceType: COMPUTER
 computerMACAddress,
 // macAddress: derived from computer lease
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",

```

```
 // ownerID: here, account number from billing system
 null,
 // classOfService: get the default COS
 "provisionedCPE",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 properties
 // properties:
);
```

- Step 5** The user interface prompts the subscriber to reboot the new computer so that BAC can give the computer its registered service level.
- Step 6** The computer is now a provisioned device with access to the appropriate level of service.
- 

## Disabling a Subscriber

A service provider needs to disable a subscriber from accessing the Internet due to recurring nonpayment.

### Desired Outcome

Use this workflow to disable an operational cable modem and computer, so that the devices temporarily restrict Internet access for the user. Additionally, this use case can redirect the user's browser to a special page that could announce:

```
You haven't paid your bill so your Internet access has been disabled.
```

- Step 1** The service provider's application uses a provisioning client program to request a list of all of the subscriber's devices from BAC. The service provider's application then uses a provisioning client to individually disable or restrict each of the subscriber's devices.

```
// The service provider's application uses a provisioning client
// to get a list of all of the subscriber's devices.

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a query.
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the devices.

getAllForOwnerID(
 "0123-45-6789"); // Query all the devices for this account number
// For each device in the retrieved list:
// We are only interested in the modems. If we disable network
// access of the modems, we disable network access for all the
// subscriber's devices. If we are using roaming standard mode,
// we may also change the computer's network access (DHCPCriteria)
// because the subscriber may still be able to access the network
// from a different modem.

DeviceLoop:
{
 if (Device.deviceType == DOCSIS_MODEM)
 {
 get-new-batch(AUTOMATIC, NO_CONFIRMATION)

 // AUTOMATIC is the activation mode because we are
 // attempting to reset the modem so that becomes
 // disabled. NO_CONFIRMATION is the confirmation mode
 // because we do not want the batch to fail if we cannot
 // reset the modem. If the modem is off, it will
 // be disabled when it is turned back on.

 // Let's change the COS of the device so that it will restrict
 // the bandwidth usage of the modem.

 changeClassOfService(
 Device.MAC_ADDRESS,
 // macAddress: unique identifier for this modem
 "DisabledCOS");
 // newClassOfService: restricts bandwidth usage

 // Let's change the DHCP Criteria so that the modem will get an IP
 // address from a disabled CNR scope. This scope also points to
 // a spoofing DNS server so that the subscriber gets a restricted
 // access page.

 changeDHCPCriteria(
 Device.MAC_ADDRESS,
 // macAddress: unique identifier for this modem
 "DisabledDHCPCriteria");
 // newDHCPCriteria: disables Internet access
 }
}
```

```
 }
 }
 // end DeviceLoop
```

**Note**

You may need to consider the impact on the CPE behind the modem when defining the characteristics of DisabledCOS and resetting the modem. This is especially important if you have voice end points behind the modem, because disrupting the cable modem might affect the telephone conversation in progress at that time.

The subscriber is now disabled.

## Preprovisioning Modems/Self-Provisioned Computers

A new subscriber contacts the service provider and requests service. The subscriber has a computer installed in a single dwelling unit. The service provider preprovisions all its cable modems in bulk.

**Desired Outcome**

Use this workflow to bring a preprovisioned cable modem, and an unprovisioned computer, online in the roaming standard mode. This must be done so that both devices have the appropriate level of service and are registered.

- Step 1** The service provider chooses a subscriber username and password for the billing system.
- Step 2** The service provider selects services that the subscriber can access.
- Step 3** The service provider's field technician installs the physical cable to the new subscriber's house and installs the preprovisioned device, connecting it to the subscriber's computer.
- Step 4** The technician turns on the modem and BAC gives it a provisioned IP address.
- Step 5** The technician turns on the computer and BAC gives it a private IP address.
- Step 6** The technician starts a browser application on the computer and points the browser to the service provider's user interface.
- Step 7** The technician accesses the service provider's user interface to complete the steps required for registering the computer behind the provisioned cable modem.

```

// To provision a computer:
// First we query the computer's lease information to its
// MAC address. We use the computers IP address (the web browser
// received this when the subscriber opened the service provider's
// web interface)

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a query
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device

LeaseResults computerLease =
 getAllForIPAddress("10.0.14.38");
 // ipAddress:

String computerMACAddress = computerLease.
 getSingleLease().get(DeviceDetailsKeys.MAC_ADDRESS);

// MSO admin UI calls the provisioning API to provision a computer.

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION will generate a new configuration for the computer
// however it will not attempt to reset it.
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the computer because this cannot be done.

add(
 DeviceType.COMPUTER,
 // deviceType: Computer
 computerMACAddress,
 // macAddress: derived from computer lease
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 null,
 // ClassOfService : get the default COS
 "provisionedCPE",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 null
 // properties: not used
);

```

**Note**

The `IPDeviceKeys.MUST_BE_BEHIND_DEVICE` property is not set on the computer and this allows roaming from behind one cable modem to another.

- Step 8** The technician restarts the computer and the computer receives a new provisioned IP address. The cable modem and the computer are now both provisioned devices. The computer has access to the Internet through the service provider's network.

## Modifying an Existing Modem

A service provider's subscriber currently has a level of service known as **Silver** and has decided to upgrade to **Gold** service. The subscriber has a computer installed at home.



### Note

The intent of this use case is to show how to modify a device. You can apply this example to devices provisioned in modes other than roaming standard.

### Desired Outcome

Use this workflow to modify an existing modem's Class of Service and pass that change of service to the service provider's external systems.

- Step 1** The subscriber phones the service provider and requests to have service upgraded. The service provider uses its user interface to change the Class of Service from **Silver** to **Gold**.
- Step 2** The service provider's application makes these API calls in BAC:

```
get-new-batch(AUTOMATIC, NO_CONFIRMATION)

// AUTOMATIC will generate a configuration for the device
// and will attempt to reset the device
// The configuration will be able to be generated because
// the device has booted. NO_CONFIRMATION is the confirmation
// mode because we don't want the batch to fail if the reset failed.
// This use case is a perfect example of the different
// confirmation modes that could be used instead of
// NO_CONFIRMATION. These confirmation modes give you
// a method to test whether a configuration was taken by
// a device. Also, these modes will take more time because
// the batch has to wait for the modem to reset.

changeClassOfService(
 "1,6,00:11:22:33:44:55",
 // macAddress: unique identifier for this modem
 "Gold"
 // newClassOfService
);
```

The subscriber can now access the service provider's network with the *Gold* service.

# Unregistering and Deleting a Subscriber's Devices

A service provider needs to delete a subscriber who has discontinued service.

## Desired Outcome

Use this workflow to permanently remove all the subscriber's devices from the service provider's network.

- 
- Step 1** The service provider's user interface discontinues service to the subscriber.
- Step 2** The service provider's application uses a provisioning client program to request a list of all the subscriber's devices from BAC, and unregisters and resets each device so that it is brought down to the default (unprovisioned) service level.



## Note

If the device specified as the parameter to the "unregister" API is already in unregistered state then the status code from the API call will be set to `CommandStatusCodes .CMD_ERROR_DEVICE_UNREGISTER_UNREGISTERED_ERROR`. This is normal/expected behavior.

---

```
// MSO admin UI calls the provisioning API to get a list of
// all the subscriber's devices.

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a query.
// NO_CONFIRMATION is the confirmation mode because we are
// not attempting to reset the devices.

getAllForOwnerID(
 "0123-45-6789");
 // query all the devices for this account number

// We need to unregister all the devices behind each modem(s) or else the
// unregister call for that modem will fail.

// for each computer in the retrieved list:
DeviceLoop:
{

 if (Device.deviceType == COMPUTER)
 {

 get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

 // NO_ACTIVATION is the activation mode because we can't
 // to reset Computers.
 // NO_CONFIRMATION is the confirmation mode because
 // the unregister call will not reset the devices.

 unregister(
 Device.MAC_ADDRESS,
 // macAddress: unique identifier for this device
);

 }
}
```



```
}

// for each modem in the retrieved list:
DeviceLoop:
{

 if (Device.deviceType == DOCSIS_MODEM)
 {

 get-new-batch(AUTOMATIC, NO_CONFIRMATION)

 // AUTOMATIC is the activation mode because we want
 // to reset as we unregister the device.

 unregister(
 Device.MAC_ADDRESS,
 // macAddress: unique identifier for this device
);

 }
}

// end DeviceLoop.
```

**Note**

---

The next step is optional as some service providers prefer to keep the cable modem in the database unless it is broken. This step needs to be run only if the devices need to be deleted from the database.

---

**Step 3**

The service provider's application uses a provisioning client program to delete each of the subscriber's remaining devices individually from the database.

```

// MSO admin UI calls the provisioning API to get a list of
// all the subscriber's devices.

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a query.
// NO_CONFIRMATION is the confirmation mode because we are
// not attempting to reset the devices.

getAllForOwnerID(
 "0123-45-6789");
 // query all the devices for this account number
// for each device in the retrieved list:

DeviceLoop:
{
 // get a new batch for each modem being deleted

 if (Device.deviceType == DOCSIS_MODEM)
 {
 get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

 // NO_ACTIVATION is the activation mode because we don't want
 // to reset as we are deleting the device.
 // NO_CONFIRMATION is the confirmation mode because
 // the delete call will not reset the devices.

 delete(
 Device.MAC_ADDRESS,
 // macAddress: unique identifier for this device
 true
 // deleteDevicesBehind: deletes CPEs behind this modem.
);
 }
}
//end DeviceLoop.

```

## Self-Provisioning First-Time Activation in Promiscuous Mode

The subscriber has a computer (with a browser application) installed in a single dwelling unit and has purchased a DOCSIS cable modem.

### Desired Outcome

Use this workflow to bring a new unprovisioned DOCSIS cable modem and computer online with the appropriate level of service.

- 
- Step 1** The subscriber purchases a DOCSIS cable modem and installs it at home.
  - Step 2** The subscriber powers on the modem, and BAC gives it restricted access.

- Step 3** The subscriber starts a browser application on the computer and a spoofing DNS server points the browser to the service provider's registration server (for example, an OSS user interface or a mediator).
- Step 4** The subscriber uses the service provider's user interface to complete the steps required for registration, including selecting a Class of Service.

The service provider's user interface passes the subscriber's information to BAC, including the selected Class of Service and computer IP address. The subscriber's cable modem and computer are then registered with BAC.

```

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a
// query. NO_CONFIRMATION is the confirmation mode because
// we are not attempting to reset the device.
// First we query the computer's information to find the
// modems MAC address.
// We use the computer's IP address (the web browser
// received this when the subscriber opened the service
// provider's web interface). We also assume that "bostonProvGroup"
// is the provisioning group used in that locality.

List provGroupList;
provGroupList = provGroupList.add("bostonProvGroup");
Map computerLease = getAllForIPAddress(
 "10.0.14.38",
 // ipAddress: restricted access computer lease
 provGroupList
 // provGroups: List containing provgroup)

// Derive the modem MAC address from the computer's network
// information. The 1,6, is a standard prefix for an Ethernet
// device. The fully qualified MAC address is required by BACC

String modemMACAddress = "1,6," +
 computerLease.getSingleLease().get(RELAY_AGENT_REMOTE_ID);

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// Now let's provision the modem
// NO_ACTIVATION will generate a configuration for the
// modem, however it will not attempt to reset it
// The configuration will be able to be generated because
// the modem has booted.
// NO_CONFIRMATION is the confirmation mode because we
// are not attempting to reset the modem
// Create a Map for the properties of the modem

Map properties;
// Set the property ModemKeys.PROMISCUOUS_MODE_ENABLED
// to enable promiscuous mode on modem

properties.put(ModemKeys.PROMISCUOUS_MODE_ENABLED, "enabled");

properties.put(ModemKeys.CPE_DHCP_CRITERIA, "provisionedCPE");

add(
 DeviceType.DOCSIS,

```

```

 // deviceType: DOCSIS
modemMACaddress,
 // macAddress: derived from computer lease
null,
 // hostName: not used in this example
null,
 // domainName: not used in this example
"0123-45-6789",
 // ownerID: here, account number from billing system
"Silver",
 // ClassOfService
"provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
properties
 // properties:
);

```

**Step 5** The user interface prompts the subscriber to reboot the computer.

**Step 6** The provisioning client calls `performOperation()` to reboot the modem and gives the modem provisioned access.

```

get-new-batch(AUTOMATIC, NO_CONFIRMATION)

// AUTOMATIC is the activation mode because we are attempting
// to reset the modem so that it receives the new class of service.
// NO_CONFIRMATION is the confirmation mode because we don't want
// the batch to fail if we can't reset the modem. The user might
// have power cycled the modem when they rebooted their computer
// send a batch to reset the modem now that the user has been
// notified to reboot their computer

performOperation(
 DeviceOperation.RESET,
 //deviceOperation: Reset operation
modemMACaddress,
 // macAddress:Modem's MAC address
null
 // properties : not used
);

```

**Step 7** When the computer is rebooted, it receives a new IP address. The cable modem is now a provisioned device. The computer is not registered with BAC, but it gains access to the Internet through the service provider's network. Computers that are online behind promiscuous modems are still available using the provisioning API.

# Bulk Provisioning 100 Modems in Promiscuous Mode

A service provider wants to preprovision 100 cable modems for distribution by a customer service representative at a service kiosk.

## Desired Outcome

Use this workflow to distribute modem data for all modems to new subscribers. The customer service representative has a list of modems available for assignment.

- 
- Step 1** The cable modem's MAC address data for new or recycled cable modems is collected into a list at the service provider's loading dock.
  - Step 2** Modems that are assigned to a particular kiosk are bulk-loaded into BAC and are flagged with the identifier for that kiosk.
  - Step 3** When the modems are distributed to new subscribers at the kiosk, the customer service representative enters new service parameters, and changes the Owner ID field on the modem to reflect the new subscriber's account number.

```

// get a single batch for bulk load or 100 modems

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)
// The activation mode for this batch should be NO_ACTIVATION.
// NO_ACTIVATION should be used in this situation because no
// network information exists for the devices because they
// have not booted yet. A configuration can't be generated if no
// network information is present. And because the devices
// have not booted, they are not online and therefore cannot
// be reset. NO_CONFIRMATION is the confirmation mode because
// we are not attempting to reset the devices.
// Create a Map for the properties of the modem
Map properties;

// Set the property ModemKeys.PROMISCUOUS_MODE_ENABLED to
// enable promiscuous mode on modem.
// This could be done at a system level if promiscuous mode
// is your default provisioning mode.
properties.put(ModemKeys.PROMISCUOUS_MODE_ENABLED, "enabled")

// The ModemKeys.CPE_DHCP_CRITERIA is used to specify the DHCP
// Criteria to be used while selecting IP address scopes for
// CPEs behind this modem in the promiscuous mode.

properties.put(ModemKeys.CPE_DHCP_CRITERIA, "provisionedCPE");

// for each modem MAC-address in list:

ModemLoop:
{
 add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 modemMACaddress,
 // macAddress: derived from computer lease
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "Silver",
 // ClassOfService
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 properties
 // properties:
);
}
//end ModemLoop.

```

# Preprovisioning First-Time Activation in Promiscuous Mode

A new subscriber contacts the service provider and requests service. The subscriber has a computer installed in a single dwelling unit.

## Desired Outcome

Use this workflow to bring a new unprovisioned cable modem and computer online with the appropriate level of service.

- 
- Step 1** The service provider chooses a subscriber username and password for the billing system.
  - Step 2** The service provider selects the services that the subscriber can access.
  - Step 3** The service provider registers the device using its own user interface.
  - Step 4** The service provider's user interface passes information, such as the modem's MAC address and the Class of Service to BAC. Additionally, the modem gets a CPE DHCP criteria setting that lets Network Registrar select a provisioned address for any computers to be connected behind the modem. The new modem is then registered with BAC.
  - Step 5** The service provider's field technician installs the physical cable to the new subscriber's house and installs the preprovisioned device, connecting it to the subscriber's computer.

```
// MSO admin UI calls the provisioning API to pre-provision
// an HSD modem.

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// The activation mode for this batch should be NO_ACTIVATION.
// NO_ACTIVATION should be used in this situation because no
// network information exists for the modem because it has not
// booted. A configuration cannot be generated if no network
// information is present. And because the modem has not booted,
// it is not online and therefore cannot be reset.
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the modem.
// Create a map for the properties of the modem.

Map properties;

// Set the property ModemKeys.PROMISCUOUS_MODE_ENABLED to enable
// promiscuous mode on modem

properties.put(ModemKeys.PROMISCUOUS_MODE_ENABLED, "enabled")

properties.put(ModemKeys.CPE_DHCP_CRITERIA, "provisionedCPE");

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 "1,6,00:11:22:33:44:55",
 // macAddress: derived from computer lease
 null,
```

```

 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "Silver",
 // ClassOfService
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 properties
 // properties:
);

```

**Step 6** The technician powers on the cable modem and BAC gives it provisioned access.

**Step 7** The technician powers on the computer and BAC gives it provisioned access. The cable modem and the computer are now both provisioned devices. The computer has access to the Internet through the service provider's network.

## Replacing an Existing Modem

A service provider wants to replace a broken modem.



### Note

If the computer has the option restricting roaming from one modem to another, and the modem is replaced, the computer's MAC address for the modem must also be changed.

### Desired Outcome

Use this workflow to physically replace an existing cable modem with a new modem without changing the level of service provided to the subscriber.

**Step 1** The service provider changes the MAC address of the existing modem to that of the new modem.

```

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ ACTIVATION is the activation mode because we will
// not be able to reset as the new modem has not booted
// on the network.
// NO_CONFIRMATION is the confirmation mode because we are
// not trying to reset the modem
// To change the MAC address of a DOCSIS modem:

changeMACAddress (
 "1,6,00:11:22:33:44:55"
 // old macAddress: unique identifier for the old modem

```



```

"1,6,11:22:33:44:55:66"
 /// new macAddress: unique identifier for the new modem
);

```

- Step 2** The service provider replaces the cable modem and turns it on. The computer must also be turned on.
- Step 3** The cable modem is now a fully provisioned device with the appropriate level of service, as is the computer behind the cable modem.
- 

## Adding a Second Computer in Promiscuous Mode

A subscriber wishes to connect a second computer behind an installed cable modem.

### Desired Outcome

Use this workflow to ensure that the subscriber's selected service permits the connection of multiple CPE, and that the subscriber has network access from both connected computers.



**Note** This case does not require calls to the provisioning API.

---

- Step 1** The subscriber connects a second computer behind the cable modem.
- Step 2** The subscriber turns on the computer.
- Step 3** If the subscriber's selected service permits connecting multiple CPE, BAC gives the second computer access to the Internet.
- 

## Self-Provisioning First-Time Activation with NAT

A university has purchased a DOCSIS cable modem with network address translation (NAT) and DHCP capability. The five occupants of the unit each have a computer installed with a browser application.

### Desired Outcome

Use this workflow to bring a new unprovisioned cable modem (with NAT) and the computers behind it online with the appropriate level of service.

---

- Step 1** The subscriber purchases a cable modem with NAT and DHCP capability and installs it in a multiple dwelling unit.
- Step 2** The subscriber turns on the modem and BAC gives it restricted access.
- Step 3** The subscriber connects a laptop computer to the cable modem, and the DHCP server in the modem provides an IP address to the laptop.
- Step 4** The subscriber starts a browser application on the computer and a spoofing DNS server points the browser to the service provider's registration server (for example, an OSS user interface or a mediator).

- Step 5** The subscriber uses the service provider's user interface to complete the steps required for cable modem registration of the modem. The registration user interface detects that the modem is using NAT and registers the modem, making sure that the modem gets a Class of Service that is compatible with NAT.

```

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a query.
// NO_CONFIRMATION is the confirmation mode because we are not
// attempting to reset the device.
// First we query the computer's information to find the modems
// MAC address.

// With NAT, the computer is never seen by the Network Registrar
// DHCP server. Instead, the modem has translated the IP address
// it assigned to the computer, so the web browser sees the
// modem's IP address. When the lease data is examined, the MAC
// address for the device and the relay-agent-remote-id are the
// same, indicating that this is an unprovisioned modem device,
// which is therefore presumed to be performing NAT.

LeaseResults modemLease =
 getAllForIPAddress("10.0.14.38");
 // ipAddress: restricted access.

String modemMACAddress = modemLease.
 getSingleLease().get(DeviceDetailsKeys.MAC_ADDRESS);
// MSO client registration program then calls the provisioning
// API to provision the NAT modem (with an appropriate class of
// service for NAT).

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION will generate a configuration for the modem
// however it will not attempt to reset it.
// The configuration will be able to be generated because the
// modem has booted. NO_CONFIRMATION is the confirmation mode
// because we are not attempting to reset the modem

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 modemMACAddress,
 // macAddress: derived from its lease
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "Silver",
 // ClassOfService
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address

```

```

 null
 // properties: not used
 };

```

**Step 6** The user interface prompts the subscriber to reboot the computer.

**Step 7** The provisioning client calls `performOperation()` to reboot the modem and gives the modem provisioned access.

```

get-new-batch(AUTOMATIC, NO_CONFIRMATION)

// AUTOMATIC is the activation mode because we are attempting
// to reset the modem so that it receives its new class of service.
// NO_CONFIRMATION is the confirmation mode because we don't want
// the batch to fail if we can't reset the modem. The user might
// have power cycled the modem when they rebooted their computer
// send a batch to reset the modem now that the user has been
// notified to reboot their computer

performOperation(
 DeviceOperation.RESET,
 //deviceOperation: Reset operation
 "1,6,00:11:22:33:44:55",
 // macAddress:Modem's MAC address
 null
 // properties : not used
);

```

**Step 8** The cable modem is now fully provisioned and the computers behind it have full access to the service provider's network.



**Note**

Certain cable modems with NAT may require you to reboot the computer to get the new Class of Service settings. If the cable modem and NAT device are separate devices, the NAT device must also be registered similarly to registering a computer.

## Adding a New Computer Behind a Modem with NAT

The landlord of an apartment building has four tenants sharing a modem and accessing the service provider's network. The landlord wants to provide Internet access to a new tenant, sharing the building's modem. The modem has NAT and DHCP capability. The new tenant has a computer connected to the modem.

### Desired Outcome

Use this workflow to bring a new unprovisioned computer online with a previously provisioned cable modem so that the new computer has the appropriate level of service.

**Note**


---

This case does not require calls to the provisioning API.

---

**Step 1** The subscriber turns on the computer.

**Step 2** The computer is now a provisioned device with access to the appropriate level of service.

**Note**


---

The provisioned NAT modem hides the computers behind it from the network.

---

## Move Device to Another DHCP Scope

A service provider is renumbering its network causing a registered cable modem to require an IP address from a different Network Registrar scope.

### Desired Outcome

A provisioning client changes the DHCP criteria, and the cable modem receives an IP address from the corresponding DHCP scope.

---

**Step 1** Change the DOCSIS modem's DHCP criteria to "newmodemCriteria".

```
get-new-batch(AUTOMATIC, NO_CONFIRMATION);
// AUTOMATIC is the Activation mode because we are attempting
// to reset the modem so that a phone line is disabled
// NO_CONFIRMATION is the Confirmation mode because we don't
// want the batch to fail if we can't reset the modem.

// This use case assumes that the DOCSIS modem has been
// previously added to the database

changeDHCPCriteria(
 "1,6,ff:00:ee:11:dd:22"
 // Modem's MAC address or FQDN
 "newmodemCriteria"
);
```

**Step 2** The modem gets an IP address from the scope targeted by "newmodemCriteria."

---

# Log Device Deletions Using Events

A service provider has multiple provisioning clients and wants to log device deletions.

## Desired Outcome

When any provisioning client deletes a device, the provisioning client logs an event in one place.

- 
- Step 1** Create a listener for the device deletion event. This class must extend the DeviceAdapter abstract class or, alternatively, implement the DeviceListener interface. This class must also override the `deletedDevice(DeviceEvent ev)` method in order to log the event.

```
public DeviceDeletionLogger
 extends DeviceAdapter
 //Extend the DeviceAdapter class.
{
 public void deletedDevice(DeviceEvent ev)
 //Override deletedDevice.
 {
 logDeviceDeletion(ev.getDeviceID());
 //Log the deletion.
 }
}
```

- Step 2** Register the listener and the qualifier for the events using the PACEConnection interface.

```
DeviceDeletionLogger deviceDeletionLogger =
 new DeviceDeletionLogger();
 // Modem's MAC address or FQDN
 "newmodemCriteria"
qualifier = new DeviceEventQualifier();
// We are interested only in device deletion.
qualifier.setDeletedDevice ();
// Add device listener using PACEConnection
connection.addDeviceListener(deviceDeletionLogger, qualifier
);
```

- Step 3** When a device is deleted from the system, the event is generated, and the listener is notified.
-

# Monitoring an RDU Connection Using Events

A service provider is running a single provisioning client and wants notification if the connection between the provisioning client and the RDU breaks.

## Desired Outcome

Use this workflow to have the event interface notify the service provider if the connection breaks.

- 
- Step 1** Create a listener for the messaging event. This class must extend the `MessagingAdapter` abstract class or, alternatively, implement the `MessagingListener` interface. This class must override the `connectionStopped(MessagingEvent ev)` method.

```
// Extend the service provider's Java program using the
// provisioning client to receive Messaging events.
public MessagingNotifier
 extends MessagingAdapter
 //Extend the MessagingAdapter class.
{
 public void connectionStopped(MessagingEvent ev)
 //Override connectionStopped.
 {
 doNotification(ev.getAddress(), ev.getPort());
 //Do the notification.
 }
}
```

- Step 2** Register the listener and the qualifier for the events using the `PACEConnection` interface.

```
MessagingQualifier qualifier =
 new MessagingQualifier();
qualifier.setAllPersistentConnectionsDown();
MessagingNotifier messagingNotifier = new MessagingNotifier();
connection.addMessagingListener(messagingNotifier, qualifier
);
```

- Step 3** If a connection breaks, the event is generated, and the listener is notified.
-

# Logging Batch Completions Using Events

A service provider has multiple provisioning clients and wants to log batch completions.

## Desired Outcome

When any provisioning client completes a batch, an event is logged in one place.

- 
- Step 1** Create a listener for the event. This class must extend the `BatchAdapter` abstract class or implement the `BatchListener` interface. This class must override the `completion(BatchEvent ev)` method in order to log the event.

```
public BatchCompletionLogger
 extends BatchAdapter
 //Extend the BatchAdapterclass.
{
 public void completion(BatchEvent ev)
 //Override completion.
 {
 logBatchCompletion(ev.BatchStatus().getBatchID());
 //Log the completion.
 }
}
```

- Step 2** Register the listener and the qualifier for the events using the `PACEConnection` interface.

```
BatchCompletionLogger batchCompletionLogger =
 new BatchCompletionLogger();
Qualify All qualifier = new Qualify All();
connection.addBatchListener(batchCompletionLogger , qualifier
);
```

- Step 3** When a batch completes, the event is generated, and the listener is notified.
- 

# Getting Detailed Device Information

A service provider wants to allow an administrator to view detailed information for a particular device.

## Desired Outcome

The service provider's administrative application displays all known details about a given device, including MAC address, lease information, provisioned status of the device, and the device type (if known).

---

**Step 1** The administrator enters the MAC address for the device being queried into the service provider's administrative user interface.

**Step 2** BAC queries the embedded database for the device details.

```
get-new-batch(AUTOMATIC, NO_CONFIRMATION);

// MSO admin UI calls the provisioning API to query the details
// for the requested device. Query may be performed based on MAC
// address or IP address, depending on what is known about the
// device.
Map deviceDetails =
 getDetails(
 "1,6,00:11:22:33:44:55",
 // macORFqdn: unique identifier for the device
 true
 // needLeaseInfo: yes we need it
);
```

**Step 3** The service provider's application presents a page of device data details, which can display everything that is known about the requested device. If the device was connected to the service provider's network, this data includes lease information (for example, IP address and relay agent identifier). The data indicates whether the device was provisioned, and if it was, the data also includes the device type.

```
// extract device detail data from the map
String deviceType = (String)deviceDetails.get(DEVICE_TYPE);
String macAddress = (String)deviceDetails.get(MAC_ADDRESS);
String ipAddress = (String)deviceDetails.get(IP_ADDRESS);
String relayAgentID = (String)deviceDetails.get(RELAY_AGENT_ID);
Boolean isProvisioned = (Boolean)deviceDetails.get(IS_PROVISIONED);
// The admin UI now formats and prints the detail data to a view page
```

---

## Searching Using the Default Class of Service

A service provider wants to allow an administrator to view data for all modems with the **default** Class of Service for DOCSIS device type.

### Desired Outcome

The service provider's administrative application returns a list of DOCSIS devices with the default Class of Service.

---

**Step 1** The administrator selects the search option in service provider's administrative user interface.

**Step 2** BAC queries the embedded database for a list of all MAC addresses for the devices that match the requested default Class of Service.



```

get-new-batch(AUTOMATIC, NO_CONFIRMATION);

// Create a MACAddressSearchType to indicate search by
// default class of service.

MACAddressSearchType mst =
 new MACAddressSearchType.getByDefaultClassOfService(
 DeviceType.DOCSIS);

// Create a MACAddressSearchFilter to get 20 devices
// at a time. This indicates that we have a page size of 20.

MACAddressSearchFilter mySearchFilter =
 new MACAddressSearchFilter(
 mst,
 // type: MAC address search type
 false,
 // isInclusive:
 20
 // maximumReturned:
);

// MAC address to start the search from.
String startMac = null;

// A list containing the MAC addresses returned from
// search.
List deviceList = null;

// If the size of deviceList is equal to 20 then
// there may be devices matching the search criteria.
while ((deviceList == null) ||
 (deviceList.size() == 20))
{
 // Use the provisioning API call to search BACC
 // database. The search starts from the MAC address
 // "startMac". If "startMac" is null then the search
 // starts from the very beginning of the index.

 deviceList = search (mySearchFilter, startMac);

 // See Step 3 for the definition of processMACAddressList

 startMac = processMACAddressList(deviceList);
}

```

- Step 3** The service provider's application requests details on these devices from BAC, and presents a page of device data. For each device, the code provides for display of the device type, MAC address, client class, and provisioned status of the device, one device per line.

```

processMACAddressList (List deviceList)
{
 Iterator iter = deviceList.iterator();

 String startMac = null;

 while (iter.hasNext())
 {
 startMac = (String) iter.next();
 // Get details for this device
 Map detailMap = getDetails (
 startMac,
 // MAC of current device
 true
 // yes, we need lease info
);

 // extract device detail data from each map
 // this can be used for displaying in UI
 String deviceType = (String)detailMap.get (DEVICE_TYPE);
 String macAddress = (String)detailMap.get (MAC_ADDRESS);
 String clientClass = (String)detailMap.get (CLIENT_CLASS);
 Boolean isProvisioned = (Boolean)detailMap.get (IS_PROVISIONED);
 // format and print above data in output line
 }

 // We return the last MAC address in the list
 // so that the next search can be started from here

 return startMAC;
}

```

## Retrieving Devices Matching a Vendor Prefix

A service provider wants to allow an administrator to view data for all devices matching a particular vendor prefix.

### Desired Outcome

The service provider's administrative application returns a list of devices matching the requested vendor prefix.

- 
- Step 1** The administrator enters the substring matching the desired vendor prefix into the service provider's administrator user interface.
  - Step 2** BAC queries the embedded database for a list of all MAC addresses for the devices that match the requested vendor prefix.

```
// Create a MACAddressPattern corresponding to the requested
// vendor prefix

MACAddressPattern pattern =
 new MACAddressPattern(
 "1,6,ff:00:ee:*",
 // macAddressPattern: the requested vendor prefix
);

// Create a MACAddressSearchType to indicate search by
// MAC address pattern

MACAddressSearchType mst =
 new MACAddressSearchType.getDevices(pattern);

// Create a MACAddressSearchFilter to get 20 devices
// at a time. This indicates that we have a page size of 20.

MACAddressSearchFilter mySearchFilter =
 new MACAddressSearchFilter(
 mst,
 // type: MAC address search type
 false,
 // isInclusive:
 20
 // maximumReturned:
);

// MAC address to start the search from.
String startMac = null;

// A list containing the MAC addresses returned from
// search.
List deviceList = null;

// If the size of deviceList is equal to 20 then
// there may be devices matching the search criteria.

while ((deviceList == null) ||

 (deviceList.size() == 20))
{

 // Use the provisioning API call to search BACC
 // database. The search starts from the MAC address
 // "startMac". If "startMac" is null then the search
 // starts from the very beginning of the index.

 deviceList = search(mySearchFilter, startMac);

 // See Step 3 for the definition of processMACAddressList

 startMac = processMACAddressList(deviceList);
}
```

- Step 3** The service provider's application requests details on these devices from BAC, and presents a page of device data. For each device, the code displays the device type, MAC address, client class, and provisioned status of the device. One device is identified per line.

```
processMACAddressList (List deviceList)
{
 Iterator iter = deviceList.iterator();

 String startMac = null;

 while (iter.hasNext())
 {
 startMac = (String) iter.next();
 // Get details for this device
 Map detailMap = getDetails (
 startMac,
 // MAC of current device
 true
 // yes, we need lease info
);

 // extract device detail data from each map
 // this can be used for displaying in UI
 String deviceType = (String)detailMap.get (DEVICE_TYPE);
 String macAddress = (String)detailMap.get (MAC_ADDRESS);
 String clientClass = (String)detailMap.get (CLIENT_CLASS);
 Boolean isProvisioned = (Boolean)detailMap.get (IS_PROVISIONED);
 // format and print above data in output line
 }

 // We return the last Mac address in the list
 // so that the next search can be started from here

 return startMac;
}
```

## Preprovisioning PacketCable eMTA

A new customer contacts a service provider to order PacketCable voice service. The customer expects to receive a provisioned embedded MTA.

### Desired Outcome

Use this workflow to preprovision an embedded MTA so that the modem MTA component has the appropriate level of service when brought online.



### Note

This use case skips the call agent provisioning that is required for making telephone calls from eMTAs.

- Step 1** The service provider chooses a subscriber username and password for the billing system.

- Step 2** The service provider chooses the appropriate Class of Service and DHCP criteria for the modem component and adds it to BAC.

```

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// Let's provision the modem and the MTA component in the same
// batch. This can be done because the activation mode of this
// batch is NO_ACTIVATION. More than one device can be operated
// on in a batch if the activation mode does not lead to more
// than one device being reset.
// To add a DOCSIS modem:

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 "1,6,01:02:03:04:05:06",
 // macAddress: scanned from the label
 null
 // hostName: not used in this example
 null
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "Silver",
 // classOfService
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 null
 // properties: not used
);

```

- Step 3** The service provider chooses the appropriate Class of Service and DHCP criteria for the MTA component and adds it to BAC.

```

// Continuation of the batch in Step2
// To add the MTA component:

add(
 DeviceType.PACKET_CABLE_MTA,
 // deviceType: PACKET_CABLE_MTA
 "1,6,01:02:03:04:05:07",
 // macAddress: scanned from the label
 null,
 // hostName: not used in this example, will be auto generated
 null,

```

```

 // domainName: not used in this example, will be auto generated.
 // The FqdnKeys.AUTO_FQDN_DOMAIN property must be set somewhere in the
 // property hierarchy.
"0123-45-6789",
 // ownerID: here, account number from billing system
"Silver",
 // ClassOfService
"provisionedMTA",
 // DHCP Criteria: Network Registrar uses this to
 // select an MTA lease granting provisioned IP address
null
 // properties: not used
);

```

**Step 4** The embedded MTA gets shipped to the customer.

**Step 5** The customer brings the embedded MTA online and makes telephone calls using it.

---

## SNMP Cloning on PacketCable eMTA

A customer has an SNMP Element Manager that wishes to gain access to a PacketCable eMTA.

### Desired Outcome

An external Element Manager is granted secure SNMPv3 access to the PacketCable eMTA.



### Note

Changes made to RW MIB variables are not permanent and are not updated in the BAC configuration for the eMTA. The information written into the eMTA MIB is lost the next time the MTA powers down or resets.

---

**Step 1** Call the provisioning API method, `performOperation()`, passing in the MAC address of the MTA and the username of the new user to create on the MTA. This will be the username used in subsequent SNMP calls by the Element Manager.

```

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION);

// NO_ACTIVATION is the activation mode because we don't want to
// reset the device.
// NO_CONFIRMATION is the confirmation mode because we are
// not attempting to reset the device.

// The goal here is to create a new user on the MTA indicated
// by the MAC address. The other parameter needed here is the new
// user name, which is passed in the Map.
// Create a map that contains one element - the name of
// the new user to be created on the MTA
HashMap map = new HashMap();
map.put(SNMPPPropertyKeys.CLONING_USERNAME, "newUser");
// The first param is the actual device operation to perform.
performOperation(
 DeviceOperation.ENABLE_SNMPV3_ACCESS,
 // deviceOperation : ENABLE_SNMPV3_ACCESS
 "1,6,00:00:00:00:00:99",
 // macORFqdn : MAC Address of the modem
 map
 // parameters: operation specific parameters
);

```

- Step 2** The provisioning API attempts to perform an SNMPv3 cloning operation to create an entry on the MTA for the new user passed in step 1. The keys used in the new user entry row are a function of two passwords defined within BAC. These passwords will be made available to the customer and the RDU command passes these passwords (the auth and priv password) through a key localization algorithm to create an auth and priv key. These are stored, along with the new user, in the eMTA's user table.

**Note**

The auth and priv passwords mentioned in this step may be changed by setting `SNMPPPropertyKeys.CLONING_AUTH_PASSWORD (/snmp/cloning/auth/password)` and `SNMPPPropertyKeys.CLONING_PRIV_PASSWORD (/snmp/cloning/priv/password)` properties respectively in the `rdu.properties` configuration file.

- Step 3** The customer issues SNMPv3 request using above specified username, passwords, and key localization algorithm to allow for secure communication with the MTA.

## Incremental Provisioning of PacketCable eMTA

A customer has a PacketCable eMTA in service with its first line (end point) enabled. The customer wants to enable the second telephone line (end point) on the eMTA and connect a telephone to it.

### Desired Outcome

The customer should be able to connect a telephone to the second line (end point) on the eMTA and successfully make phone calls from it without any service interruption.

**Note**

In order to use the second line on the eMTA, the Call Agent need to be configured accordingly. This use case does not address provisioning of call agents.

**Step 1**

The service provider's application invokes the BAC API to change the Class of Service of the eMTA. The new Class of Service supports two end points on the eMTA. This change in Class of Service does not take effect until the eMTA is reset. Disrupting the eMTA is not desirable; therefore, incremental provisioning is undertaken in the next step.

```
get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because we don't want to
// reset the device.
// NO_CONFIRMATION is the Confirmation mode because we are not
// disrupting the device.

changeClassOfService(
 "1,6,ff:00:ee:11:dd:22" // eMTA's MAC address or FQDN
 ,"twoLineEnabledCOS" // This COS supports two lines.
);
```

**Step 2**

The service provider's application uses the BAC incremental update feature to set SNMP objects on the eMTA and thereby enabling the service without disrupting the eMTA.

```
// The goal here is to enable a second phone line, assuming one
// phone line is currently enabled. We will be adding a new
// row to the pktcNcsEndPntConfigTable.

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because we don't want to
// reset the device.
// NO_CONFIRMATION is the confirmation mode because we are
// not attempting to reset the device.

// Create a map containing one element - the list of SNMP
// variables to set on the MTA
HashMap map = new HashMap();

// Create an SnmpVarList to hold SNMP varbinds
SnmpVarList list = new SnmpVarList();

// An SnmpVariable represents an oid/value/type triple.

// pktcNcsEndPntConfigTable is indexed by the IfNumber, which in this
// case we will assume is interface number 12 (this is the last
// number in each of the oids below.

// The first variable represents the creation of a new row in
// pktcNcsEndPntConfigTable we are setting the RowStatus
// column (column number 26). The value of 4 indicates that
// a new row is to be created in the active state.
SnmpVariable variable = new SnmpVariable(
```



```

 ".1.3.6.1.4.1.4491.2.2.2.1.2.1.1.26.12",
 "4",
 SnmpType.INTEGER);
list.add(variable);

// The next variable represents the call agent id for this new
// interface, which we'll assume is 'test.com'
SnmpVariable variable = new SnmpVariable(
 ".1.3.6.1.4.1.4491.2.2.2.1.2.1.1.1.12",
 "test.com",
 SnmpType.STRING);
list.add(variable);

// The final variable represents the call agent port
SnmpVariable variable = new SnmpVariable(
 ".1.3.6.1.4.1.4491.2.2.2.1.2.1.1.2.12",
 "2728",
 SnmpType.INTEGER);
list.add(variable);

// Add the SNMP variable list to the Map to use in the API call
map.put(SNMPPropertyKeys.SNMPVAR_LIST, list);

// Invoke the BACC API to do incremental update on the eMTA.
performOperation(DeviceOperation.INCREMENTAL_UPDATE // device operation
 , "1,6,00:00:00:00:00:99" // MAC Address
 , map // Parameters for the operation
);

```

- Step 3** The eMTA is enabled to use the second telephone line. The eMTA continues to receive the same service, after being reset, since the Class of Service was changed in step 1.
- 

## Preprovisioning DOCSIS Modems with Dynamic Configuration Files

A new customer contacts a service provider to order a DOCSIS modem with high-speed *Gold* data service for two CPE behind it.

### Desired Outcome

Use this workflow to preprovision a DOCSIS modem with a Class of Service that uses DOCSIS templates. The dynamic configuration file generated from the templates is used while the modem comes online.

- Step 1** The service provider chooses a subscriber username and password for the billing system.
- Step 2** The service provider chooses Gold Class of Service, and the appropriate DHCP criteria, and then adds the cable modem to BAC.

```

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

Map properties;

// Set the property ModemKeys.PROMISCUOUS_MODE_ENABLED to enable
// promiscuous mode on modem

properties.put(ModemKeys.PROMISCUOUS_MODE_ENABLED, "enabled")

// No CPE DHCP Criteria is specified.
// The CPEs behind the modem will use the default provisioned
// promiscuous CPE DHCP criteria specified in the system defaults.

// This custom property corresponds to a macro variable in the
// DOCSIS template for "gold" class of service indicating the
// maximum number of CPEs allowed behind this modem. We set it
// to two CPEs from this customer.

properties.put("docsis-max-cpes", "2");

// To add a DOCSIS modem:

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 "1,6,01:02:03:04:05:06",
 // macAddress: scanned from the label
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "gold",
 // classOfService:
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 properties
 // properties:
);

```

**Step 3** The cable modem is shipped to the customer.

**Step 4** The customer brings the cable modem online and connects the computers behind it.

---

# Optimistic Locking

An instance of the service provider application needs to ensure that it is not overwriting the changes made by another instance of the same application.

## Desired Outcome

Use this workflow to demonstrate the optimistic locking capabilities provided by the BAC API.



### Note

Locking of objects is done in multi-user systems to preserve integrity of changes, so that one person's changes do not accidentally get overwritten by another. With optimistic locking, you write your program assuming that any commit has a chance to fail if at least one of the objects being committed was changed by someone else since you began the transaction.

**Step 1** The service representative selects the search option in the service provider's user interface and enters the cable modem's MAC address.

**Step 2** BAC queries the embedded database, gets the details of the device and the MSO user interface displays the information.

```
// MSO admin UI calls the provisioning API to query the details
// for the requested device.
Map deviceDetails =
 getDetails(
 "1,6,00:11:22:33:44:55",
 // macORFqdn: unique identifier for the device
 true
 // needLeaseInfo: yes we need it
);

// extract device detail data from the map
String deviceType = (String)deviceDetails.get(DEVICE_TYPE);
String macAddress = (String)deviceDetails.get(MAC_ADDRESS);
String ipAddress = (String)deviceDetails.get(IP_ADDRESS);
String relayAgentID = (String)deviceDetails.get(RELAY_AGENT_ID);

Boolean isProvisioned = (Boolean)deviceDetails.get(IS_PROVISIONED);
// service provider admin UI displays this information.

// Let's save the OID_REVISION_NUMBER property so that we can see in
// step 3.
String oidRevisionNumber = (String)deviceDetails.get(OID_REVISION_NUMBER);
```

**Step 3** The service representative attempts to change the Class of Service and the DHCP criteria of the modem using the user interface. This in turn invokes the BAC API.

```

// We need a reference to Batch instance so that ensureConsistency()
// method can be invoked on it.
Batch batch = conn.newBatch() ;

List oidList = new ArrayList();
// Add the oid-rev number saved from step 2 to the list
oidList.add(oidRevisionNumber);

// Sends a list of OID revision numbers to validate before processing the
// batch. This ensures that the objects specified have not been modified
// since they were last retrieved.
batch.ensureConsistency(oidList);

 batch.changeClassOfService (
 "1,6,00:11:22:33:44:55",
 // macORFqdn: unique identifier for the device.
 "gold"
 // newCOSName : Class of service name.
)

batch.changeDHCPCriteria (
 "1,6,00:11:22:33:44:55",
 // macORFqdn: unique identifier for the device.
 "specialDHCPCriteria"
 // newDHCPCriteria : New DHCP Criteria.
)

// This batch fails with BatchStatusCodes.BATCH_NOT_CONSISTENT,
// in case if the device is updated by another client in the mean time.
// If there is a conflict occurs, then the service provider client
// is responsible for resolving the conflict by querying the database
// again and then applying changes appropriately.

```

**Step 4** The user is ready to receive Gold Class of Service with appropriate DHCP criteria.

---

## Temporarily Throttling a Subscriber's Bandwidth

An MSO has a service that allows a subscriber to download only 10 MB of data a month. Once the subscriber reaches that limit, their downstream bandwidth is turned down from 10 MB to 56 K. When the month is over they are moved back up to 10 MB.



### Note

You may want to consider changing upstream bandwidth as well, since peer-to-peer users and users who run websites tend to have heavy upload bandwidth.

---

### Desired Outcome

Use this workflow to move subscribers up and down in bandwidth according to their terms of agreement.

---

**Step 1** The MSO has a rate tracking system, such as NetFlow, which keeps track of each customer's usage by MAC address. Initially a customer is provisioned at the Gold Class of Service level with 1 MB downstream.

- Step 2** When the rate tracking software determines that a subscriber has reached the 10-MB limit it notifies the OSS. The OSS then makes a call into the BAC API to change the subscriber's Class of Service from Gold to Gold-throttled.

```
get-new-batch(AUTOMATIC, NO_CONFIRMATION)

// AUTOMATIC is the activation mode because we are
// attempting to reset the modem so that it
// receives low bandwidth service.
// NO_CONFIRMATION is the confirmation mode
// because we do not want the batch to fail if we cannot
// reset the modem. If the modem is off, when it will
// be disabled when it is turned back on.

// Let's change the COS of the device so that it restricts
// bandwidth usage of the modem.
 changeClassOfService(
 Device.MAC_ADDRESS,
 // macAddress: unique identifier for this modem
 "Gold-throttled");
 // newClassOfService: restricts bandwidth usage to 56k
```

- Step 3** At the end of the billing period, the OSS calls the BAC API to change the subscriber's Class of Service back to *Gold*.

## Preprovisioning CableHome WAN-MAN

A new customer contacts a service provider to order home networking service. The customer expects a provisioned CableHome device to be shipped to him.

### Desired Outcome

Use this workflow to preprovision a CableHome device so that the cable modem and WAN-MAN components on it will have appropriate level of service when brought online.

- Step 1** The service provider chooses a subscriber username and password for the billing system.
- Step 2** The service provider chooses the appropriate Class of Service and the DHCP criteria for the modem component, then adds it to BAC.

```
get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// Let's provision the modem and the WAN-Man component in the same
// batch.
// To add a DOCSIS modem:

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
```

```

"1,6,01:02:03:04:05:06",
 // macAddress: scanned from the label
null,
 // hostName: not used in this example
null,
 // domainName: not used in this example
"0123-45-6789",
 // ownerID: here, account number from billing system
"Silver",
 // classOfService
"provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
null
 // properties: not used
);

```

**Step 3** The service provider chooses the appropriate Class of Service and DHCP criteria for the WAN-MAN component, then adds it to BAC.

```

// Continuation of the batch in Step2
// To add the WAN-Man component:
add(
 DeviceType.CABLEHOME_WAN_MAN,
 // deviceType: CABLEHOME_WAN_MAN
"1,6,01:02:03:04:05:07",
 // macAddress: scanned from the label
null,
 // hostName: not used in this example.
null,
 // domainName: not used in this example.
"0123-45-6789",
 // ownerID: here, account number from billing system
"silverWanMan",
 // ClassOfService
"provisionedWanMan",
 // DHCP Criteria: Network Registrar uses this to
 // select an MTA lease granting provisioned IP address
null
 // properties: not used
);

```

- Step 4** The CableHome device gets shipped to the customer.
- Step 5** The customer brings the CableHome device online.
- 

## CableHome with Firewall Configuration

A customer contacts a service provider to order a home networking service with the firewall feature enabled. The customer expects to receive a provisioned CableHome device.

### Desired Outcome

Use this workflow to preprovision a CableHome device so that the cable modem and the WAN-MAN components on it, have the appropriate level of service when brought online.

---

- Step 1** The service provider chooses a subscriber username and password for the billing system.
- Step 2** The service provider chooses the appropriate Class of Service and DHCP criteria for the cable modem component, then adds it to BAC.

```
get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// Let's provision the modem and the WAN-Man component in the same
// batch.
// To add a DOCSIS modem:

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 "1,6,01:02:03:04:05:06",
 // macAddress: scanned from the label
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "Silver",
 // classOfService
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 null
 // properties: not used
);
```

- Step 3** The service provider chooses the appropriate Class of Service and DHCP criteria for the WAN-MAN component and adds it to BAC.

```

// Continuation of the batch in Step2
// To add the WAN-Man component:

// Create a Map to contain WanMan's properties

Map properties;

// The fire wall configuration for the Wan Man component is specified
// using the CableHomeKeys.CABLEHOME_WAN_MAN_FIREWALL_FILE property.
// This use case assumes that the firewall configuration file named
// "firewall_file.cfg" is already present in the RDU database and the
// firewall configuration is enabled in the Wan Man configuration file
// specified with the corresponding class of service.

properties.put(CableHomeKeys.CABLEHOME_WAN_MAN_FIREWALL_FILE,
"firewall_file.cfg");

add(
 DeviceType.CABLEHOME_WAN_MAN,
 // deviceType: CABLEHOME_WAN_MAN
 "1,6,01:02:03:04:05:07",
 // macAddress: scanned from the label
 null,
 // hostName: not used in this example.
 null,
 // domainName: not used in this example.
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "silverWanMan",
 // ClassOfService
 "provisionedWanMan",
 // DHCP Criteria: Network Registrar uses this to
 // select an MTA lease granting provisioned IP address
 properties
 // properties: contains the firewall config file
);

```

**Step 4** The CableHome device gets shipped to the customer.

**Step 5** The customer brings the CableHome device online and the cable modem and the WAN-MAN component get provisioned IP addresses and proper configuration files.

---



# Retrieving Device Capabilities for CableHome WAN-MAN

A service provider wants to allow an administrator to view capabilities information for a CableHome WAN-MAN device.

## Desired Outcome

The service provider's administrative application displays all known details about a given CableHome WAN-MAN component, including MAC address, lease information, provisioned status, and the device capabilities information.

- 
- Step 1** The administrator enters the MAC address of the WAN-MAN being queried into the service provider's user interface.
- Step 2** BAC queries the embedded database for details of the device identified using the MAC address entered.

```
get-new-batch(NO_ACTIVATION, NO_CONFIRMATION);

// MSO admin UI calls the provisioning API to query the details
// for the requested device.

Map deviceDetails =
 getDetails(
 1,6,00:11:22:33:44:55", "
 // macORFqdn: unique identifier for the device
 true
 // needLeaseInfo: yes we need it
);
```

- Step 3** The service provider's application then presents a page of device data details, which can display everything that is known about the requested device. If the device was connected to the service provider's network, this data includes lease information, such as the IP address or the relay agent identifier. This data indicates whether the device is provisioned. If it is provisioned, the data also includes the device type and device capabilities information.

```
// extract device details information from the map
String deviceType = (String) deviceDetails.get(DeviceDetailsKeys.DEVICE_TYPE);
String macAddress = (String) deviceDetails.get(DeviceDetailsKeys.MAC_ADDRESS);
String ipAddress = (String) deviceDetails.get(DeviceDetailsKeys.IP_ADDRESS);
String relayAgentID = (String) deviceDetails.get(
 DeviceDetailsKeys.RELAY_AGENT_ID);
Boolean isProvisioned = (Boolean) deviceDetails.get(
 DeviceDetailsKeys.IS_PROVISIONED);
String formation = (String) deviceDetails.get(
 IPDeviceCapabilities.FORMATION);
String deviceList = (String) deviceDetails.get(
 IPDeviceCapabilities.DEVICE_LIST);
String serNum = (String) deviceDetails.get(
 IPDeviceCapabilities.SERIAL_NUMBER);
String hwVer = (String) deviceDetails.get(
 IPDeviceCapabilities.HARDWARE_VERSION);
String swVer = (String) deviceDetails.get(
 IPDeviceCapabilities.SOFTWARE_VERSION);
String brVer = (String) deviceDetails.get(
 IPDeviceCapabilities.BOOT_ROM_VERSION);
String vendorOui = (String) deviceDetails.get(
 IPDeviceCapabilities.VENDOR_OUI);
String modelNum = (String) deviceDetails.get(
 IPDeviceCapabilities.MODEL_NUMBER);
String vendorNum = (String) deviceDetails.get(
 IPDeviceCapabilities.VENDOR_NAME);
String sysDesc = (String) deviceDetails.get(
 IPDeviceCapabilities.SYSTEM_DESCRIPTION);
String fwVer = (String) deviceDetails.get(
 IPDeviceCapabilities.FIRMWARE_VERSION);
String fwVer = (String) deviceDetails.get(
 IPDeviceCapabilities.FIREWALL_VERSION);
// The admin UI now formats and prints the detail data to a view page
```


---

# Self-Provisioning CableHome WAN-MAN

A subscriber has a computer with a browser application installed in a single dwelling unit and has purchased an embedded CableHome device.

## Desired Outcome

Use this workflow to bring a new unprovisioned embedded CableHome device online with the appropriate level of service, and give the subscriber Internet access from computers connected to the embedded CableHome device.

- 
- Step 1** The subscriber purchases an embedded CableHome device and installs it at home.
- Step 2** The subscriber powers on the embedded CableHome device. BAC gives the embedded cable modem restricted access, allowing two CPE: one for the CableHome WAN-MAN and the other for the computer.
- 
-  **Note** This use case assumes an unprovisioned DOCSIS modem allows two CPE behind it. Until configured to do otherwise, BAC supports only a single device behind an unprovisioned DOCSIS modem. You can change this behavior by defining an appropriate Class of Service that supports two CPE and then using it as the default Class of Service for DOCSIS devices.
- 
- Step 3** BAC configures the CableHome WAN-MAN, including IP connectivity and downloading the default CableHome boot file. The default CableHome boot file configures the CableHome device in passthrough mode. The CableHome device is still unprovisioned.
- Step 4** The subscriber connects the computer to the CableHome device. The computer gets an unprovisioned (restricted) IP address. The subscriber starts a browser application on the computer. A spoofing DNS server points the browser to the service provider's registration server (for example, an OSS user interface or a mediator).
- Step 5** The subscriber uses the service provider's user interface to complete the steps required for cable modem registration, including selecting a Class of Service. The subscriber also selects a CableHome Class of Service.
- Step 6** The service provider's user interface passes the subscriber's information to BAC, including the selected Class of Service for cable modem and CableHome, and computer IP address. The subscriber is then registered with BAC.

```
get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// NO_ACTIVATION is the activation mode because this is a
// query. NO_CONFIRMATION is the confirmation mode because
// we are not attempting to reset the device.
// First we query the computer's information to find the
// modems MAC address.
// We use the computers IP address (the web browser
// received this when the subscriber opened the service
// providers web interface). We also assume that "bostonProvGroup"
// is the provisioning group used in that locality.

List provGroupList;
provGroupList = provGroupList.add("bostonProvGroup");
Map computerLease = getAllForIPAddress(
 "10.0.14.38",
 // ipAddress: restricted access computer lease
```

```

 provGroupList
 // provGroups: List containing provgroup)
// Derive the modem MAC address from the computer's network
// information. The 1,6, is a standard prefix for an Ethernet
// device. The fully qualified MAC address is required by BPR

String modemMACAddress = "1,6," +

 computerLease.getSingleLease().get(RELAY_AGENT_REMOTE_ID);

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

// Now let's provision the modem
// NO_ACTIVATION will generate a configuration for the
// modem however it will not attempt to reset it
// The configuration will be able to be generated because
// the modem has booted.
// NO_CONFIRMATION is the confirmation mode because we
// are not attempting to reset the modem
// Create a Map for the properties of the modem

Map properties;
// Set the property ModemKeys.PROMISCUOUS_MODE_ENABLED
// to enable promiscuous mode on modem

properties.put (ModemKeys.PROMISCUOUS_MODE_ENABLED, "enabled");

properties.put (ModemKeys.CPE_DHCP_CRITERIA, "provisionedCPE");

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 modemMACAddress,
 // macAddress: derived from computer lease
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
 "Silver",
 // ClassOfService
 "provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
 properties
 // properties:
);

```

**Step 7** The user interface prompts the subscriber to reboot the computer.

**Step 8** The provisioning client calls `performOperation()` to reboot the modem and gives the modem provisioned access.

```

get-new-batch(AUTOMATIC, NO_CONFIRMATION)

// AUTOMATIC is the activation mode because we are attempting
// to reset the modem so that it receives its new class of service
// NO_CONFIRMATION is the confirmation mode because we don't want
// the batch to fail if we can't reset the modem. The user might
// have power cycled the modem when they rebooted their computer
// send a batch to reset the modem now that the user has been
// notified to reboot their computer

performOperation(
 DeviceOperation.RESET,
 //deviceOperation: Reset operation
 modemMACAddress,
 // macAddress:Modem's MAC address
 null
 // properties : not used
);

```

**Step 9** When the computer is rebooted, it receives a new IP address from the CableHome device's DHCP server. The cable modem and the CableHome device are now both provisioned. Now the subscriber can connect a number of computers to the Ethernet ports of the CableHome device and they have access to the Internet.

**Note**

If the configuration file supplied to the WAN-MAN component enables the WAN-Data component on the box, it will get provisioned in the promiscuous mode. This assumes that the promiscuous mode is enabled at the technology defaults level for the DeviceType.CABLEHOME\_WAN\_DATA device type.

## Lease Reservation Use Cases

This section describes use cases specifically related to the use of the lease reservation feature. Standard administrative operations, such as managing Class of Service, DHCP criteria, licenses, and so on are not addressed here. The lease reservation use cases include:

- [Bringing a Device Online Using IP Address Provided by Service Provider, page C-50](#)
- [Removing and Re-Creating a Reservation, page C-51](#)
- [Assigning a New Device with an Old Device's IP Address, page C-52](#)
- [Removing a Reservation and Assigning a New IP Address, page C-53](#)
- [Rebooting a Device with the Same IP Address, page C-54](#)
- [Removing a Device from BAC, page C-55](#)
- [A Submitted Batch Fails when BAC Uses CCM, page C-56](#)
- [A Submitted Batch Fails when BAC Does Not Use CCM, page C-56](#)

## API Calls Affected by the Lease Reservation Feature

The implementation of these API calls supports the lease reservation feature:

- `IPDevice.add(DeviceType DeviceType, String macAddress, String hostName, String domainName, String ownerID, String cosName, String dhcpCriteria, Map Properties)`
- `IPDevice.changeProperties(String macORFqdn, Map newPropToAdd, List propToDelete)`
- `IPDevice.changeMACAddress(String macORFqdn, String newMAC)`
- `IPDevice.delete(String macORFqdn, Boolean deleteDevicesBehind)`

## Bringing a Device Online Using IP Address Provided by Service Provider

When a device is added to the BAC system datastore, the service provider configures that device with the specific IP address for the device using a BAC property (`IPDeviceKeys.IP_RESERVATION`).

### Desired Outcome

Bring a device online with the exact IP address set in the property by the service provider. The granted IP address is reserved (lease reservation) for that device.

- 
- Step 1** The service provider adds a new device to the BAC system. The service provider configures the new device with a specific IP address 10.10.10.1.
- Step 2** The service provider's user interface passes device information to BAC, such as the MAC address, FQDN, and Class of Service. The BAC property (`IPDeviceKeys.IP_RESERVATION`) is used to reserve a specific IP address (10.10.10.1) for this device.

```
Map properties;

// Set the property IPDeviceKeys.IP_RESERVATION to a specific
// IP address

properties.put(IPDeviceKeys.IP_RESERVATION, "10.10.10.1");

// To add a DOCSIS modem:

get-new-batch(NO_ACTIVATION, NO_CONFIRMATION)

add(
 DeviceType.DOCSIS,
 // deviceType: DOCSIS
 "1,6,01:02:03:04:05:06",
 // macAddress: scanned from the label
 null,
 // hostName: not used in this example
 null,
 // domainName: not used in this example
 "0123-45-6789",
 // ownerID: here, account number from billing system
```

```

"gold",
 // classOfService:
"provisionedCM",
 // DHCP Criteria: Network Registrar uses this to
 // select a modem lease granting provisioned IP address
properties
 // properties:
);

```

- Step 3** The new device is then registered with BAC. The reservation of 10.10.10.1 is also created for MAC address “01:02:03:04:05:06” in the BAC/Network Registrar system.
- Step 4** When the device is booted, it receives the exact IP address (10.10.10.1) set in the property by the service provider.

Rollback occurs if needed when the API command results in an error during processing or the change failed to commit to the RDU database. The command implementation removes the lease reservation of 10.10.10.1 for MAC address “01:02:03:04:05:06” from the BAC/Network Registrar if it was made during command processing.

## Removing and Re-Creating a Reservation

After a device is registered to the BAC system with a reserved IP address, the service provider reassigns the device with a different IP address using a BAC property (`IPDeviceKeys.IP_RESERVATION`).

### Desired Outcome

The device is rebooted with the exact IP address set in the property by the service provider. The reservation will be removed and re-created. The previously assigned IP address is unreserved for that device; and the new IP address is granted and reserved (lease reservation) for that device.

- Step 1** A device is registered to BAC with a reserved IP address of 10.10.10.1.
- Step 2** The service provider’s application makes these API calls in BAC to change the reserved IP to 10.10.10.5.

```

get-new-batch(AUTOMATIC, NO_CONFIRMATION);
// AUTOMATIC is the Activation mode because we are attempting
// to reset the device
// NO_CONFIRMATION is the Confirmation mode because we don't
// want the batch to fail if we can't reset the modem.

// This use case assumes that the DOCSIS modem has been
// previously added to the database

```

```

Map properties;

```

```
// Set the property IPDeviceKeys.IP_RESERVATION to a specific
// IP address

properties.put(IPDeviceKeys.IP_RESERVATION, "10.10.10.5");
// To reassign a different IP to:

 changeProperties(
 "1,6,01:02:03:04:05:06",
 // macAdd
 properties, null
);
```

**Step 3** The reservation of 10.10.10.1 for MAC address “01:02:03:04:05:06” is removed from the BAC/Network Registrar system. The reservation of 10.10.10.5 for MAC address “01:02:03:04:05:06” is created in the BAC/Network Registrar system.

**Step 4** When the device is rebooted as the result of device disruption by PACE disruptor (AUTOMATIC is used in the Activation mode), it receives the exact IP address (10.10.10.5) set in the property by the service provider.

Rollback occurs if needed when the API command results in an error during processing or the change failed to commit to the RDU database. The command implementation attempts to roll the device back to the prior working configuration. In this case, reservation of 10.10.10.1 for MAC address “01:02:03:04:05:06” will be re-added to the BAC/Network Registrar system if it was removed; the reservation of 10.10.10.5 for MAC address “01:02:03:04:05:06” will be removed from BAC/Network Registrar system if it was created during command processing.

---

## Assigning a New Device with an Old Device’s IP Address

After a device is registered to the BAC system with a reserved IP address, the service provider needs to replace the broken device with a device with a new MAC address.

### Desired Outcome

The new device is booted with the same IP address granted to the broken old device. The reservation will be removed and recreated, as if the IP address of the device changed.

---

**Step 1** A device is registered to BAC with a reserved IP address of 10.10.10.5.

**Step 2** The service provider changes the MAC address of the existing device (“01:02:03:04:05:06”) to that of the new device (“01:02:03:04:05:07”) in the BAC system.



```

get-new-batch(AUTOMATIC, NO_CONFIRMATION);
// NO_ACTIVATION is the activation mode because we will
// not be able to reset as the new device has not booted
// on the network.
// NO_CONFIRMATION is the confirmation mode because we are
// not trying to reset the device

// To change the MAC address of a device:

changeMACAddress (
 "1,6,01:02:03:04:05:06",
 // old macAddress: unique identifier for the old device
 "1,6,01:02:03:04:05:07"
 //// new macAddress: unique identifier for the new device
);

```

**Step 3** The reservation of 10.10.10.5 for MAC address “01:02:03:04:05:06” is removed from the BAC/Network Registrar system. The reservation of 10.10.10.5 for MAC address “01:02:03:04:05:07” is created in the BAC/Network Registrar system.

**Step 4** When the device with MAC address “01:02:03:04:05:07” is turned on, it receives the same IP address (10.10.10.5) granted to the broken old device.

Rollback occurs if needed when the API command results in an error during processing or the change failed to commit to the RDU database. The command implementation attempts to roll the device back to the prior working configuration. In this case, the reservation of 10.10.10.5 for MAC address “01:02:03:04:05:06” will be re-added to BAC/Network Registrar system if it was removed, the reservation of 10.10.10.5 for MAC address “01:02:03:04:05:07” will be removed from BAC/Network Registrar system if it was created during command processing.

## Removing a Reservation and Assigning a New IP Address

After a device is registered to the BAC system with a reserved IP address, the service provider needs to remove the reservation of the specific IP address since the device no longer needs a reserved IP assignment.

### Desired Outcome

The reservation will be removed from the system and Network Registrar/BAC selects the next available IP address in the appropriate IP pool based on the selected DHCP criteria and assigns it to the device.

**Step 1** A device is registered to BAC with a reserved IP address of 10.10.10.5.

**Step 2** The service provider's application makes these API calls in BAC to remove the reservation.

```
get-new-batch(AUTOMATIC, NO_CONFIRMATION);
// AUTOMATIC is the Activation mode because we are attempting
// to reset the device
// NO_CONFIRMATION is the Confirmation mode because we don't
// want the batch to fail if we can't reset the modem.

// Add the property IPDeviceKeys.IP_RESERVATION the list to be removed

list.add(IPDeviceKeys.IP_RESERVATION);

// To reassign a different IP to:

changeProperties(
 "1,6,01:02:03:04:05:07",
 // macAdd
 null, list
);
```

- Step 3** The reservation of 10.10.10.5 for MAC address "1,6,01:02:03:04:05:07" is removed from the system.
- Step 4** When the device is rebooted as the result of device disruption (AUTOMATIC is used in the Activation mode), dynamic address assignment occurs. Network Registrar/BAC selects the next available IP address in an appropriated IP address pool based on the selected DHCP criteria set on the device and grants it to the device.

Rollback occurs if needed when the API command results in an error during processing or the change failed to commit to the RDU database. The command implementation attempts to roll the device back to the prior working configuration. In this case, the reservation of 10.10.10.5 for MAC address "01:02:03:04:05:07" will be re-added from the BAC/Network Registrar system if it was removed during command processing.

---

## Rebooting a Device with the Same IP Address

After a device is registered to the BAC system with a provisioned IP address (a dynamic IP assignment, not a reserved IP) granted by BAC/Network Registrar, the service provider reassigns the device with a different IP address using a BAC property (`IPDeviceKeys.IP_RESERVATION`).

### Desired Outcome

The device is rebooted with the exact IP address set in the property (`IPDeviceKeys.IP_RESERVATION`) by the service provider. The granted new IP address is now reserved (lease reservation) for that device.

---

- Step 1** A device is registered to BAC with dynamic IP address (Network Registrar/BAC selected an IP address in an appropriated IP pool based on selected DHCP criteria on the device and granted it to the device).
- Step 2** The service provider's application makes these API calls in BAC to reassign the IP address.

```

get-new-batch(AUTOMATIC, NO_CONFIRMATION);
// AUTOMATIC is the Activation mode because we are attempting
// to reset the device
// NO_CONFIRMATION is the Confirmation mode because we don't
// want the batch to fail if we can't reset the modem.

Map properties;

// Set the property IPDeviceKeys.IP_RESERVATION to a specific
// IP address

properties.put(IPDeviceKeys.IP_RESERVATION, "10.10.10.1");

// To reassign a different IP to:

changeProperties(

 "1,6,01:02:03:04:05:08",
 // macAdd
 properties, null

):

```

**Step 3** The reservation of 10.10.10.1 for Mac address "01:02:03:04:05:08" is made with BAC/Network Registrar.

**Step 4** When the device is rebooted as the result of device disruption (AUTOMATIC is used in the Activation mode), it receives the exact IP address (10.10.10.1) set in the property by the service provider.

Rollback occurs if needed when the API command results in an error during processing or the change failed to commit to the RDU database. The command implementation attempts to roll the device back to the prior working configuration. The reservation of 10.10.10.1 for MAC address "01:02:03:04:05:08" will be removed from BAC/Network Registrar system if it was added during command processing. When the device reboots, BAC/Network Registrar selects the next available IP address in the appropriate IP pool based on the selected DHCP criteria and grants it to the device.

## Removing a Device from BAC

A service provider needs to remove the subscriber's device with a reserved IP from the BAC system.

### Desired Outcome

Permanently remove the subscriber's device from the BAC system. The granted IP address is unreserved for that device.

**Step 1** A device is registered to BAC with a reserved IP address of 10.10.10.5.

**Step 2** The service provider uses its own user interface to remove the device from the BAC system. The service provider's user interface, acting as a BAC client, passes the information to BAC. BAC updates the device information and removes the device.

```

delete(
 "1,6,01:02:03:04:05:07",
 // macAdd
 // deleteDevicesBehind res: unique identifier for this device
 true: deletes CPEs behind this modem.
):

```

- Step 3** The reservation of 10.10.10.5 for MAC address “01:02:03:04:05:07” is also removed from BAC/Network Registrar system.

Rollback occurs if needed when the API command results in an error during processing or the change failed to commit to the RDU database. The command implementation attempts to roll the device back to the prior working configuration. The reservation of 10.10.10.5 for MAC address "01:02:03:04:05:07" will be re-added from the BAC/Network Registrar system if it was removed during command processing.

---

## A Submitted Batch Fails when BAC Uses CCM

BAC is configured to use CCM. The OSS submits an API request to add or remove a reservation, but CCM is unavailable (connection down, CCM is incorrectly configured, CCM License issue, and so on).

### Desired Outcome:

When a submitted batch fails, no reservation should be added or removed in BAC/Network Registrar; a predefined, well-known error code is correctly returned.

- 
- Step 1** The OSS builds and submits a batch containing the API calls listed in [API Calls Affected by the Lease Reservation Feature, page C-50](#), for adding or removing reservations.
- Step 2** The RDU receives the batch and processes it. During the processing, the RDU (via the API commands listed in [API Calls Affected by the Lease Reservation Feature, page C-50](#)) makes an external CCM call for lease reservation related tasks.
- Step 3** CCM is unavailable and an error occurs. Return status code from the API call will be set to `CommandStatusCodes .CMD_ERROR_CCM_UNREACHABLE`.
- 

## A Submitted Batch Fails when BAC Does Not Use CCM

BAC is *not* configured to use CCM. The OSS submits an API request to add or remove a reservation.

### Desired Outcome

Submitted batch failed. No reservation is added/removed in the BAC/Network Registrar system; a predefined, well-known error code is correctly returned.

- 
- Step 1** The OSS builds and submits a batch containing the API calls listed in [API Calls Affected by the Lease Reservation Feature, page C-50](#), for adding or removing reservations.

- Step 2** The RDU receives the batch and processes it. During the processing, the RDU (via the API commands listed in [API Calls Affected by the Lease Reservation Feature, page C-50](#)) detects that CCM is not configured for the lease reservation feature.
- Step 3** An error occurs. Return status code from the API call will be set to CommandStatusCodes .CMD\_ERROR\_CCM\_NOT\_CONFIGURED.
-





## GLOSSARY

---

### A

- alert** A syslog or SNMP message notifying an operator or administrator of a problem.
- API** Application programming interface. Specification of function-call conventions that defines an interface to a service.
- audit log** A log file containing a summary of major changes in the RDU database. This includes changes to system defaults, technology defaults, DHCP criteria, and Class of Service.

---

### B

- bandwidth** The difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.
- BACC** *See* Broadband Access Center.
- broadband** Transmission system that multiplexes multiple independent signals onto one cable. In Telecommunications terminology; any channel having a bandwidth greater than a voice-grade channel (4 kHz). In LAN terminology; a co-axial cable on which analog signaling is used.
- Broadband Access Center** An integrated solution for data-over-cable service providers to configure and manage broadband modems, and enable and administer subscriber self-registration and activation. BAC is a scalable product capable of supporting millions of devices.

---

### C

- cable modem termination system** *See* CMTS.
- CableHome** A CableLabs initiative to develop a standardized infrastructure to let cable operators extend high-quality, value-added services to the home local area network.
- caching** A form of replication in which information learned during a previous transaction is used to process later transactions.
- client class** A Network Registrar feature that provides differentiated services to users that are connected to a common network. The client class is used in the BAC DHCP criteria to provide differentiated DHCP services to devices.

|                                 |                                                                                                                                                                                                                     |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CMTS</b>                     | Cable modem termination system. A CMTS is a component that exchanges digital signals with cable modems on a cable network. The CMTS is usually located in the cable provider's local office.                        |
| <b>CMTS shared secret</b>       | <i>See</i> shared secret.                                                                                                                                                                                           |
| <b>configuration file</b>       | A file containing configuration parameters for the device to be provisioned.                                                                                                                                        |
| <b>configuration generation</b> | The process of generating configurations at the RDU for devices and distributing them to the DPE. The configuration instructions are cached by the DPE and informed about action needed to be performed on the CPE. |
| <b>CPE</b>                      | Customer premises equipment. Terminating equipment, such as telephones, computers, and modems, supplied and installed at a customer location.                                                                       |

---

## D

|                                                        |                                                                                                                                                                                                 |
|--------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Data Over Cable Service Interface Specification</b> | <i>See</i> DOCSIS.                                                                                                                                                                              |
| <b>device provisioning engine</b>                      | <i>See</i> DPE.                                                                                                                                                                                 |
| <b>DOCSIS</b>                                          | Data over cable service interface specification. DOCSIS defines functionality in cable modems involved in high-speed data distribution over cable television system networks.                   |
| <b>DOCSIS Shared Secret</b>                            | Shared secret for communication between DOCSIS devices in a BAC deployment.                                                                                                                     |
| <b>DPE</b>                                             | Device provisioning engine. The DPE caches device information. These distributed serves automatically synchronize with the RDU to obtain the latest configurations and provide BAC scalability. |
| <b>DSTB</b>                                            | Digital set-top box. A device that enables a television to become a user interface to the Internet and to receive and decode digital television signals.                                        |
| <b>Dynamic Configuration File</b>                      | A dynamically created configuration file that uses template files to provide greater flexibility and security in the provisioning process.                                                      |

---

## F

|             |                                                                                                                                                               |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FQDN</b> | Fully qualified domain name. FQDN is the full name of a system, rather than just its hostname. For example, cisco is a hostname and www.cisco.com is an FQDN. |
|-------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|

---

## I

|                   |                                                                                                                                      |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>IP address</b> | An IP address is a 32-bit number that identifies each sender or receiver of information that is sent in packets across the Internet. |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------|



---

**K**

**KDC** A key distribution center that implements limited Kerberos functionality. Used in the provisioning of PacketCable MTAs.

---

**M**

**MAC address** Standardized data link layer address that is required for every port or device that connects to a LAN. Other devices in the network use these addresses to locate specific ports in the network and to create and update routing tables and data structures. MAC addresses are 6 bytes long and are controlled by IEEE. Also known as hardware address, MAC-layer address, or physical address. Compare with *network address*.

**MTA** Media Terminal Adapter. Equipment at the customer end of a broadband (PacketCable) network.

---

**N**

**NAT** Network address translation. Mechanism for reducing the need for globally unique IP addresses. NAT allows an organization with addresses that are not globally unique to connect to the Internet by translating those addresses into globally routeable address space. This is also known as Network Address Translation.

**network administrator** Person responsible for operation, maintenance, and management of a network. *See also* network operator.

**network operator** Person who routinely monitors and controls a network, performing such tasks as reviewing and responding to alarms, monitoring throughput, configuring new circuits, and resolving problems. *See also* network administrator.

**Network Time Protocol** *See* NTP.

**NR** Cisco Network Registrar. A software product that provides IP addresses, configuration parameters, and DNS names to DOCSIS cable modems and PCs, based on network and service policies.

**NTP** Network Time Protocol. NTP is a protocol designed to synchronize server clocks over a network.

---

**P**

**provisioning API** A series of BAC functions that programs can use to make the operating system perform various functions.

**provisioning groups** Groupings of devices with a defined set of associated DPE and DHCP servers, based on either network topology or geography.

**publishing** The process of publishing provisioning information to an external datastore in real time. Publishing plug-ins must be developed to write data to a datastore.

---

**R**

- RDU** Regional distribution unit. The RDU is the primary server in the BAC provisioning system. It manages generation of device configurations, processes all API requests, and manages the BAC system.
- realm** The logical network served by a single Kerberos database and a set of Key Distribution Centers.
- realm names** By convention, realm names are generally all uppercase letters, to differentiate the realm from the Internet domain. *See* realm.
- redundancy** In internetworking, the duplication of devices, services, or connections so that, in the event of a failure, the redundant devices, services, or connections can perform the work of those that failed.

---

**S**

- selection tags** Selection tags associated with Network Registrar scopes. These tags define the clients and client classes associated with a scope.
- shared secret** A character string used to provide secure communication between two servers or devices.
- static configuration files** These files are used as a configuration file for a device. For example, a static configuration file called gold.cm would identify the gold DOCSIS class of service. BAC treats this file type like any other binary file.

---

**T**

- template files** Text files that contain DOCSIS or PacketCable MTA options and values that, when used in conjunction with a DOCSIS or PacketCable MTA Class of Service, provide dynamic file generation.
- TFTP** Trivial File Transfer Protocol. Simplified version of File Transfer Protocol (FTP) that allows files to be transferred from one computer to another over a network.
- TLV** Type-Length-Value. A tuple within a DOCSIS or PacketCable configuration file.
- tuple** In programming languages, a tuple is an ordered set of values. Common uses for the tuple as a data type are: for passing a string of parameters from one program to another, or to represent a set of value attributes in a relational database.
- Type Length Value** *See* TLV.

---

**U**

- uBr** Universal Broadband Router (such as the Cisco 7246 or 7223), which is the Cisco router implementation of a DOCSIS CMTS.

---

**V**

**VoIP** Voice over IP. VoIP is the ability to make telephone calls and send faxes over IP-based data networks with a suitable quality of service (QoS) and superior cost/benefit.

---

**W**

**watchdog agent** A watchdog agent is a daemon process that is used to monitor, stop, start, and restart BAC component processes such as the RDU, Tomcat, and the SNMP agent.

---

**X**

**XGCP** A Gateway Control Protocol used to pass data between networks. This includes M (for Media) GCP and S (Simple) GCP.





## INDEX

### A

- adding a license [11-31](#)
- adding vendor-specific MIBs [8-9](#)
- administrator provisioning examples
  - accounts, maintaining [12-7](#)
    - cable modems, managing [12-8](#)
    - class of service, managing [12-7](#)
    - computers, managing [12-8](#)
    - deleting an account [12-8](#)
    - new accounts, registering [12-7](#)
  - accounts, searching for
    - by account number [12-6](#)
    - by IP address [12-6](#)
    - by MAC address [12-6](#)
- administrator user interface
  - about [2-20](#)
  - accessing [9-2](#)
  - class of service
    - about [11-1](#)
    - adding [11-3](#)
    - deleting [11-5](#)
    - modifying [11-4](#)
  - configuring [9-1](#)
  - custom property, configuring [11-5](#)
  - defaults, configuring
    - about [11-6](#)
    - ATA 186 [11-7](#)
    - ATA 188 [11-8](#)
    - CableHome WAN-Data [11-8](#)
    - CableHome WAN-MAN [11-8, 11-11](#)
    - computer [11-12](#)
    - DOCSIS [11-13](#)
    - Network Registrar [11-15](#)
    - PacketCable [11-17](#)
    - RDU [11-19](#)
    - system [11-21](#)
    - xGCP [11-23](#)
  - devices, managing
    - about [10-4](#)
    - adding record [10-14](#)
    - deleting record [10-15](#)
    - modifying record [10-14](#)
    - regenerating configuration [10-15](#)
    - relating and unrelating [10-16](#)
    - resetting [10-16](#)
    - searching [10-5](#)
    - unregistering [10-16](#)
    - viewing details [10-9](#)
  - DHCP criteria, managing
    - about [11-24](#)
    - adding [11-24](#)
    - deleting [11-25](#)
    - modifying [11-25](#)
  - dynamic DOCSIS version selection, configuring [4-5](#)
  - external files, managing [11-26](#)
    - adding [11-27](#)
    - deleting [11-30](#)
    - exporting [11-29](#)
    - replacing [11-29](#)
    - viewing [11-27](#)
  - license keys, managing [11-30](#)
    - adding [11-31](#)
    - deleting [11-31](#)
    - modifying [11-31](#)
  - logging in [9-2, 9-5](#)

- HTTP over SSL [12-2](#)
- logging out [9-5](#)
- navigating [9-5](#)
- nodes, managing
  - adding [10-18](#)
  - deleting [10-18](#)
  - modifying [10-18](#)
  - viewing details [10-19](#)
- node types, managing [10-16](#)
  - adding [10-17](#)
  - deleting [10-18](#)
  - modifying [10-17](#)
  - relating and unrelating to nodes [10-19](#)
- provisioning data, publishing [11-35](#)
  - disabling plug-in [11-35](#)
  - enabling plug-in [11-35](#)
  - modifying plug-in settings [11-36](#)
- RDU extensions, managing [11-32](#)
  - installing custom points [11-34](#)
  - required points (table) [11-32](#)
  - viewing [11-34](#)
  - writing new class [11-33](#)
- servers, monitoring
  - DPE [10-19](#)
  - Network Registrar extensions [10-23](#)
  - provisioning group [10-25](#)
  - RDU [10-25](#)
- starting [9-1](#)
- stopping [9-1](#)
- understanding icons (table) [9-6](#)
- users, managing
  - about [10-1, 10-13](#)
  - adding [10-2](#)
  - deleting [10-3](#)
  - modifying [10-3](#)
- adminui.properties file [9-1](#)
- advanced concepts
  - See* tools and advanced concepts
- agents
  - alerts [A-5](#)
  - BAC architecture and [2-13](#)
  - process watchdog
    - about [2-14](#)
    - command line, using [2-14](#)
    - commands (table) [2-14](#)
  - SNMP agent
    - about [2-13](#)
    - MIB support [2-13](#)
- alert messages [A-1](#)
  - message format [A-1](#)
  - relating to
    - Network Registrar extensions [A-6](#)
    - process watchdog [A-5](#)
    - RDU [A-2](#)
    - Solaris DPE [A-3](#)
- architecture [2-1](#)
  - administrator user interface [2-20](#)
  - agents
    - process watchdog [2-14](#)
    - SNMP agent [2-13](#)
  - DPE [2-4](#)
    - device types [2-5](#)
    - DSS, and [2-9](#)
    - license keys [2-5](#)
    - server states [2-7](#)
    - synchronization with RDU [2-7](#)
    - TACACS+ authentication [2-6](#)
    - TFTP server [2-8](#)
  - KDC [2-11](#)
    - certificates [2-12, 5-9](#)
    - default KDC properties [5-7](#)
    - licenses [2-12, 5-9](#)
    - multiple realm support [5-10](#)
- logging [2-15](#)
  - log files [2-18, 2-19](#)
  - log files, rotating [2-17](#)
  - log levels and structure [2-15](#)
  - severity levels (table) [2-16](#)

- severity levels, configuring [2-17](#)
- MIBs [2-12, 2-13](#)
- Network Registrar [2-10](#)
  - DHCP, and [2-10](#)
  - DNS, and [2-11](#)
  - lease reservation [2-11](#)
- provisioning groups [2-10](#)
- RDU [2-3](#)
  - configuration generation [2-3](#)
  - service-level selection [2-4](#)
- registration modes [2-2](#)
  - mixed mode [2-3](#)
  - promiscuous mode [2-2](#)
  - roaming mode [2-2](#)
  - standard mode [2-2](#)
- sample user interface [2-20](#)
- ATA 186 defaults, configuring [11-7](#)
- ATA 188 defaults, configuring [11-8](#)
- audit.log [2-18](#)
- automatic FQDN generation [11-38](#)

---

## B

- backup and recovery of database [14-4](#)
  - See also* database management
- bacupDb.sh tool [14-4](#)

---

## C

### CableHome

- configuring [7-1](#)
  - DPE [7-4](#)
  - Network Registrar [7-3](#)
  - RDU [7-3](#)
- option support [8-26](#)
- provisioning, non-secure
  - checklist [3-11](#)
  - flow [7-1](#)

- support [1-2](#)
  - WAN defaults, configuring [11-8](#)
- CableHome WAN-Data default, configuring [11-10](#)
- CableHome WAN-MAN default, configuring [11-11](#)
- CableLabs certificate trust hierarchy [6-9](#)
  - certificate, validating [6-10](#)
- MTA device [6-10](#)
  - device certificate [6-12](#)
  - manufacturer certificate [6-11](#)
  - root certificate [6-11](#)
- operational ancillary certificates
  - Delivery Function (DF) certificate [6-17](#)
  - KDC certificate [6-16](#)
  - PacketCable Server certificates [6-17](#)
- service provider [6-13](#)
  - CA certificate [6-14](#)
  - CA certificate, local system [6-15](#)
  - root certificate [6-14](#)
- CableLabs code verification certificate hierarchy
  - CA certificate [6-20](#)
  - certificate revocation lists [6-22](#)
  - manufacturer certificate [6-21](#)
  - requirements [6-19](#)
  - root CA certificate [6-20](#)
  - service provider certificate [6-21](#)
- cautions, regarding
  - cnr\_ep.properties file, setting property instances [B-1](#)
  - custom properties, deleting [11-5](#)
  - DHCP options, settings in Network Registrar [3-5](#)
  - DOCSIS Modem class of service, adding [11-3](#)
  - DSS, configuring multiple in provisioning group [2-9](#)
  - evaluation license key, deploying in network [11-30](#)
  - KDC certificates, missing or uninstalled [2-12, 5-9](#)
  - KDC license, copying [5-10](#)
  - SUI [12-1](#)
  - template files, deleting [11-30](#)
  - troubleshooting devices by MAC address [13-22](#)
- certificate trust hierarchy, PacketCable [6-9](#)
  - ancillary certificates

- delivery function [6-17](#)
  - KDC [6-16](#)
  - PacketCable server [6-17](#)
- certificate validation [6-10](#)
- MTA [6-10](#)
  - device certificate [6-12](#)
  - manufacturer certificate [6-11](#)
  - root certificate [6-11](#)
- service provider [6-13](#)
  - CA certificate [6-14](#)
  - CA certificate, local system [6-15](#)
  - root certificate [6-14](#)
- changeNRProperties.sh tool [13-13](#)
- CISCO-BACC-DPE-MIB [2-13](#)
- CISCO-BACC-RDU-MIB [2-13](#)
- CISCO-BACC-SERVER-MIB [2-13](#)
- CISCO-CW-APPLIANCE-MIB [2-13](#)
- CISCO-NMS-APPL-HEALTH-MIB [2-13](#)
- class of service, managing [11-1](#)
  - configuring
    - adding a class [11-3](#)
    - deleting a class [11-5](#)
    - modifying a class [11-4](#)
  - via SUI
    - administrator provisioning example [12-7](#)
    - configuring sample [12-2](#)
- code verification certificate hierarchy, PacketCable [6-19](#)
  - CA certificate [6-20](#)
  - code verification certificate requirements [6-19](#)
  - manufacturer certificate [6-21](#)
  - root CA certificate [6-20](#)
  - service provider certificate [6-21](#)
- code verification certificate requirements [6-19](#)
- computer defaults, configuring [11-12](#)
- configuration file utility, using [8-27](#)
  - adding template [8-29](#)
  - binary file
    - converting to template files [8-30](#)
    - external, viewing [8-38](#)
    - local, viewing [8-37](#)
    - output, specifying [8-36](#)
- dynamic DOCSIS version selection, configuring [4-5](#)
- macro variables
  - specifying a device for [8-35](#)
  - specifying from CLI [8-34](#)
- PacketCable Basic flow, activating [8-39](#)
- running [8-28](#)
- template processing, testing
  - external template files [8-32](#)
  - local template file and adding shared secret [8-33](#)
  - local template files [8-31](#)
- configuration workflows and checklists (tables) [3-1](#)
  - component workflows [3-1](#)
    - hardware DPE checklist [3-2](#)
    - Network Registrar checklist [3-5](#)
    - RDU checklist [3-1](#)
    - Solaris DPE checklist [3-3](#)
  - technology workflows [3-5](#)
    - DOCSIS [3-6](#)
    - non-secure CableHome [3-11](#)
    - PacketCable, Basic [3-10](#)
    - PacketCable, Secure [3-6](#)
- configuring BAC
  - class of service [11-1](#)
    - adding a class [11-3](#)
    - deleting [11-5](#)
    - modifying [11-4](#)
  - custom properties [11-5](#)
  - defaults [11-6](#)
    - ATA 186 [11-7](#)
    - ATA 188 [11-8](#)
    - CableHome WAN defaults [11-8](#)
    - computer [11-12](#)
    - configuration options, selecting [11-6](#)
    - DOCSIS [11-13](#)
    - Network Registrar extensions [11-15](#)
    - PacketCable [11-17](#)
  - DHCP criteria [11-24](#)



- adding criteria [11-24](#)
  - deleting criteria [11-25](#)
  - modifying criteria [11-25](#)
- external files, managing [11-26](#)
  - adding files [11-27](#)
  - deleting files [11-30](#)
  - exporting files [11-29](#)
  - replacing files [11-29](#)
  - viewing files [11-27](#)
- FQDN, automatic generation [11-38](#)
  - format [11-38](#)
  - properties [11-38](#)
  - sample [11-39](#)
  - validation [11-39](#)
- license keys, managing [11-30](#)
  - adding a license [11-31](#)
  - deleting a license [11-31](#)
  - modifying a license [11-31](#)
- provisioning data, publishing [11-35](#)
  - datastore changes [11-35](#)
  - plug-in settings, modifying [11-36](#)
- RDU unit extensions, managing [11-32](#)
  - custom extension points, installing [11-34](#)
  - new class, writing [11-33](#)
  - viewing [11-34](#)
- SNMPv3 cloning on RDU, DPE [11-37](#)
  - key generation [11-37](#)
  - key material [11-37](#)
- SRV record in Network Registrar DNS server [11-36](#)
- configuring CableHome
  - DPE [7-4](#)
  - Network Registrar [7-3](#)
  - provisioning flow [7-1](#)
  - RDU [7-3](#)
- configuring DOCSIS
  - features in BAC
    - DOCSIS version support [4-5](#)
    - DPE TFTP IP validation [4-4](#)
    - dynamic configuration TLVs [4-4](#)
    - dynamic DOCSIS version selection [4-5](#)
    - provisioning flow [4-1](#)
    - troubleshooting [4-6](#)
    - workflow checklist [3-6](#)
- configuring Network Registrar and CableHome [7-3](#)
  - defaults [11-15](#)
  - SRV record in DNS server [11-36 to 11-37](#)
  - workflow checklist (table) [3-5](#)
- configuring PacketCable [5-1](#)
  - automatic FQDN generation [11-38](#)
  - certificate trust hierarchies [6-22](#)
  - certificate trust hierarchies, certificate revocation [6-19](#)
  - defaults [11-17](#)
  - Euro PacketCable
    - about [5-29](#)
    - MIBs, configuring [5-30](#)
  - FQDN, automatic generation [11-38](#)
  - PacketCable Basic
    - provisioning flow [5-28](#)
  - PacketCable Secure
    - about [5-1](#)
    - KDC, configuring for multiple realms [5-10](#)
    - KDC properties [5-7](#)
    - provisioning flow [5-1](#)
  - service keys, generating via KeyGen tool [13-11](#)
  - troubleshooting eMTA provisioning
    - components involved [6-1](#)
    - key variables [6-3](#)
    - scenarios [6-5](#)
    - tools [6-4](#)
- configuring RDU
  - CableHome and
    - WAN-Data [7-4](#)
    - WAN-MAN [7-3](#)
  - defaults [11-19](#)
  - workflow checklist (table) [3-1](#)
- configuring SUI

- administrator provisioning examples [12-6](#)
  - accounts, maintaining [12-7](#)
  - accounts, searching for [12-6](#)
- sample configuration options
  - administrative access levels [12-3](#)
  - class of service [12-2](#)
  - ISP, selecting [12-3](#)
  - promiscuous mode [12-2](#)
  - technician login, using [12-3](#)
- sample sampleui.properties file [12-9](#)
- subscriber provisioning examples
  - promiscuous customer premise equipment registration [12-5](#)
  - standard customer premise equipment registration [12-4](#)
- cos/docsis/file/1.0, 1.1, 2.0 [4-5](#)

---

## D

- database
  - See* database management
- database management
  - backup and recovery [14-4](#)
    - backing up [14-4](#)
    - recovering [14-5](#)
    - restoring [14-6](#)
  - disk space
    - out of space, handling [14-3](#)
    - requirements [14-3](#)
  - failure resiliency [14-1](#)
  - files [14-2](#)
    - automatic log management [14-2](#)
    - DB\_VERSION [14-3](#)
    - history log [14-3](#)
    - storage [14-2](#)
    - transaction log [14-2](#)
  - location, changing [14-7](#)
  - RDU, migrating [14-8](#)
- Data Over Cable Service Interface Specification

- See* DOCSIS [4-1](#)
- defaults, configuring [11-6](#)
  - ATA 186 [11-7](#)
  - ATA 188 [11-8](#)
  - CableHome WAN [11-8](#)
    - WAN-Data [11-10](#)
    - WAN-MAN [11-11](#)
  - computer [11-12](#)
  - DOCSIS [11-13](#)
  - Network Registrar [11-15](#)
  - PacketCable [11-17](#)
  - RDU [11-19](#)
  - system [11-21](#)
  - xGCP [11-23](#)
- deleting a license [11-31](#)
- device management [10-4](#)
  - about [10-13](#)
  - adding devices [10-14](#)
  - controls [10-7](#)
  - deleting devices [10-15](#)
  - device configurations, regenerating [10-15](#)
  - device details, viewing [10-9](#)
  - modifying devices [10-14](#)
  - relating and unrelating devices [10-16](#)
  - resetting devices [10-16](#)
  - searching for devices [10-5](#)
  - unregistering devices [10-16](#)
- Device Provisioning Engine
  - See* DPE
- DHCP
  - criteria defaults, configuring [11-24](#)
    - adding criteria [11-24](#)
    - deleting criteria [11-25](#)
    - modifying criteria [11-25](#)
  - Network Registrar, and [2-10, 6-2](#)
- disk\_monitor.sh tool [13-21](#)
- disk space, monitoring [13-21](#)
- DNS, Network Registrar, and [2-11](#)
- DOCSIS

- /cos/docsis/file/1.0, 1.1, 2.0 [4-5](#)
- about [1-2](#)
- configuring
  - defaults [11-13](#)
  - provisioning workflow [4-1](#)
  - workflow checklist [3-6](#)
- dynamic version selection
  - configuration file [4-5](#)
  - device GIADDR [4-5](#)
- features
  - 1.0, 1.1, 2.0 version support [4-5](#)
  - dynamic configuration TLVs [4-4](#)
  - dynamic version selection [4-5](#)
- high-speed data support [1-2](#)
- MIBs, using with dynamic DOCSIS templates [4-3](#)
- option support [8-15](#)
- troubleshooting [4-6](#)

DOCSIS defaults, configuring [11-13](#)

DOCSIS shared secret

*See* DSS

Domain Name System

*See* DNS [2-11](#)

DPE

- about [2-4](#)
- alerts, Solaris [A-3](#)
- configuring
  - CableHome and [7-4](#)
  - SNMPv3 cloning [11-37](#)
- configuring DOCSIS shared secret [2-9](#)
- device types [2-5](#)
  - hardware [2-5](#)
  - Solaris [2-5](#)
- DSS
  - about [2-9](#)
  - resetting [2-9](#)
- license keys [2-5](#)
- log file
  - about [2-18](#)
  - viewing [2-19, 9-2, 10-22](#)

- server, viewing details [10-19](#)
- server state [2-7](#)
- SNMP agent [2-13](#)
- SNMPv3 cloning, configuring [11-37](#)
  - key generation [11-37](#)
  - key material [11-37](#)
- synchronization with RDU [2-7](#)
- TACACS+, and DPE authentication [2-6](#)
  - client settings [2-6](#)
  - privilege levels [2-6](#)
- TFTP server, and [2-8](#)
- viewing details [10-19](#)
- workflow checklist
  - hardware [3-2, 3-3](#)
  - Solaris [3-3](#)

dpe.log [2-19](#)

DSS, and DPEs

- about [2-9](#)
- resetting [2-9](#)

dynamic DOCSIS version selection

- about [4-5](#)
- configuration file [4-5](#)
- device GIADDR, using [4-5](#)

---

## E

- eMTA provisioning for PacketCable, troubleshooting [6-1](#)
  - components
    - call management server [6-3](#)
    - DHCP server [6-2](#)
    - DNS server [6-2](#)
    - embedded MTA [6-2](#)
    - KDC [6-2](#)
    - PacketCable provisioning server [6-3](#)
  - key variables
    - certificates [6-3](#)
    - MTA configuration file [6-4](#)
    - scope-selection tag [6-4](#)
- error messages, RDU [A-7](#)

extension points

*See* Network Registrar

extensions, RDU [11-32](#)

external files, managing [11-26](#)

adding [11-27](#)

deleting [11-30](#)

exporting [11-29](#)

replacing [11-29](#)

viewing [11-27](#)

---

## F

features, overview [1-1](#)

FQDN, automatic generation

about [11-38](#)

format [11-38](#)

properties [11-38](#)

sample [11-39](#)

validation [11-39](#)

---

## G

Gateway Control Protocol

*See* xGCP

GUI

*See* administrator user interface

---

## H

hardware DPE, DPE-2115

about [2-5](#)

configuring workflow [3-2](#)

---

## I

include files [8-3](#)

ISP, selecting [12-3](#)

---

## K

KDC

BAC architecture, and [2-11](#)

certificates [5-9](#)

certificates, managing via PKCert.sh tool

creating [13-6](#)

running the PKCert tool [13-5](#)

setting log level for debug output [13-8](#)

validating [13-7](#)

default properties [5-7](#)

licenses [5-9](#)

multiple realm support

about [5-10](#)

configuring [5-11](#)

directory structure (table) [5-11](#)

template, authoring [5-25](#)

verifying service keys [13-13](#)

KeyGen tool

using [13-11](#)

verifying service keys [13-13](#)

---

## L

license keys, managing [11-30](#)

about [11-30](#)

adding a license [11-31](#)

deleting a license [11-31](#)

KDC [5-9](#)

modifying a license [11-31](#)

logging

BAC architecture, and [2-15](#)

log files

DPE [2-18](#)

Network Registrar [2-19](#)

RDU [2-18](#)

rotating [2-17](#)

log levels and structures [2-15](#)

log level tool, using [13-2](#)

- severity levels (table) [2-16](#)
- severity log levels, configuring [2-17](#)
- logging in [9-2](#)
- logging out [9-5](#)
- log level tool, using [13-2](#)
  - setting [13-3](#)
  - viewing log level [13-4](#)

---

## M

### MAC address

- troubleshooting devices [13-21](#)

### MIBs

- BAC architecture, and [2-12](#)
- CableHome, and SNMP VarBind [8-6](#)
- DOCSIS, and SNMP VarBind [8-5](#)
- Euro PacketCable, and PacketCable voice configuration [5-30](#)
- PacketCable, and SNMP VarBind [8-6](#)
- SNMP agent, and MIB support [2-13](#)
- TLV 38, and MIB support [5-29](#)
- vendor-specific, adding [8-9](#)

- migrating, RDU database [14-8](#)

- modes of registration [2-2](#)

---

## N

### Network Registrar

- about [2-10](#)
  - DHCP, and [2-10](#)
  - DNS, and [2-11](#)
  - lease reservation [2-11](#)
- architecture [2-10](#)
- configuring CableHome [7-3](#)
- defaults, configuring [11-15](#)
- DHCP, and [2-10](#)
- DNS
  - about [2-11](#)
  - SRV record, configuring [11-36](#)

- extension point alerts [A-6](#)
- extension points, viewing details [10-23](#)
- lease reservation [2-11](#)
- log file [2-19](#)
- viewing details [10-23](#)
- workflow checklist [3-5](#)

- Network Registrar defaults, configuring [11-15](#)

- Network Registrar log [2-19](#)

- nodes, managing [10-16](#)

- about [10-18](#)
- adding [10-18](#)
- deleting [10-18](#)
- details, viewing [10-19](#)
- modifying [10-18](#)
- node types [10-16](#)
  - adding [10-17](#)
  - deleting [10-18](#)
  - modifying [10-17](#)
- relating and unrelating node types to nodes [10-19](#)

- NRProperties.sh tool, using [13-13](#)

---

## O

### option support

- CableHome non-secure [8-26](#)
- DOCSIS [8-15](#)
- PacketCable [8-25](#)

### overview

- features and benefits [1-1](#)
- product [1-1](#)
- technologies supported [1-1](#)

---

## P

### PacketCable

- about [1-2](#)
- BAC properties, mapping to DHCP options [B-1](#)
  - Option 122 and BAC property comparison [B-1](#)

- Option 177 and BAC property comparison [B-2](#)
  - defaults, configuring [11-17](#)
    - xGCP gateway control protocol defaults [11-23](#)
  - MTAs, SNMPv3 cloning, and [11-37](#)
    - key generation [11-37](#)
    - key material [11-37](#)
  - option support [8-25](#)
  - viewing device details [10-8](#)
  - voice services support
    - non-secure [1-2](#)
    - standard [1-2](#)
  - workflow checklists [3-6](#)
    - Euro PacketCable [3-7, 3-10](#)
    - non-secure, Basic PacketCable [3-10](#)
    - Secure PacketCable [3-6](#)
  - PacketCable defaults, configuring [11-17](#)
  - PacketCable service keys, generating [13-11](#)
  - PacketCable voice configuration [5-1](#)
    - certificate trust hierarchy [6-9](#)
      - CableLabs Service Provider [6-13](#)
      - code verification [6-19](#)
      - MTA device certificate [6-12](#)
      - MTA device certificate hierarchy [6-10](#)
      - revocation [6-19](#)
      - validation [6-10](#)
  - defaults, configuring [11-17](#)
  - eMTA provisioning, troubleshooting [6-1](#)
    - components [6-1](#)
    - key variables [6-3](#)
  - Euro PacketCable
    - about [5-29](#)
    - MIBs, configuring [5-30](#)
  - PacketCable Basic
    - checklist [3-10](#)
    - provisioning workflow [5-28](#)
    - SNMP v2C notifications [5-29](#)
    - TLV 38 and MIB support [5-29](#)
  - PacketCable Secure
    - Euro PacketCable checklist [3-6](#)
    - North American checklist [3-7](#)
    - provisioning workflow [5-1](#)
  - troubleshooting scenarios [6-5 to 6-8](#)
  - troubleshooting tools
    - Ethereal, SnifferPro, and other [6-5](#)
    - logs [6-5](#)
  - PKCert.sh tool, using [13-5](#)
    - KDC certificate, creating [13-6](#)
    - KDC certificates, validating [13-7](#)
    - running [13-5](#)
    - setting log level for debug output [13-8](#)
    - validating certificates [13-7](#)
  - product overview [1-1](#)
  - promiscuous mode [2-2](#)
  - provisioning data, publishing [11-35](#)
    - datastore changes [11-35](#)
    - plug-in settings, modifying [11-36](#)
  - provisioning group
    - viewing details [10-25](#)
  - provisioning groups [2-10](#)
- 
- ## R
- RDU
    - about [2-3](#)
    - alert messages [A-2](#)
    - alerts [A-2](#)
    - configuration generation [2-3](#)
    - configuring, and CableHome
      - WAN-Data [7-4](#)
      - WAN-MAN [7-4](#)
    - database migration [14-8](#)
    - defaults, configuring [11-19](#)
    - device configuration, generating [2-3](#)
    - extensions, managing
      - custom extension points, installing [11-34](#)
      - new class, writing [11-33](#)
      - viewing [11-34](#)
    - log files

- setLogLevel.sh tool, using [13-2](#)
- log level tool, using [13-2](#)
  - current log level, viewing [13-4](#)
  - setting [13-3](#)
- logs
  - about [2-18](#)
  - default log level [13-2](#)
  - viewing [2-18, 9-2, 10-27](#)
- server, viewing [10-25](#)
- service-level selection [2-4](#)
- SNMP agent [2-13](#)
- SNMPv3 cloning, configuring [11-37](#)
  - key generation [11-37](#)
  - key material [11-37](#)
- unit extensions, managing [11-32](#)
- viewing details [10-25](#)
- workflow checklist [3-1](#)

rdu.log [2-18](#)

RDU defaults, configuring [11-19](#)

recoverDb.sh tool [14-5](#)

registration modes
 

- mixed [2-3](#)
- promiscuous [2-2](#)
- roaming [2-2](#)
- standard [2-2](#)

restoreDb.sh tool [14-6](#)

runCfgUtil.sh script, running [8-28](#)

---

## S

sample user interface

*See* SUI

- servers, viewing [10-19](#)
  - DPE [10-19](#)
  - Network Registrar extensions [10-23](#)
  - provisioning groups [10-25](#)
  - RDU [10-25](#)
- service classes
  - See* class of service

- service keys, PacketCable [13-11](#)
- setLogLevel.sh tool [13-3](#)
- shared secret
  - configuration file utility, and [8-33](#)
  - DSS (DOCSIS Shared Secret)
    - about [2-9](#)
    - DPEs, and [2-9](#)
    - resetting [2-9](#)
- SNMP
  - adding TLVs without MIB [8-8](#)
  - agent
    - about [2-13](#)
    - MIB support [2-13](#)
    - starting [13-18](#)
    - stopping [13-18](#)
  - cloning on PacketCable eMTA (use case) [C-34](#)
  - configuring v3 cloning [11-37](#)
  - SNMP agent
    - community, adding [13-17](#)
    - community, deleting [13-17](#)
    - location, changing [13-19](#)
    - settings, listing [13-20](#)
    - starting [13-18](#)
    - stopping [13-18](#)
  - snmpAgentCfgUtil.sh tool [13-15](#)
    - hosts, adding [13-16](#)
    - hosts, deleting [13-16](#)
    - SNMP contacts, setting up new [13-19](#)
    - SNMP listening port, identifying [13-18](#)
    - SNMP notification types, specifying [13-20](#)
  - v3 cloning, configuring on RDU, DPE
    - key generation [11-37](#)
    - key material [11-37](#)
- snmpAgentCfgUtil.sh
  - adding agent community [13-17](#)
  - adding a host [13-16](#)
  - changing agent location [13-19](#)
  - configuring agent port [13-18](#)
  - deleting agent community [13-17](#)

- deleting a host [13-16](#)
  - setting up contacts [13-19](#)
  - specify notification type [13-20](#)
  - view agent settings [13-20](#)
  - SRV record in Network Registrar DNS server, configuring [11-36 to 11-37](#)
  - subscriber provisioning examples
    - promiscuous customer premise equipment registration [12-5](#)
      - existing cable modem and new computer [12-6](#)
      - new cable modem and new computer [12-5](#)
    - standard customer premise equipment registration [12-4](#)
      - existing cable modem and new computer [12-4](#)
      - existing computer ISP, altering [12-5](#)
      - new cable modem and new computer [12-4](#)
  - SUI
    - about [2-20, 12-1](#)
    - accessing
      - HTTP [9-3](#)
      - HTTP over SSL [9-3](#)
    - configuring [12-1](#)
      - administrative access levels [12-3](#)
      - class of service [12-2](#)
      - ISP, selecting [12-3](#)
      - promiscuous mode [12-2](#)
      - sample configuration options [12-2](#)
      - starting and stopping [12-2](#)
      - technician login, using [12-3](#)
    - sampleui.properties file sample [12-9](#)
  - syslog alerts
    - See* alert messages
  - system defaults, configuring [11-21](#)
- 
- T**
- template files, developing [8-1](#)
    - definition options, encoding types for [8-12](#)
      - BITS value syntax [8-14](#)
      - OCTETSTRING syntax [8-14](#)
  - DOCSIS option support [8-15](#)
    - grammar [8-2](#)
      - comments [8-2](#)
      - include files [8-3](#)
      - instance modifier [8-4](#)
      - options [8-3](#)
    - macro variables [8-6](#)
    - non-secure CableHome option support [8-26](#)
    - option support
      - DOCSIS [8-15](#)
      - non-secure CableHome [8-26](#)
      - PacketCable [8-25](#)
    - PacketCable option support [8-25](#)
    - SNMP VarBind [8-5](#)
    - SNMP Varbind
      - CableHome MIBs [8-6](#)
      - DOCSIS MIBs [8-5](#)
      - PacketCable MIBs [8-6](#)
  - tools
    - bprAgent, using [2-14](#)
    - changeNRProperties.sh, using [13-13](#)
    - configuration file utility, using [8-27](#)
    - disk\_monitor.sh, using [13-21](#)
    - KeyGen, using [13-11](#)
    - PKCert.sh, using [13-5](#)
    - RDU log level, using [13-2](#)
    - setLogLevel.sh, using [13-2](#)
    - snmpAgentCfgUtil.sh, using [13-15](#)
  - tools and advanced concepts [13-1](#)
    - configuration file utility [8-27](#)
      - binary file, external, viewing [8-38](#)
      - binary file, local, viewing [8-37](#)
      - binary file output, specifying [8-36](#)
      - binary files, converting to template files [8-30](#)
      - macro variables, specifying a device for [8-35](#)
      - macro variables, specifying through CLI [8-34](#)
      - PacketCable Basic flow, activating [8-39](#)
      - running [8-28](#)



- testing template processing, external files [8-32](#)
- testing template processing, local files [8-31](#)
- testing template processing, local files and adding shared secret [8-33](#)
- using tool [8-27](#)
- disk\_monitor.sh tool [13-21](#)
- KeyGen tool [13-11](#)
- NRProperties.sh tool [13-11 to 13-15](#)
- PKCert.sh Tool [13-5 to 13-7](#)
  - KDC certificate, creating [13-6](#)
  - KDC certificates, validating [13-7](#)
  - running [13-5](#)
  - setting log level [13-8](#)
- RDU log level tool [13-2 to 13-5](#)
  - current log level, viewing [13-4](#)
  - setting [13-3](#)
- snmpAgentCfgUtil.sh tool [13-15](#)
  - hosts, adding [13-16](#)
  - hosts, deleting [13-16](#)
  - SNMP agent, starting [13-18](#)
  - SNMP agent, stopping [13-18](#)
  - SNMP agent community, adding [13-17](#)
  - SNMP agent community, deleting [13-17](#)
  - SNMP agent location, changing [13-19](#)
  - SNMP agent settings, listing [13-20](#)
  - SNMP contacts, setting up new [13-19](#)
  - SNMP listening port, identifying [13-18](#)
  - SNMP notification types, specifying [13-20](#)
- template files, developing
  - definition options, encoding types for [8-12](#)
  - DOCSIS option support [8-15](#)
  - grammar [8-2](#)
  - macro variables [8-6](#)
  - non-secure CableHome option support [8-26](#)
  - PacketCable option support [8-25](#)
  - SNMP VarBind [8-5](#)
- troubleshooting devices by MAC address [13-21](#)
  - relating device to node [13-22](#)
  - viewing devices [13-23](#)

- troubleshooting
  - alert messages [A-1](#)
    - message format [A-1](#)
    - RDU alerts [A-2](#)
    - Solaris DPE alerts [A-3](#)
    - watchdog agent alerts [A-5](#)
  - device, using MAC address [13-21](#)
  - devices, using MAC address
    - relating device to node [13-22](#)
    - sample log output [13-23](#)
    - viewing devices [13-23](#)
  - DOCSIS networks [4-6](#)
  - eMTA provisioning for PacketCable [6-1](#)
    - components [6-1](#)
    - key variables [6-3](#)
    - logs [6-5](#)
    - scenarios [6-5](#)
    - tools [6-4](#)
  - tools for PacketCable voice configuration [6-4](#)
- troubleshooting PacketCable provisioning [6-1](#)
- troubleshooting PacketCable voice technology
  - components
    - call management server [6-3](#)
    - DHCP server [6-2](#)
    - DNS server [6-2](#)
    - eMTA [6-2](#)
    - KDC [6-2](#)
    - PacketCable provisioning server [6-3](#)
  - key variables
    - certificates [6-3](#)
    - MTA configuration file [6-4](#)
    - scope-selection tag [6-4](#)

---

## U

- uBr, definition [1-4](#)
- upgrading license keys [11-31](#)
- use cases
  - about [C-1](#)

- adding
    - new computer behind a modem with NAT [C-23](#)
    - new computer in fixed standard mode [C-5](#)
    - second computer in promiscuous mode [C-21](#)
  - bulk provisioning 100 modems in promiscuous mode [C-17](#)
  - CableHome with firewall configuration [C-43](#)
  - disabling a subscriber [C-7](#)
  - getting detailed device information [C-27](#)
  - incremental provisioning of PacketCable eMTA [C-35](#)
  - Lease Reservation use cases [C-49](#)
    - API calls affected by lease reservation [C-50](#)
    - assigning new device with old device IP address [C-52](#)
    - bringing a device online using service provider IP address [C-50](#)
    - rebooting a device with the same IP address [C-54](#)
    - removing a device from BAC [C-55](#)
    - removing and re-creating a reservation [C-51](#)
    - removing a reservation and assigning a new IP address [C-53](#)
    - submitted batch fails when BAC does not use CCM [C-56](#)
    - submitted batch fails when BAC uses CCM [C-56](#)
  - logging
    - batch completions using events [C-27](#)
    - device deletions using events [C-25](#)
  - modifying an existing modem [C-11](#)
  - monitoring an RDU connection using events [C-26](#)
  - moving a device to another DHCP scope [C-24](#)
  - optimistic locking [C-39](#)
  - preprovisioning
    - CableHome WAN-MAN [C-41](#)
    - DOCSIS modems with dynamic configuration files [C-37](#)
    - first-time activation in promiscuous mode [C-19](#)
    - modems and self-provisioned computers [C-9](#)
    - PacketCable eMTA [C-32](#)
  - replacing an existing modem [C-20](#)
  - retrieving, devices matching a vendor prefix [C-30](#)
    - retrieving device capabilities for CableHome WAN-MAN [C-45](#)
    - searching using the default class of service [C-28](#)
    - self-provisioning
      - CableHome WAN-MAN [C-47](#)
      - first-time activation in promiscuous mode [C-14](#)
      - first-time activation with NAT [C-21](#)
      - modem and computer in fixed standard mode [C-2](#)
      - SNMP cloning on PacketCable eMTA [C-34](#)
      - subscriber bandwidth, temporarily throttling [C-40](#)
      - unregistering and deleting subscriber device [C-12](#)
  - user interface
    - See* administrator user interface
  - users, managing [10-1](#)
    - adding [10-2](#)
    - deleting [10-3](#)
    - modifying [10-3](#)
  - Users menu, about
    - Administrator [10-1](#)
    - Read/Write user [10-2](#)
    - Read-Only user [10-2](#)
- 
- ## V
- vendor-specific MIBs, adding [8-9](#)
  - verifyDb.sh tool [14-5](#)
  - voice services support [1-2](#)
    - PacketCable [1-2](#)
    - PacketCable, non-secure [1-2](#)
  - voice technology
    - about [1-2](#)
    - See also* PacketCable [1-2](#)
- 
- ## W
- watchdog agent alerts [A-5](#)

---

**X**

## xGCP

- configuring defaults [11-23](#)

- definition [1-5](#)

