Alcatel-Lucent OpenTouch™ Session Border Controler – R2.0

User's Manual

8AL90067USAAed01
April 2014

Table of Contents

1 *Overview* 14

Getting Started with Initial Connectivity 16

2 *Default OAMP IP Address* 17

3 *Installing the Software* 19

4 *Changing Default IP Address to Suit your Network Addressing Scheme* 21

5 *Licensing the Device* 23

Management Tools 25

6 *Introduction* 26

7 *Web-Based Management* 28

7.1 Getting Acquainted with the Web Interface 28

7.1.1 Computer Requirements 28

7.1.2 Accessing the Web Interface 28

7.1.3 Areas of the GUI 29

7.1.4 Toolbar Description 30

7.1.5 Navigation Tree 31

7.1.6 Working with Configuration Pages 33

7.1.7 Searching for Configuration Parameters 37

7.1.8 Creating a Login Welcome Message 38

7.1.9 Getting Help 39

7.1.10 Logging Off the Web Interface 40

7.2 Viewing the Home Page 41

7.3 Configuring Web User Accounts 42

7.3.1 Basic User Accounts Configuration 44

7.3.2 Advanced User Accounts Configuration 45

7.4 Displaying Login Information upon Login 49

7.5 Configuring Web Security Settings 49

7.6 Web Login Authentication using Smart Cards 50

7.7 Configuring Web and Telnet Access List 51

7.8 RADIUS Authentication 52

7.8.1 Setting Up a Third-Party RADIUS Server 52

7.8.2 Configuring RADIUS Authentication 53

7.8.3 Securing RADIUS Communication 55

7.8.4 Authenticating RADIUS in the URL 55

8 *CLI-Based Management* 56

8.1 Enabling CLI 56

8.1.1 Enabling Telnet for CLI 56

8.1.2 Enabling SSH with RSA Public Key for CLI 56

8.2 Establishing a CLI Session 58

8.3 Understanding Configuration Modes 59

8.4 Using CLI Shortcuts 59

8.5 Common CLI Commands 60

8.6 Configuring Tables in CLI 61

8.7 Understanding CLI Error Messages 62

8.8 Configuring Maximum Telnet/SSH Sessions 62

8.9 Viewing Current CLI Sessions 62

8.10 Terminating a User's CLI Session 63

8.11 Configuring Displayed Output Lines in CLI Terminal Window 63

9 *SNMP-Based Management* 65

9.1 Enabling SNMP and Configuring SNMP Community Strings 65
9.2 Configuring SNMP Trap Destinations 66
9.3 Configuring SNMP Trusted Managers 67
9.4 Configuring SNMP V3 Users 68

10 INI File-Based Management 70
10.1 INI File Format 70
10.1.1 Configuring Individual ini File Parameters 70
10.1.2 Configuring Table ini File Parameters 70
10.1.3 General ini File Formatting Rules 72
10.2 Configuring an ini File 72
10.3 Loading an ini File to the Device 72
10.4 Secured Encoded ini File 73
10.5 Configuring Password Display in ini File 74
10.6 INI Viewer and Editor Utility 74

General System Settings 75

11 Configuring Certificates 76
11.1 Replacing the Device's Certificate 76
11.2 Loading a Private Key 77
11.3 Mutual TLS Authentication 78
11.4 Self-Signed Certificates 79
11.5 TLS Server Certificate Expiry Check 79
11.6 Configuring Certificate Revocation Checking (OCSP) 80
11.7 Loading Certificate Chain for Trusted Root 81

12 Date and Time 82
12.1 Configuring Date and Time Manually 82
12.2 Automatic Date and Time through SNTP Server 82

General VoIP Configuration 86

13 Network 87
13.1 Configuring Physical Ethernet Ports 87
13.2 Configuring Tx/Rx for Ethernet Port-Pair Groups 88
13.3 Configuring Underlying Ethernet Devices 90
13.4 Configuring IP Network Interfaces 91
13.4.1 Assigning NTP Services to Application Types 95
13.4.2 Multiple Interface Table Configuration Summary and Guidelines 95
13.4.3 Networking Configuration Examples 96
13.5 Configuring Static IP Routes 99
13.5.1 Configuration Example of Static IP Routes 100
13.5.2 Static Route Table Configuration Guidelines 101
13.5.3 Troubleshooting the Routing Table 101
13.6 Configuring Quality of Service 102
13.7 Configuring ICMP Messages 104
13.8 DNS 105
13.8.1 Configuring the Internal DNS Table 105
13.8.2 Configuring the Internal SRV Table 106
13.9 Configuring NFS Settings 108
13.10 Network Address Translation Support 109
13.10.1 Device Located behind NAT 110
13.10.2 Remote UA behind NAT 112
13.11 Robust Receipt of Media Streams by Media Latching 114
13.12 Multiple Routers Support 115

14 Security 116
14.1 Configuring Firewall Settings 116
14.2 Configuring General Security Settings 120

14.3 Intrusion Detection System ... 120
14.3.1 Enabling IDS ... 121
14.3.2 Configuring IDS Policies ... 121
14.3.3 Assigning IDS Policies ... 125
14.3.4 Viewing IDS Alarms ... 126

15 Media ... 128
15.1 Configuring RTP/RTCP Settings ... 128
15.1.1 Configuring RTP Base UDP Port ... 128
15.2 Configuring Media (SRTP) Security ... 129

16 Services ... 132
16.1 SIP-based Media Recording ... 132
16.1.1 Enabling SIP-based Media Recording ... 134
16.1.2 Configuring SIP Recording Routing Rules ... 135
16.1.3 Configuring SIP User Part for SRS ... 136
16.1.4 Interworking SIP-based Media Recording with Third-Party Vendors ... 136
16.2 Routing Based on LDAP Active Directory Queries ... 137
16.2.1 Enabling the LDAP Service ... 137
16.2.2 Configuring the Device's LDAP Cache ... 138
16.2.3 Configuring LDAP Servers ... 139
16.3 Least Cost Routing ... 142
16.3.1 Overview ... 142
16.3.2 Configuring LCR ... 144
16.4 Configuring Call Setup Rules ... 148
16.4.1 Call Setup Rule Examples ... 151

17 Quality of Experience ... 155
17.1 Configuring the SEM Server ... 155
17.2 Configuring Quality of Experience Profiles ... 155
17.3 Configuring Bandwidth Profiles ... 159
17.4 Configuring Media Enhancement Profiles ... 161

18 Control Network ... 164
18.1 Configuring Media Realms ... 164
18.2 Configuring Remote Media Subnets ... 166
18.3 Configuring SRDs ... 169
18.4 Configuring SIP Interfaces ... 171
18.5 Configuring IP Groups ... 174
18.6 Configuring Proxy Sets ... 182

19 SIP Definitions ... 187
19.1 Configuring SIP Parameters ... 187
19.2 Configuring Registration Accounts ... 187
19.2.1 Regular Registration Mode ... 189
19.2.2 Single Registration for Multiple Phone Numbers using GIN ... 189
19.3 Configuring Proxy and Registration Parameters ... 190
19.3.1 SIP Message Authentication Example ... 192
19.4 Configuring SIP Message Manipulation ... 193
19.5 Configuring SIP Message Policy Rules ... 198

20 Coders and Profiles ... 201
20.1 Configuring IP Profiles ... 201

Session Border Controller Application ... 213

21 SBC Overview ... 214
21.1 SIP Network Definitions ... 215
21.2 SIP Dialog Initiation Process ... 215
21.3 User Registration ... 217
21.3.1 Initial Registration Request Processing ... 218

21.3.2 SBC Users Registration Database 218
21.3.3 Routing using Users Registration Database 219
21.3.4 Registration Refreshes 219
21.3.5 Registration Restriction Control 219
21.4 SBC Media Handling 220
21.4.1 Media Anchoring without Transcoding (Transparent) 221
21.4.2 No Media Anchoring 222
21.4.3 Restricting Coders 223
21.4.4 Prioritizing Coder List in SDP Offer 224
21.4.5 SRTP-RTP and SRTP-SRTP Transcoding 224
21.4.6 Multiple RTP Media Streams per Call Session 225
21.5 Limiting SBC Call Duration 225
21.6 SBC Authentication 225
21.6.1 SIP Authentication Server Functionality 225
21.6.2 User Authentication based on RADIUS 226
21.7 Interworking SIP Signaling 226
21.7.1 Interworking SIP 3xx Redirect Responses 227
21.7.2 Interworking SIP Diversion and History-Info Headers 228
21.7.3 Interworking SIP REFER Messages 229
21.7.4 Interworking SIP PRACK Messages 229
21.7.5 Interworking SIP Session Timer 230
21.7.6 Interworking SIP Early Media 230
21.7.7 Interworking SIP re-INVITE Messages 232
21.7.8 Interworking SIP UPDATE Messages 232
21.7.9 Interworking SIP re-INVITE to UPDATE 233
21.7.10 Interworking Delayed Offer 233
21.7.11 Interworking Call Hold 233
21.8 Call Survivability 234
21.8.1 Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server for Survivability 234
21.8.2 BroadSoft's Shared Phone Line Call Appearance for SBC Survivability 234
21.8.3 Call Survivability for Call Centers 236
21.8.4 Survivability Mode Display on Aastra IP Phones 238
21.9 Call Forking 238
21.9.1 Initiating SIP Call Forking 239
21.9.2 SIP Forking Initiated by SIP Proxy Server 239
21.9.3 Call Forking-based IP-to-IP Routing Rules 240
21.10 Alternative Routing on Detection of Failed SIP Response 240

22 SBC Configuration 241
22.1 Enabling the SBC Application 241
22.2 Configuring General Settings 241
22.2.1 Interworking Dialog Information in SIP NOTIFY Messages 242
22.3 Configuring Admission Control 244
22.4 Configuring Allowed Audio Coder Groups 247
22.5 Configuring Allowed Video Coder Groups 248
22.6 Routing SBC 248
22.6.1 Configuring Classification Rules 249
22.6.2 Configuring Message Condition Rules 254
22.6.3 Configuring SBC IP-to-IP Routing 255
22.6.4 Configuring SIP Response Codes for Alternative Routing Reasons 262
22.7 SBC Manipulations 264
22.7.1 Configuring IP-to-IP Inbound Manipulations 266
22.7.2 Configuring IP-to-IP Outbound Manipulations 268

High Availability System 274

23 HA Overview 275
23.1 Connectivity and Synchronization between Devices 276
23.2 Device Switchover upon Failure 276

23.3 HA Status on the Home Page 277

24 HA Configuration 279
24.1 Initial HA Configuration 279
24.1.1 Network Topology Types and Tx/Rx Ethernet Port Group Settings 279
24.1.2 Configuring the HA Devices 280
24.2 Configuration while HA is Operational 283
24.3 Configuring Firewall Allowed Rules 284
24.4 Monitoring IP Entity and HA Switchover upon Ping Failure 284

25 HA Maintenance 285
25.1 Maintenance of Redundant Device 286
25.2 Replacing a Failed Device 286
25.3 Forcing a Switchover 286
25.4 Software Upgrade 286
25.5 Rescue Options 286
25.5.1 Taking a Snapshot 286
25.5.2 Viewing Available Snapshots 287
25.5.3 Changing the Default Snapshot 287
25.5.4 Deleting a Snapshot 287
25.5.5 Manual Recovery 287
25.5.6 Automatic Recovery 290

Maintenance 291

26 Basic Maintenance 292
26.1 Resetting the Device 292
26.2 Remotely Resetting Device using SIP NOTIFY 293
26.3 Locking and Unlocking the Device 294
26.4 Saving Configuration 295

27 High Availability Maintenance 296
27.1 Initiating an HA Switchover 296
27.2 Resetting the Redundant Unit 296

28 Disconnecting Active Calls 297

29 Software Upgrade 299
29.1 Loading Auxiliary Files 299
29.1.1 Call Progress Tones File 300
29.1.2 Prerecorded Tones File 302
29.1.3 Dial Plan File 303
29.1.4 User Information File 305
29.2 Configuring the Product Key 308
29.3 Software License Key 309
29.3.1 Obtaining the Software License Key File 309
29.3.2 Installing the Software License Key 310
29.4 Software Upgrade Wizard 312
29.5 Backing Up and Loading Configuration File 316

30 Automatic Update Mechanism 318
30.1 Overview of Automatic Update Mechanism 318
30.1.1 Files Updated by Automatic Update 318
30.1.2 File Location for Automatic Update 318
30.1.3 Triggering Automatic Update 318
30.1.4 Querying Provisioning Server for Updated Files 319
30.2 Configuring Automatic Update 321
30.3 Automatic Configuration Methods 323
30.3.1 DHCP-based Configuration Server 323
30.3.2 HTTP-based Automatic Updates 324
30.3.3 Configuration using FTP or NFS 325

30.4 Loading Files Securely by Disabling TFTP 325

31 Restoring Factory Defaults 328
31.1 Restoring Defaults using CLI 328
31.2 Restoring Defaults using an ini File 329

Status, Performance Monitoring and Reporting 330

32 System Status 331
32.1 Viewing Device Information 331
32.2 Viewing Ethernet Port Information 331

33 Carrier-Grade Alarms 333
33.1 Viewing Active Alarms 333
33.2 Viewing Alarm History 333

34 Performance Monitoring 335
34.1 Viewing MOS per Media Realm 335
34.2 Viewing Quality of Experience 336
34.3 Viewing Average Call Duration 337

35 VoIP Status 339
35.1 Viewing Active IP Interfaces 339
35.2 Viewing Ethernet Device Status 339
35.3 Viewing Static Routes Status 339
35.4 Viewing Registered Users 340
35.5 Viewing Registration Status 341

36 Reporting Information to External Party 343
36.1 RTP Control Protocol Extended Reports (RTCP XR) 343
36.2 Generating Call Detail Records 346
36.2.1 Configuring CDR Reporting 346
36.2.2 CDR Field Description 347
36.3 Configuring RADIUS Accounting 350

Diagnostics 354

37 Syslog and Debug Recordings 355
37.1 Syslog Message Format 355
37.1.1 Event Representation in Syslog Messages 356
37.1.2 Identifying AudioCodes Syslog Messages using Facility Levels 357
37.1.3 SNMP Alarms in Syslog Messages 358
37.2 Configuring Syslog Settings 358
37.3 Configuring Debug Recording 359
37.4 Filtering Syslog Messages and Debug Recordings 360
37.4.1 Filtering IP Network Traces 362
37.5 Viewing Syslog Messages 362
37.6 Collecting Debug Recording Messages 363
37.7 Debug Capturing on Physical VoIP Interfaces 364

38 Enabling Debug File upon Device Crash 366

39 Testing SIP Signaling Calls 368
39.1 Configuring Test Call Endpoints 368
39.2 Starting and Stopping Test Calls 371
39.3 Viewing Test Call Statistics 372
39.4 Configuring DTMF Tones for Test Calls 373
39.5 Configuring SBC Test Call with External Proxy 374
39.6 Test Call Configuration Examples 375

Appendix 378

40 Dialing Plan Notation for Routing and Manipulation 379

41 Configuration Parameters Reference 381
41.1 Management Parameters 381
41.1.1 General Parameters 381
41.1.2 Web Parameters 382
41.1.3 Telnet Parameters 384
41.1.4 ini File Parameters 385
41.1.5 SNMP Parameters 385
41.1.6 Serial Parameters 387
41.1.7 Auxiliary and Configuration File Name Parameters 388
41.1.8 Automatic Update Parameters 389
41.2 Networking Parameters 391
41.2.1 Ethernet Parameters 391
41.2.2 Multiple VoIP Network Interfaces and VLAN Parameters 392
41.2.3 Routing Parameters 393
41.2.4 Quality of Service Parameters 393
41.2.5 NAT Parameters 394
41.2.6 NFS Parameters 395
41.2.7 DNS Parameters 395
41.2.8 DHCP Parameters 396
41.2.9 NTP and Daylight Saving Time Parameters 397
41.3 Debugging and Diagnostics Parameters 399
41.3.1 General Parameters 399
41.3.2 SIP Test Call Parameters 399
41.3.3 Syslog, CDR and Debug Parameters 400
41.3.4 Resource Allocation Indication Parameters 404
41.3.5 HA Ping Parameters 404
41.4 Security Parameters 406
41.4.1 General Security Parameters 406
41.4.2 HTTPS Parameters 407
41.4.3 SRTP Parameters 408
41.4.4 TLS Parameters 410
41.4.5 SSH Parameters 412
41.4.6 IDS Parameters 413
41.4.7 OCSP Parameters 414
41.4.8 RADIUS Parameters 414
41.5 Quality of Experience Parameters 416
41.6 Control Network Parameters 419
41.6.1 IP Group, Proxy, Registration and Authentication Parameters 419
41.6.2 Network Application Parameters 425
41.7 General SIP Parameters 427
41.8 Coders and Profile Parameters 441
41.9 Channel Parameters 442
41.9.1 RTP, RTCP and T.38 Parameters 442
41.10 SBC Parameters 444
41.11 Services 453
41.11.1 SIP-based Media Recording Parameters 453
41.11.2 LDAP Parameters 454
41.11.3 Least Cost Routing Parameters 455
41.11.4 Call Setup Rules Parameters 456

42 SBC and DSP Channel Capacity 457
42.1 Signaling-Media Sessions & User Registrations 457
42.2 Mediant Server Edition 458
42.3 Mediant Virtual Edition SBC 458

43 Technical Specifications 459
43.1 Technical Specifications 459

Reader's Notes

Notice

This document describes AudioCodes Mediant Server Edition and Mediant Virtual Edition Session Border Controllers (SBCs).

Information contained in this document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, AudioCodes cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Before consulting this document, check the corresponding Release Notes regarding feature preconditions and/or specific support in this release. In cases where there are discrepancies between this document and the Release Notes, the information in the Release Notes supersedes that in this document. Updates to this document and other documents as well as software files can be downloaded by registered customers at http://www.audiocodes.com/downloads.

This document is subject to change without notice.

Date Published: December-17-2013

Trademarks

AudioCodes, AC, AudioCoded, Ardito, CTI2, CTI², CTI Squared, HD VoIP, HD VoIP Sounds Better, InTouch, IPmedia, Mediant, MediaPack, NetCoder, Netrake, Nuera, Open Solutions Network, OSN, Stretto, TrunkPack, VMAS, VoicePacketizer, VoIPerfect, VoIPerfectHD, What's Inside Matters, Your Gateway To VoIP and 3GX are trademarks or registered trademarks of AudioCodes Limited. All other products or trademarks are property of their respective owners. Product specifications are subject to change without notice.

WEEE EU Directive

Pursuant to the WEEE EU Directive, electronic and electrical waste must not be disposed of with unsorted waste. Please contact your local recycling authority for disposal of this product.

Customer Support

Customer technical support and service are generally provided by AudioCodes' Distributors, Partners, and Resellers from whom the product was purchased. For technical support for products purchased directly from AudioCodes, or for customers subscribed to AudioCodes Customer Technical Support (ACTS), contact support@audiocodes.com.

Abbreviations and Terminology

Each abbreviation, unless widely used, is spelled out in full when first used.

Related Documentation

Manual Name
SIP CPE Release Notes
Mediant Server Edition SBC Installation Manual
Mediant Virtual Edition SBC Installation Manual
Complementary Guides
CLI Reference Guide
SNMP User's Guide
SBC Design Guide
Recommended Security Guidelines Configuration Note
SIP Message Manipulations Quick Reference Guide
SAS Application Configuration Guide
Utility Guides
INI Viewer & Editor Utility User's Guide
AcBootP Utility User's Guide

Note and Warnings

Note: The scope of this document does not fully cover security aspects for deploying the device in your environment. Security measures should be done in accordance with your organization's security policies. For basic security guidelines, refer to AudioCodes *Recommended Security Guidelines* document.

Note: Throughout this manual, unless otherwise specified, the term *device* refers to your AudioCodes products.

Note: Before configuring the device, ensure that it is installed correctly as instructed in the *Hardware Installation Manual*.

Note: This device is considered an INDOOR unit and therefore, must be installed only indoors.

Notes:

- By default, the device supports export-grade (40-bit and 56-bit) encryption due to US government restrictions on the export of security technologies. To enable 128-bit and 256-bit encryption on your device, contact your AudioCodes sales representative.
- This device includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).
- This device includes cryptographic software written by Eric Young (eay@cryptsoft.com).

Note: Some of the features listed in this document are available only if the relevant Software License Key has been purchased from AudioCodes and installed on the device. For a list of Software License Keys that can be purchased, please consult your AudioCodes sales representative.

Note: OPEN SOURCE SOFTWARE. Portions of the software may be open source software and may be governed by and distributed under open source licenses, such as the terms of the GNU General Public License (GPL), the terms of the Lesser General Public License (LGPL), BSD and LDAP, which terms are located at: http://www.audiocodes.com/support and all are incorporated herein by reference. If any open source software is provided in object code, and its accompanying license requires that it be provided in source code as well, Buyer may receive such source code by contacting AudioCodes, by following the instructions available on AudioCodes website.

Documentation Feedback

AudioCodes continually strives to produce high quality documentation. If you have any comments (suggestions or errors) regarding this document, please fill out the Documentation Feedback form on our Web site at http://www.audiocodes.com/downloads.

Reader's Notes

1 Overview

AudioCodes Mediant Software Enterprise Session Border Controllers (E-SBC) are pure-software products, enabling connectivity and security between Enterprises' and Service Providers' VoIP networks. The Mediant Software product line include the following product variants:

- **Mediant Server Edition SBC:** x86 server-based platform, which must be installed on a server that complies to the specified hardware requirements (see "Technical Specifications" on page 458 or refer to the *Mediant Server Edition SBC Installation Manual*)
- **Mediant Virtual Edition SBC:** Installed and hosted in a virtual machine environment that complies to specified requirements (see "Technical Specifications" on page 458 or refer to the *Mediant Virtual Edition SBC Installation Manual*)

These devices provide perimeter defense for protecting companies from malicious VoIP attacks; voice and signaling mediation and normalization for allowing the connection of any PBX and/or IP-PBX to any Service Provider; and service assurance for service quality and manageability. The device offers call "survivability", ensuring service continuity to enterprises served by a centralized SIP-based IP-Centrex server or branch offices of distributed enterprises. Survivability functionality enables internal office communication between SIP clients in the case of disconnection from the centralized SIP IP-Centrex server or IP-PBX.

The device offers multiple local and remote management platforms, including HTTP/S-based Web server, command-line interface (CLI), and SNMP.

Note: For maximum call capacity figures, see "SBC and DSP Channel Capacity" on page 456.

Reader's Notes

Part I

Getting Started with Initial Connectivity

2 Default OAMP IP Address

The device is shipped with a factory default IP address for operations, administration, maintenance, and provisioning (OAMP), through its VoIP LAN interface. You can use this address to initially access the device from any of its management tools (embedded Web server, EMS, or Telnet/SSH). You can also access the device through the console CLI, by connecting the device's serial (RS-232) port to a PC.

The table below lists the device's default IP address.

Table 1: Default VoIP LAN IP Address for OAMP

IP Address	Value
Application Type	OAMP + Media + Control
IP Address	192.168.0.1
Prefix Length	255.255.255.0 (24)
Underlying Device	1

Reader's Notes

3 Installing the Software

For installing the device, refer to the following documents:

- **Mediant Server Edition SBC:** *Mediant Server Edition SBC Installation Manual*
- **Mediant Virtual Edition SBC:** *Mediant Virtual Edition SBC Installation Manual*

Reader's Notes

4 Changing Default IP Address to Suit your Network Addressing Scheme

After initial installation, the device is assigned with the following default IP address:

- **IP Address:** 192.168.0.1
- **Subnet Mask:** 255.255.255.0

You can change this default IP address to suit your network addressing scheme. Once done, you can connect to the device's Web-based management tool (*Web interface*) using this new IP address.

> **Note:** The Server Edition orders available NICs in alphabetical order of corresponding MAC addresses. If, however, the device identifies an on-board NIC, it selects it first even if external NICs' MAC addresses precede it alphabetically.

➢ To change the IP address using CLI:

1. Establish a CLI session with the device:
   - **Server Edition:** Use a VGA monitor and keyboard to connect to the CLI management interface.
   - **Virtual Edition:** Click the VM's **Console** tab to connect to the CLI management interface.
2. At the CLI prompt, type the username (default is "Admin" - case sensitive), and then press Enter:

```
Username: Admin
```

3. At the prompt, type the password (default is "Admin" - case sensitive), and then press Enter:

```
Password: Admin
```

The following prompt appears:

```
Welcome to AudioCodes CLI
Username: Admin
Password:
Mediant SW>
```

4. At the prompt, type the following, and then press Enter:

```
# enable
```

5. At the prompt, type the password, and then press Enter:

```
Password: Admin
```

6. At the prompt, type the following commands to access the network interface configuration:

```
# configure voip
(config-voip)# interface network-if 0
(network-if-0)#
```

> **Note:** To ensure that you type the correct command syntax, use the Tab key to auto-complete partially entered commands.

7. At the prompt, type the following commands to configure the IP address, prefix length and default gateway:

```
(network-if-0)# ip-address <new IP address, e.g. 10.4.212.155>
(network-if-0)# prefix-length <prefix length, e.g., 16>
(network-if-0)# gateway <default gateway IP address, e.g., 10.4.0.1>
```

8. If the device is connected to an IP network that uses VLAN ID, type the following command to configure it:

```
(network-if-0)# vlan-id <VLAN, e.g., 10>
```

9. At the prompt, type the following commands to complete configuration:

```
(network-if-0)# exit
(config-voip)# exit
```

10. At the prompt, make sure that Port #1 is connected (i.e., link is UP) using the **show voip ports** command. This port is mapped to network-if-0, by default. For more information on mapping physical ports to the logical configuration ports, see "Configuring Tx/Rx for Ethernet Port-Pair Groups" on page 88.
11. At the prompt, type the following to reset the device and activate the new configuration:

```
# reload now
```

Once you have assigned an IP address that suits your network environment, you can connect remotely with this IP address to the device's Web interface for management and configuration. To access the Web interface, see "Web-Based Management" on page 27.

For initial setup, it is recommended to configure the following network settings:

- To modify and configure IP network interfaces, see "Configuring IP Network Interface" on page 91
- To configure the used physical Ethernet ports (Native VLAN, speed, and mode), see "Configuring Physical Ethernet Ports" on page 87.

5 Licensing the Device

By default, the device is shipped with a pre-installed Software License Key that enables **only one** call session. After installation has completed successfully, contact your AudioCodes sales representative and provide your Product Key and installation Serial Number in order to obtain a Software License Key file to enable the call capacity and features that you ordered. For loading a Software License Key to the device, see "Software License Key" on page 309.

Reader's Notes

Part II

Management Tools

6 Introduction

This part provides an overview of the various management tools that can be used to configure the device. It also provides step-by-step procedures on how to configure these management tools.

The device provides the following management tools:

- Embedded HTTP/S-based Web server - see "Web-based Management" on page 27
- Command Line Interface (CLI) - see "CLI-Based Management" on page 55
- Simple Network Management Protocol (SNMP) - see "SNMP-Based Management" on page 64
- Configuration *ini* file - see "INI File-Based Management" on page 69

Notes:

- Some configuration settings can only be done using a specific management tool. For example, some configuration can only be done using the Configuration *ini* file method.
- Throughout this manual, whenever a parameter is mentioned, its corresponding Web, CLI, and ini file parameter is mentioned. The *ini* file parameters are enclosed in square brackets [...].
- For a list and description of all the configuration parameters, see "Configuration Parameters Reference" on page 380.

Reader's Notes

7 Web-Based Management

The device provides an embedded Web server (hereafter referred to as *Web interface*), supporting fault management, configuration, accounting, performance, and security (FCAPS), including the following:

- Full configuration
- Software and configuration upgrades
- Loading auxiliary files, for example, the Call Progress Tones file
- Real-time, online monitoring of the device, including display of alarms and their severity
- Performance monitoring of voice calls and various traffic parameters

The Web interface provides a user-friendly, graphical user interface (GUI), which can be accessed using any standard Web browser (e.g., Microsoft™ Internet Explorer).

Access to the Web interface is controlled by various security mechanisms such as login user name and password, read-write privileges, and limiting access to specific IP addresses.

Notes:

- The Web interface allows you to configure most of the device's settings. However, additional configuration parameters may exist that are not available in the Web interface and which can only be configured using other management tools.
- Some Web interface pages and/or parameters are available only for certain hardware configurations or software features. The software features are determined by the installed Software License Key (see "Software License Key" on page 309).

7.1 Getting Acquainted with the Web Interface

This section provides a description of the Web interface.

7.1.1 Computer Requirements

The client computer requires the following to work with the Web interface of the device:

- A network connection to the device
- One of the following Web browsers:
  - Microsoft™ Internet Explorer™ (Version 6.0 and later)
  - Mozilla Firefox® (Versions 5 through 9.0)
- Recommended screen resolutions: 1024 x 768 pixels, or 1280 x 1024 pixels

Note: Your Web browser must be JavaScript-enabled to access the Web interface.

7.1.2 Accessing the Web Interface

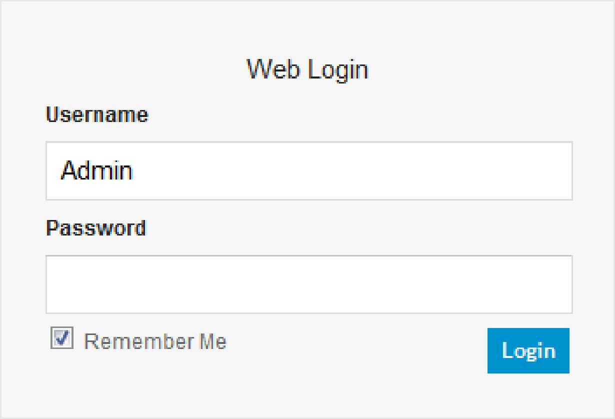
The procedure below describes how to access the Web interface.

➢ To access the Web interface:

1. Open a standard Web browser (see "Computer Requirements" on page 28).
2. In the Web browser, specify the OAMP IP address of the device (e.g., http://10.1.10.10); the Web

interface's Login window appears, as shown below:

Figure 1: Web Login Screen



Web Login

Username: Admin

Password

☑ Remember Me

3. In the 'Username' and 'Password' fields, enter the case-sensitive, user name and password respectively.
4. Click **Login**; the Web interface is accessed, displaying the Home page. For a detailed description of the Home page, see "Viewing the Home Page" on page 41.

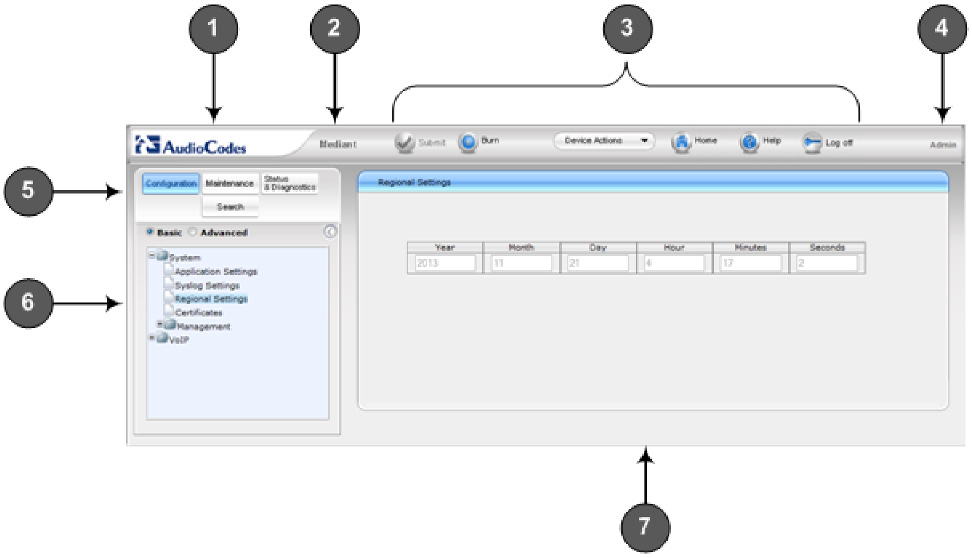
Notes:

- By default, Web access is only through the IP address of the OAMP interface. However, you can allow access from all of the device's IP network interfaces, by setting the EnableWebAccessFromAllInterfaces parameter to 1.
- The default login username and password is "Admin". To change the login credentials, see "Configuring the Web User Accounts" on page 42.
- If you want the Web browser to remember your password, select the 'Remember Me' check box and then agree to the browser's prompt (depending on your browser) to save the password for future logins. On your next login attempt, simply press the Tab or Enter keys to auto-fill the 'Username' and 'Password' fields, and then click **Login**.
- Depending on your Web browser's settings, a security warning box may be displayed. The reason for this is that the device's certificate is not trusted by your PC. The browser may allow you to install the certificate, thus skipping the warning box the next time you connect to the device. If you are using Windows Internet Explorer, click **View Certificate**, and then **Install Certificate**. The browser also warns you if the host name used in the URL is not identical to the one listed in the certificate. To resolve this, add the IP address and host name (ACL_nnnnnn, where *nnnnnn* is the serial number of the device) to your hosts file, located at /etc/hosts on UNIX or C:\Windows\System32\Drivers\ETC\hosts on Windows; then use the host name in the URL (e.g., https://ACL_280152). Below is an example of a host file:
  ```
  127.0.0.1    localhost
  10.31.4.47   ACL_280152
  ```

7.1.3 Areas of the GUI

The areas of the Web interface's GUI are shown in the figure below and described in the subsequent table.

Figure 2: Main Areas of the Web Interface GUI



Year	Month	Day	Hour	Minutes	Seconds
2013	11	21	4	17	2

Table 2: Description of the Web GUI Areas

Item #	Description
1	AudioCodes company logo.
2	Product name.
3	Toolbar, providing frequently required command buttons. For more information, see "Toolbar Description" on page 30.
4	Displays the username of the Web user that is currently logged in.
5	Navigation bar, providing the following tabs for accessing various functionalities in the Navigation tree: ▪ **Configuration**, **Maintenance**, and **Status & Diagnostics** tabs: Access the configuration menus (see "Working with Configuration Pages" on page 33) ▪ **Search** tab: Enables a search engine for searching configuration parameters (see "Searching for Configuration Parameters" on page 37)
6	Navigation tree, displaying a tree-like structure of elements (configuration menus or search engine) pertaining to the selected tab on the Navigation bar. For more information, see "Navigation Tree" on page 31.
7	Work pane, displaying the configuration page of the selected menu in the Navigation tree. This is where configuration is done. For more information, see "Working with Configuration Pages" on page 33.

7.1.4 Toolbar Description

The toolbar provides frequently required command buttons, described in the table below:

Table 3: Description of Toolbar Buttons

Icon	Button Name	Description
✓	**Submit**	Applies parameter settings to the device (see "Saving Configuration" on page 295). **Note:** This icon is grayed out when not applicable to the currently opened page.

Icon	Button Name	Description
◉	**Burn**	Saves parameter settings to flash memory (see "Saving Configuration" on page 295).
Device Actions ▾	**Device Actions**	Opens a drop-down list with frequently needed commands: ▪ **Load Configuration File:** Opens the Configuration File page for loading an *ini* file to the device (see "Backing Up and Loading Configuration File" on page 316). ▪ **Save Configuration File:** Opens the Configuration File page for saving the *ini* file to a folder on your PC (see "Backing Up and Loading Configuration File" on page 316). ▪ **Reset:** Opens the Maintenance Actions page for performing various maintenance procedures such as resetting the device (see "Resetting the Device" on page 292). ▪ **Software Upgrade Wizard:** Starts the Software Upgrade Wizard for upgrading the device's software (see "Software Upgrade Wizard" on page 312). ▪ **Switch Over:** Opens the High Availability Maintenance page for switching between Active and Redundant devices (see High Availability Maintenance on page 295). ▪ **Reset Redundant:** Opens the High Availability Maintenance page for resetting the Redundant device (see High Availability Maintenance on page 295).
🏠	**Home**	Opens the Home page (see "Viewing the Home Page" on page 41).
?	**Help**	Opens the Online Help topic of the currently opened configuration page (see "Getting Help" on page 39).
🔑	**Log off**	Logs off a session with the Web interface (see "Logging Off the Web Interface" on page 40).
-	**Reset**	If you modify a parameter on a page that takes effect only after a device reset, after you click the **Submit** button, the toolbar displays "Reset". This is a reminder that you need to later save your settings to flash memory and reset the device.

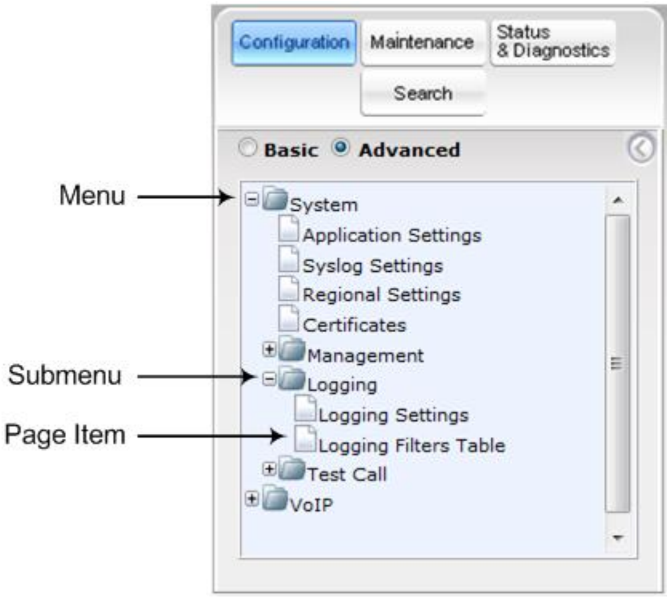
7.1.5 Navigation Tree

The Navigation tree is located in the Navigation pane and displays a tree-like structure of menus pertaining to the selected tab on the Navigation bar. You can drill-down to the required page item level to open its corresponding page in the Work pane.

The terminology used throughout this manual for referring to the hierarchical structure of the tree is as follows:

- *Menu*: first level (highest level)
- *Submenu*: second level - contained within a menu
- *Page item*: last level (lowest level in a menu) - contained within a menu or submenu

Figure 3: Navigating in Hierarchical Menu Tree (Example)



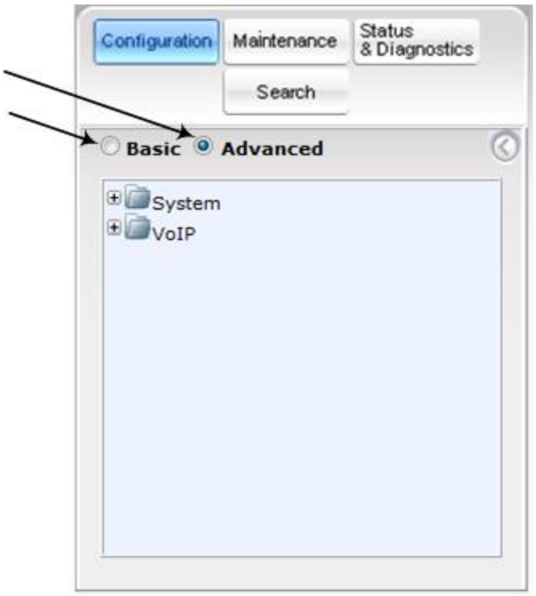
Note: The figure above is used only as an example. The displayed menus depend on supported features based on the Software License Key installed on your device.

7.1.5.1 *Displaying Navigation Tree in Basic and Full View*

You can view an expanded or reduced display of the Navigation tree. This affects the number of displayed menus and submenus in the tree. The expanded view displays all the menus pertaining to the selected configuration tab; the reduced view displays only commonly used menus.

- To display a reduced menu tree, select the **Basic** option (default).
- To display all menus and submenus, select the **Advanced** option.

Figure 4: Basic and Full View Options



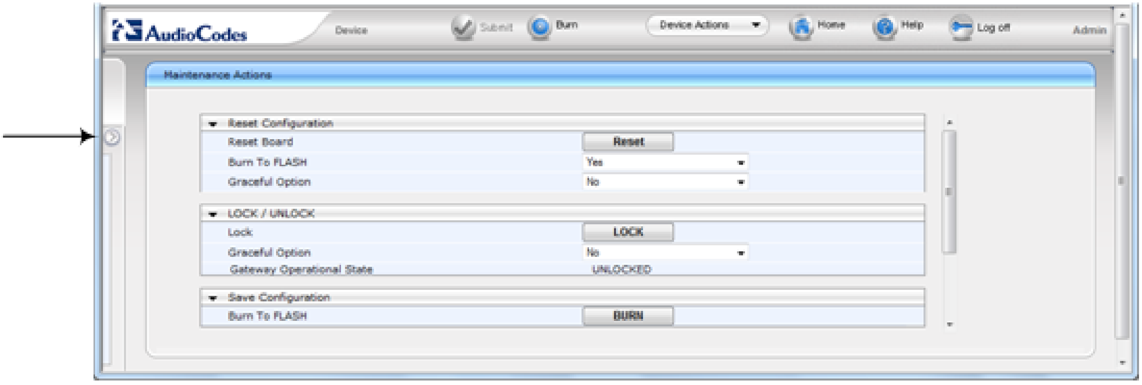
> **Note:** After you reset the device, the Web GUI is displayed in **Basic** view.

7.1.5.2 Showing / Hiding the Navigation Pane

You can hide the Navigation pane to provide more space for elements displayed in the Work pane. This is especially useful when the Work pane displays a wide table. The arrow button located below the Navigation bar is used to hide and show the pane.

- To hide the Navigation pane, click the left-pointing arrow ◀ ; the pane is hidden and the button is replaced by the right-pointing arrow button.
- To show the Navigation pane, click the right-pointing arrow ▶ ; the pane is displayed and the button is replaced by the left-pointing arrow button.

Figure 5: Show and Hide Button (Navigation Pane in Hide View)



Reset Configuration
Reset Board
Burn To FLASH: Yes
Graceful Option: No
LOCK / UNLOCK
Lock
Graceful Option: No
Gateway Operational State: UNLOCKED
Save Configuration
Burn To FLASH

7.1.6 Working with Configuration Pages

The configuration pages contain the parameters for configuring the device and are displayed in the Work pane.

7.1.6.1 Accessing Pages

The configuration pages are accessed by clicking the required page item in the Navigation tree.

➢ To open a configuration page:

1. On the Navigation bar, click the required tab (**Configuration**, **Maintenance**, or **Status & Diagnostics**); the menus pertaining to the selected tab appear in the Navigation tree.
2. Navigate to the required page item, by performing the following:
   - Drill-down using the **plus** ⊞ sign to expand the menu and submenus.
   - Drill-up using the **minus** ⊟ sign to collapse the menu and submenus.
3. Click the required page item; the page opens in the Work pane.

You can also access previously opened pages by clicking the Web browser's **Back** button until you have reached the required page. This is useful if you want to view pages in which you have performed configurations in the current Web session.

Note: Depending on the access level of your Web user account, certain pages may not be accessible or may be read-only (see "Configuring Web User Accounts" on page 42). If a page is read-only, "Read-Only Mode" is displayed at the bottom of the page.

7.1.6.2 *Viewing Parameters*

Some pages allow you to view a reduced or expanded display of parameters. The Web interface provides two methods for displaying page parameters:

- Displaying "basic" and "advanced" parameters - see "Displaying Basic and Advanced Parameters" on page 34
- Displaying parameter groups - see "Showing / Hiding Parameter Groups" on page 34

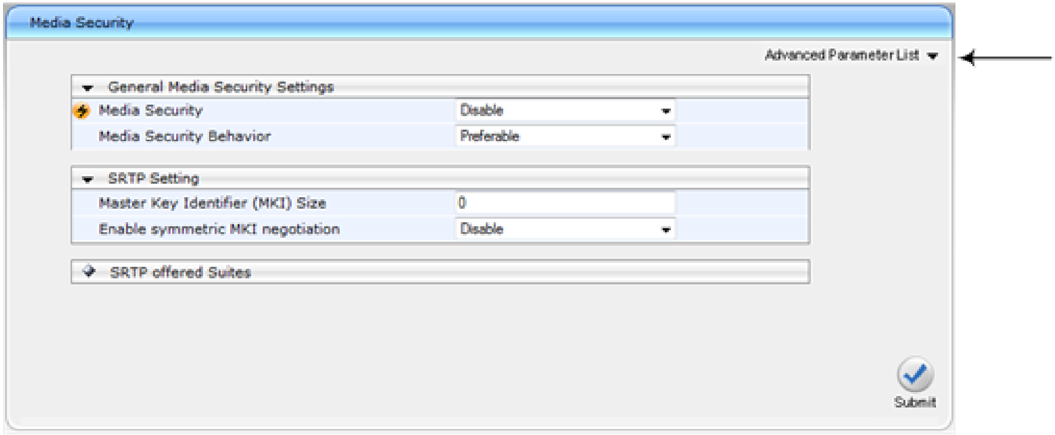
7.1.6.2.1 Displaying Basic and Advanced Parameters

Some pages provide a toggle button that allows you to show and hide parameters. This button is located on the top-right corner of the page and has two display states:

- **Advanced Parameter List** button with down-pointing arrow: click this button to display all parameters.
- **Basic Parameter List** button with up-pointing arrow: click this button to show only common (*basic*) parameters.

The figure below shows an example of a page displaying basic parameters only. If you click the **Advanced Parameter List** button (shown below), the page will also display the advanced parameters.

Figure 6: Toggling between Basic and Advanced View



Media Security

Advanced Parameter List

General Media Security Settings
Media Security: Disable
Media Security Behavior: Preferable

SRTP Setting
Master Key Identifier (MKI) Size: 0
Enable symmetric MKI negotiation: Disable

SRTP offered Suites

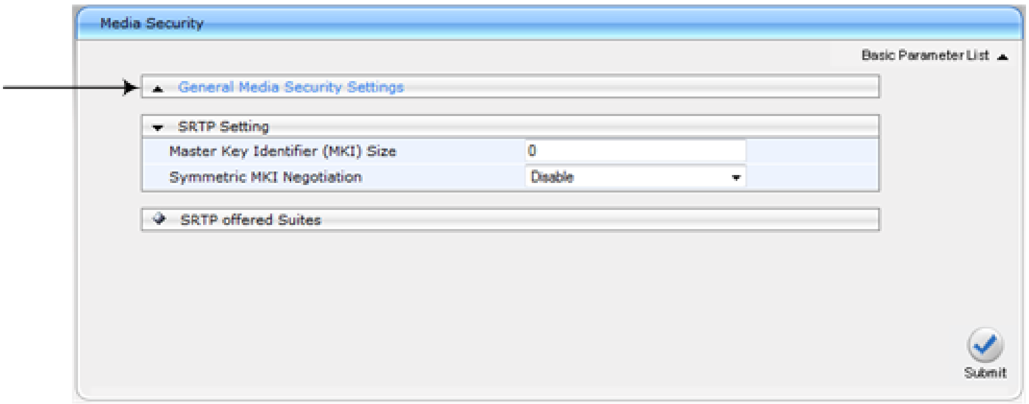
Notes:

- When the Navigation tree is in **Advanced** display mode (see "Navigation Tree" on page 31), configuration pages display all their parameters.
- If you reset the device, the Web pages display only the basic parameters.
- The basic parameters are displayed in a different background color to the advanced parameters.

7.1.6.2.2 Showing / Hiding Parameter Groups

Some pages group parameters under sections, which can be hidden or shown. To toggle between hiding and showing a group, simply click the group title name that appears above each group. The button appears with a down-pointing or up-pointing arrow, indicating that it can be collapsed or expanded when clicked, respectively.

Figure 7: Expanding and Collapsing Parameter Groups



Media Security

Basic Parameter List ▲

▲ General Media Security Settings

▼ SRTP Setting

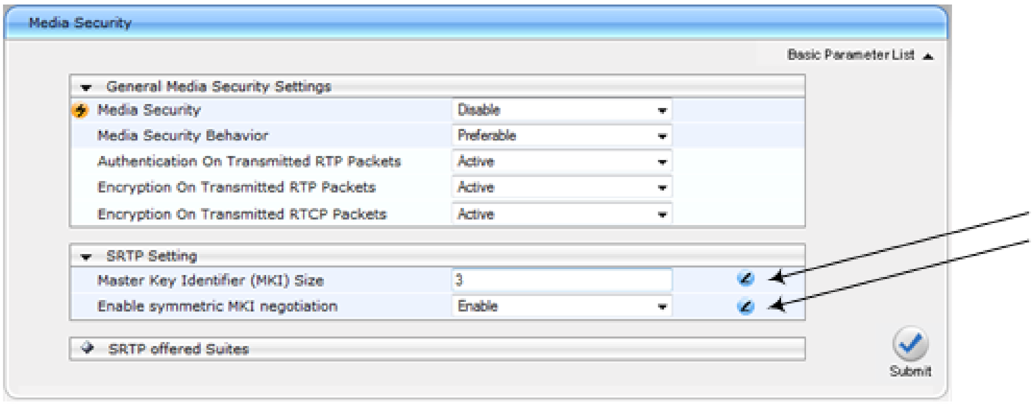
Master Key Identifier (MKI) Size	0
Symmetric MKI Negotiation	Disable

SRTP offered Suites

7.1.6.3 *Modifying and Saving Parameters*

When you modify a parameter value on a page, the **Edit** ✎ icon appears to the right of the parameter. This indicates that the parameter has been modified, but has yet to be applied (submitted). After you click **Submit** the ✎ icon disappears.

Figure 8: Edit Symbol after Modifying Parameter Value



Media Security

Basic Parameter List ▲

▼ General Media Security Settings

⚡ Media Security	Disable
Media Security Behavior	Preferable
Authentication On Transmitted RTP Packets	Active
Encryption On Transmitted RTP Packets	Active
Encryption On Transmitted RTCP Packets	Active

▼ SRTP Setting

Master Key Identifier (MKI) Size	3 ✎
Enable symmetric MKI negotiation	Enable ✎

SRTP offered Suites

➢ To save configuration changes on a page to the device's volatile memory (RAM):

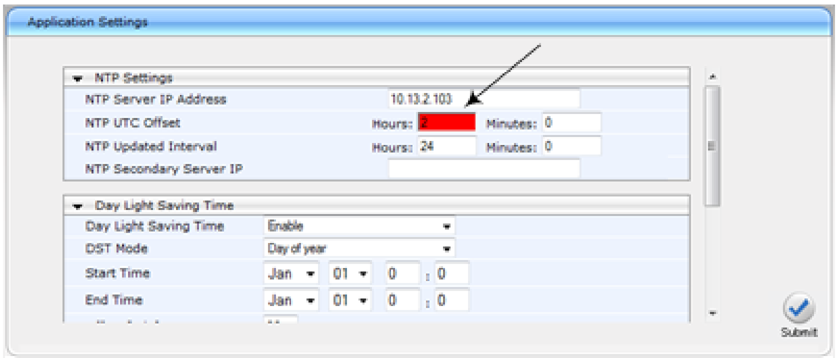
- On the toolbar, click the **Submit** ✓ button.
- At the bottom of the page, click the **Submit** ✓ button.

When you click **Submit**, modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect. Parameters displayed on the page with the lightning ⚡ icon take effect only after a device reset. For resetting the device, see "Resetting the Device" on page 292.

⚠ **Note:** Parameters saved to the volatile memory (by clicking **Submit**), revert to their previous settings after a hardware or software reset, or if the device is powered down. Thus, to ensure parameter changes (whether on-the-fly or not) are retained, save ('burn') them to the device's non-volatile memory, i.e., flash (see "Saving Configuration" on page 295).

If you enter an invalid parameter value (e.g., not in the range of permitted values) and then click **Submit**, a message box appears notifying you of the invalid value. In addition, the parameter value reverts to its previous value and is highlighted in red, as shown in the figure below:

Figure 9: Value Reverts to Previous Valid Value



Application Settings

NTP Settings

- NTP Server IP Address: 10.13.2.103
- NTP UTC Offset: Hours: 2 Minutes: 0
- NTP Updated Interval: Hours: 24 Minutes: 0
- NTP Secondary Server IP:

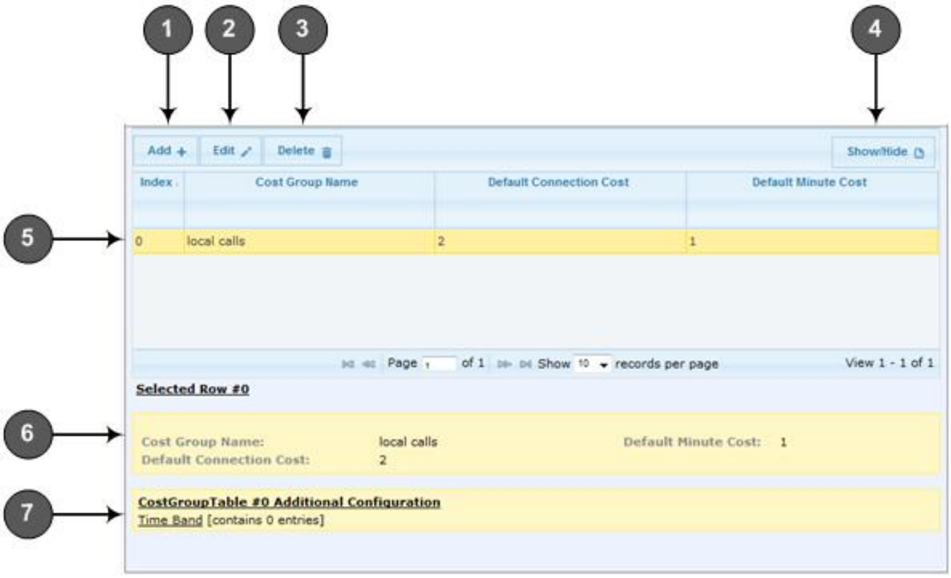
Day Light Saving Time

- Day Light Saving Time: Enable
- DST Mode: Day of year
- Start Time: Jan 01 0 : 0
- End Time: Jan 01 0 : 0

7.1.6.4 *Working with Tables*

Many of the Web configuration pages provide tables for configuring various functionalities of the device. The figure below and subsequent table describe the areas of a typical configuration table:

Figure 10: Displayed Details Pane



Index	Cost Group Name	Default Connection Cost	Default Minute Cost
0	local calls	2	1

Page 1 of 1 Show 10 records per page View 1 - 1 of 1

Selected Row #0

Cost Group Name: local calls
Default Connection Cost: 2
Default Minute Cost: 1

CostGroupTable #0 Additional Configuration
Time Band [contains 0 entries]

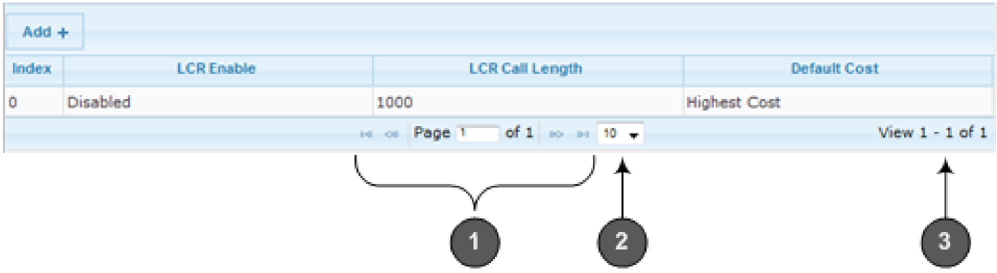
Table 4: Enhanced Table Design Description

Item #	Button	
1	**Add**	Adds a new index entry row to the table. When you click this button, a dialog box appears with parameters for configuring the new entry. When you have completed configuration, click the **Submit** button in the dialog box to add it to the table.
2	**Edit**	Edits the selected row.
3	**Delete**	Removes the selected row from the table. When you click this button, a confirmation box appears requesting you to confirm deletion. Click **Delete** to accept deletion.
4	**Show/Hide**	Toggles between displaying and hiding the full configuration of a selected row. This configuration is displayed below the table (see Item #6) and is useful for large tables that cannot display all its columns in the work pane.

Item #	Button	
5	-	Selected index row entry for editing, deleting and showing configuration.
6	-	Displays the full configuration of the selected row when you click the **Show/Hide** button.
7	-	Links to access additional configuration tables related to the current configuration.

You can also define the number of rows to display on the page and to navigate between pages displaying multiple rows. This is done using the page navigation area located below the table, as shown in the figure below:

Figure 11: Viewing Table Rows per Page



Index	LCR Enable	LCR Call Length	Default Cost
0	Disabled	1000	Highest Cost

Table 5: Row Display and Page Navigation

Item #	Description
1	Defines the page that you want to view. Enter the required page number or use the following page navigation buttons: ▪ ▶ - Displays the next page ▪ ▶| - Displays the last page ▪ ◀ - Displays the previous page ▪ |◀ - Displays the first page
2	Defines the number of rows to display per page. You can select 5 or 10, where the default is 10.
3	Displays the currently displayed page number.

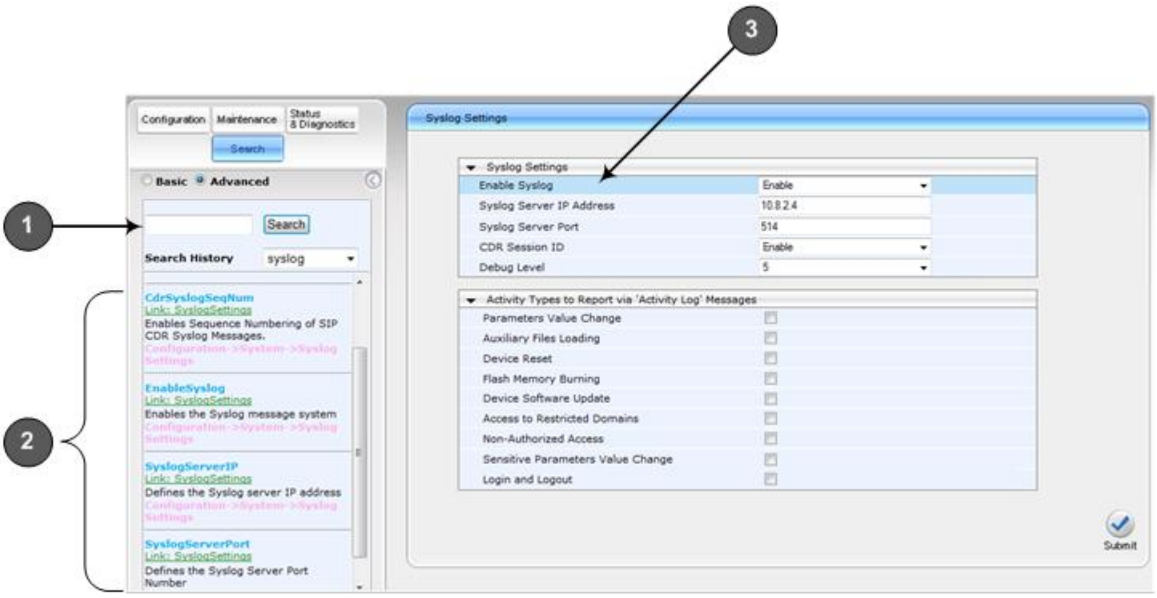
7.1.7 Searching for Configuration Parameters

You can locate the exact Web page on which a specific parameter appears, by using the Search feature. To search for a Web parameter, you must use the *ini* file parameter name as the search key. The search key can include the full parameter name (e.g., "EnableSyslog") or a substring of it (e.g., "sys"). If you search for a substring, all parameters containing the specified substring in their names are listed in the search result.

➢ To search for a parameter:

1. On the Navigation bar, click the **Search** tab; the Search engine appears in the Navigation pane.
2. In the field alongside the **Search** button, enter the parameter name or a substring of the name for which you want to search. If you have done a previous search for such a parameter, instead of entering the required string, you can use the 'Search History' drop-down list to select the string saved from a previous search.
3. Click **Search**; a list of found parameters based on your search key appears in the Navigation pane. Each searched result displays the following:
   - *ini* file parameter name
   - Link (in green) to the Web page on which the parameter appears
   - Brief description of the parameter
   - Menu navigation path to the Web page on which the parameter appears
4. In the searched list, click the required parameter (green link) to open the page on which the parameter appears; the relevant page opens in the Work pane and the searched parameter is highlighted in the page for easy identification, as shown in the figure below:

Figure 12: Searched Result Screen



○ Basic ● Advanced

Search History: syslog

CdrSyslogSeqNum
Link: SyslogSettings
Enables Sequence Numbering of SIP CDR Syslog Messages.
Configuration->System->Syslog Settings

EnableSyslog
Link: SyslogSettings
Enables the Syslog message system
Configuration->System->Syslog Settings

SyslogServerIP
Link: SyslogSettings
Defines the Syslog server IP address
Configuration->System->Syslog Settings

SyslogServerPort
Link: SyslogSettings
Defines the Syslog Server Port Number

Syslog Settings

Syslog Settings

Enable Syslog	Enable
Syslog Server IP Address	10.8.2.4
Syslog Server Port	514
CDR Session ID	Enable
Debug Level	5

Activity Types to Report via 'Activity Log' Messages

Parameters Value Change	□
Auxiliary Files Loading	□
Device Reset	□
Flash Memory Burning	□
Device Software Update	□
Access to Restricted Domains	□
Non-Authorized Access	□
Sensitive Parameters Value Change	□
Login and Logout	□

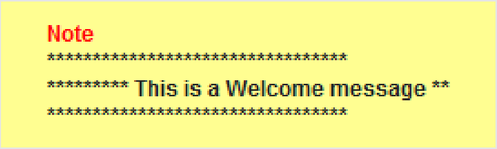
Table 6: Search Description

Item #	Description
1	Search field for entering search key and **Search** button for activating the search process.
2	Search results listed in Navigation pane.
3	Found parameter, highlighted on relevant Web page

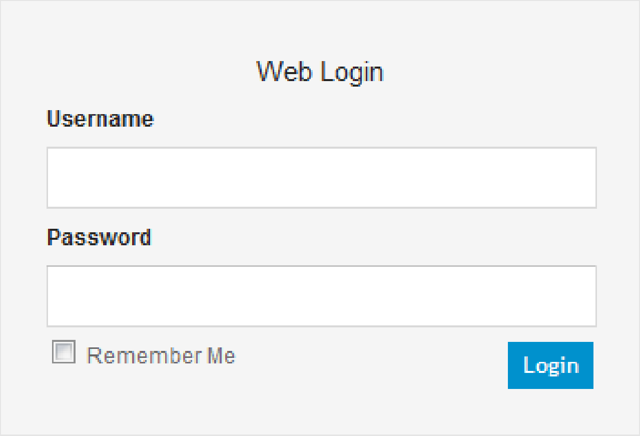
7.1.8 Creating a Login Welcome Message

You can create a Welcome message box that is displayed on the Web Login page. The figure below displays an example of a Welcome message:

Figure 13: User-Defined Web Welcome Message after Login



Note
****
********* This is a Welcome message **
****



To enable and create a Welcome message, use the WelcomeMessage table ini file parameter, as described in the table below. If this parameter is not configured, no Welcome message is displayed.

Table 7: *ini* File Parameter for Welcome Login Message

Parameter	Description
[WelcomeMessage]	Enables and defines a Welcome message that appears on the Web Login page for logging in to the Web interface. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; [\WelcomeMessage] For Example: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; WelcomeMessage 1 = "*********************************"; WelcomeMessage 2 = "********* This is a Welcome message **"; WelcomeMessage 3 = "*********************************"; [\WelcomeMessage] Each index row represents a line of text in the Welcome message box. Up to 20 lines (or rows) of text can be defined.

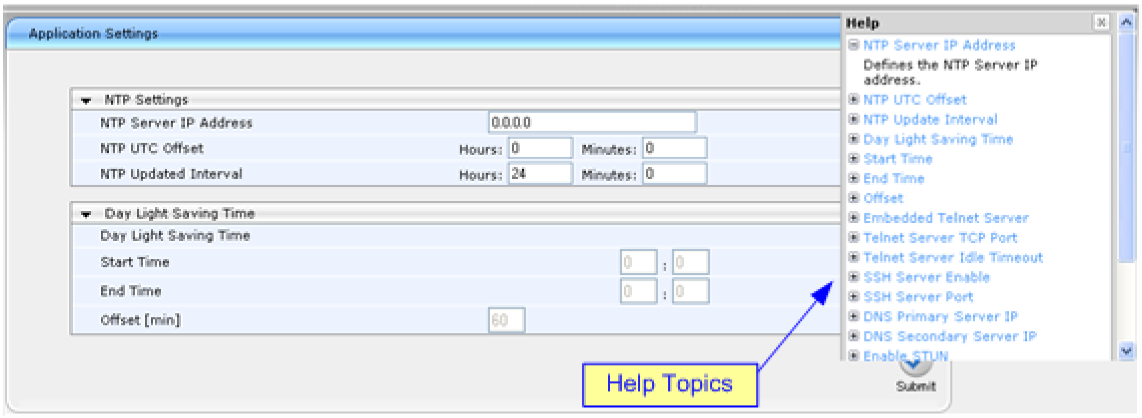
7.1.9 Getting Help

The Web interface provides you with context-sensitive Online Help. The Online Help provides brief descriptions of parameters pertaining to the currently opened page.

➢ To view the Help topic of a currently opened page:

1. On the toolbar, click the **Help** button; the Help topic pertaining to the opened page appears, as shown below:

Figure 14: Help Topic for Current Page



Application Settings

NTP Settings
- NTP Server IP Address: 0.0.0.0
- NTP UTC Offset: Hours: 0 Minutes: 0
- NTP Updated Interval: Hours: 24 Minutes: 0

Day Light Saving Time
- Day Light Saving Time
- Start Time: 0 : 0
- End Time: 0 : 0
- Offset [min]: 60

Help

⊟ NTP Server IP Address

Defines the NTP Server IP address.

2. To view a description of a parameter, click the **plus** ⊞ sign to expand the parameter. To collapse the description, click the **minus** ⊟ sign.

3. To close the Help topic, click the **close** ☒ button located on the top-right corner of the Help topic window or simply click the **Help** button.

Note: Instead of clicking the **Help** button for each page you open, you can open it once for a page and then simply leave it open. Each time you open a different page, the Help topic pertaining to that page is automatically displayed.

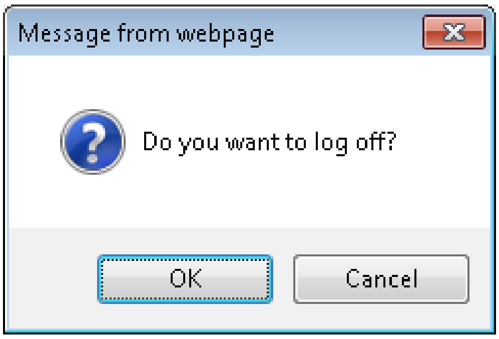
7.1.10 Logging Off the Web Interface

The procedure below describes how to log off the Web interface.

➢ To log off the Web interface:

1. On the toolbar, click the **Log Off** icon; the following confirmation message box appears:

Figure 15: Log Off Confirmation Box



Message from webpage

Do you want to log off?

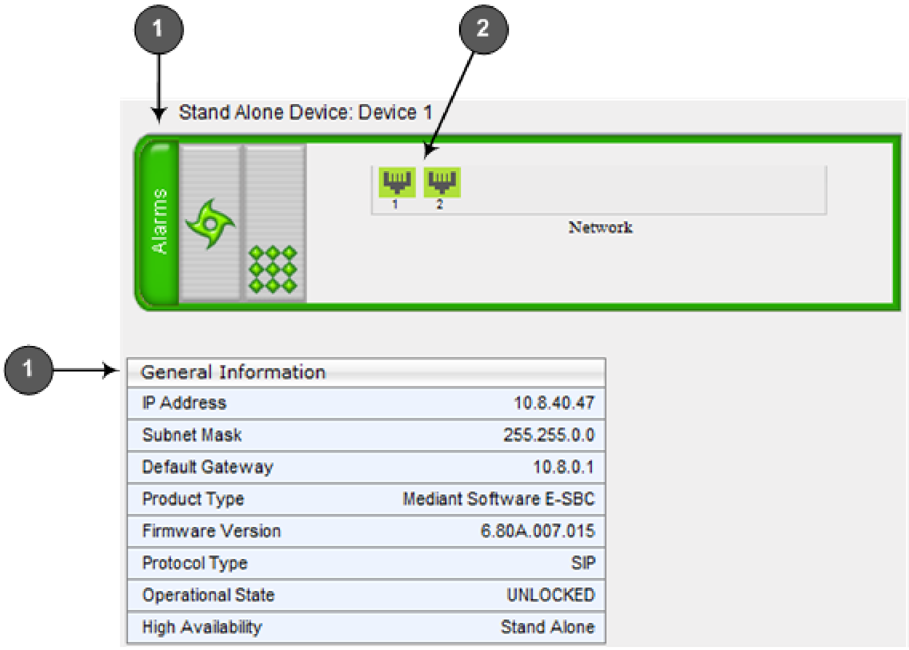
2. Click **OK**; you are logged off the Web session and the Web Login dialog box appears enabling you to re-login, if required.

7.2 Viewing the Home Page

The Home page is displayed when you access the device's Web interface. The Home page provides you with a graphical display of the device's front panel, showing color-coded status icons for various operations device.

➢ To access the Home page:

- On the toolbar, click the **Home** icon.



General Information	
IP Address	10.8.40.47
Subnet Mask	255.255.0.0
Default Gateway	10.8.0.1
Product Type	Mediant Software E-SBC
Firmware Version	6.80A.007.015
Protocol Type	SIP
Operational State	UNLOCKED
High Availability	Stand Alone

In addition to the color-coded status information depicted on the graphical display of the device, the Home page displays various read-only information in the General Information pane:

- **IP Address:** IP address of the device
- **Subnet Mask:** Subnet mask address of the device
- **Default Gateway Address:** Default gateway used by the device
- **Firmware Version:** Software version running on the device
- **Protocol Type:** Signaling protocol currently used by the device (i.e. SIP)
- **Gateway Operational State:**
  - "LOCKED": device is locked (i.e. no new calls are accepted)
  - "UNLOCKED": device is not locked
  - "SHUTTING DOWN": device is currently shutting down

  To perform these operations, see "Basic Maintenance" on page 292.
- **High Availability:** Status of the device's HA mode (see HA Status on the Home Page on page 277)

The table below describes the areas of the Home page.

Table 8: Home Page Description

Item #	Description
1	Displays the highest severity of an active alarm raised (if any) by the device: ▪ Green = No alarms ▪ Red = Critical alarm ▪ Orange = Major alarm

Item #	Description
	▪ Yellow = Minor alarm To view active alarms, click this Alarms area to open the Active Alarms page (see Viewing Active Alarms on page 333).
2	Gigabit Ethernet port status icons: ▪ 🟩 (green): Ethernet link is working ▪ ⬜ (gray): Ethernet link is not connected To view detailed Ethernet port information, click these icons to open the Ethernet Port Information page (see Viewing Ethernet Port Information on page 331).
3	General Information pane, displaying the following: ▪ Firmware Version: software version currently running on the device ▪ Protocol Type: signaling protocol currently used by the device (i.e. SIP) ▪ Gateway Operational State: operational state of the device: ✓ "LOCKED" - device is locked (i.e. no new calls are accepted) ✓ "UNLOCKED" - device is not locked ✓ "SHUTTING DOWN" - device is currently shutting down ▪ High Availability: status of the device's HA mode: ✓ "Not Operational": HA is not configured or device not installed with HA Software License Key ✓ "Synchronizing": Redundant device synchronizing with Active device ✓ "Operational": Device is in HA mode ✓ "Stand Alone": HA is configured but Redundant device is missing and HA is currently unavailable ✓ "Not Available": HA is not configured correctly (error)

7.3 Configuring Web User Accounts

Web user accounts define users for the Web interface and CLI. User accounts permit login access to these interfaces as well as different levels of read and write privileges. Thus, user accounts prevent unauthorized access to these interfaces, permitting access only to users with correct credentials (i.e., username and password).

Each user account is based on the following:

- **Username and password:** Credentials that enable authorized login access to the Web interface.
- **User level (user type):** Access privileges specifying what the user can view in the Web interface and its read/write privileges. The table below describes the different types of Web user account access levels:

Table 9: Web User Access Levels and Privileges

User Level	Numeric Representation in RADIUS	Privileges
Security Administrator	200	Read / write privileges for all pages. It can create all user types and is the only one that can create the first Master user. **Note:** At least one Security Administrator user must exits.
Master	220	Read / write privileges for all pages. Can create all user types, including additional Master users and Security Administrators. It can delete all users except the last Security Administrator.

User Level	Numeric Representation in RADIUS	Privileges
Administrator	100	Read / write privileges for all pages, except security-related pages (read-only).
Monitor	50	No access to security-related and file-loading pages; read-only access to all other pages.
No Access	0	No access to any page. **Note:** This access level is not applicable when using advanced Web user account configuration in the Web Users table.

By default, the device is pre-configured with the following two Web user accounts:

Table 10: Pre-configured Web User Accounts

User Access Level	Username (Case-Sensitive)	Password (Case-Sensitive)
Security Administrator	Admin	Admin
Monitor	User	User

After you log in to the Web interface, the username is displayed on the toolbar.

If the Web session is idle (i.e., no actions are performed) for more than five minutes, the Web session expires and you are once again requested to login with your username and password. Users can be blocked for a period of time upon a user-defined number of unsuccessful login attempts. Login information (such as how many login attempts were made and the last successful login time) can be presented to the user.

➢ To prevent user access after a specific number of failed logins:

1. From the 'Deny Access On Fail Count' drop-down list, select the number of failed logins after which the user is prevented access to the device for a user-defined time (see next step).
2. In the 'Deny Authentication Timer' field, enter the interval (in seconds) that the user needs to wait before a new login attempt from the same IP address can be done after reaching the number of failed login attempts (defined in the previous step).

Notes:

- For security, it's recommended that you change the default username and password of the pre-configured users (i.e., Security Administrator and Monitor users).
- The Security Administrator user can change all attributes of all Web user accounts. Web users with access levels other than Security Administrator can change only their password and username.
- To restore the two Web user accounts to default settings (usernames and passwords), set the *ini* file parameter ResetWebPassword to 1.
- To log in to the Web interface with a different Web user, click the **Log off** button and then login with with a different username and password.
- You can set the entire Web interface to read-only (regardless of Web user access levels), by using the *ini* file parameter DisableWebConfig (see "Web and Telnet Parameters" on page 381).
- Only Security Administrator and Master user levels can access the CLI interface.
- You can define additional Web user accounts using a RADIUS server (see "RADIUS Authentication" on page 52).

7.3.1 Basic User Accounts Configuration

This section describes basic Web user account configuration. This is relevant only if the two default, pre-configured Web user accounts--Security Administrator ("Admin") and Monitor ("User")--are sufficient for your management scheme.

The Web user account parameters that can be modified depends on the access level of the currently logged-in Web user:

Table 11: Allowed Modifications per Web User Level

Logged-in User	Web User Level	Allowed Modifications
Security Administrator	(Default) Security Administrator	Username and password
	Monitor	Username, password, and access level
Monitor	(Default) Security Administrator	None
	Monitor	Username and password

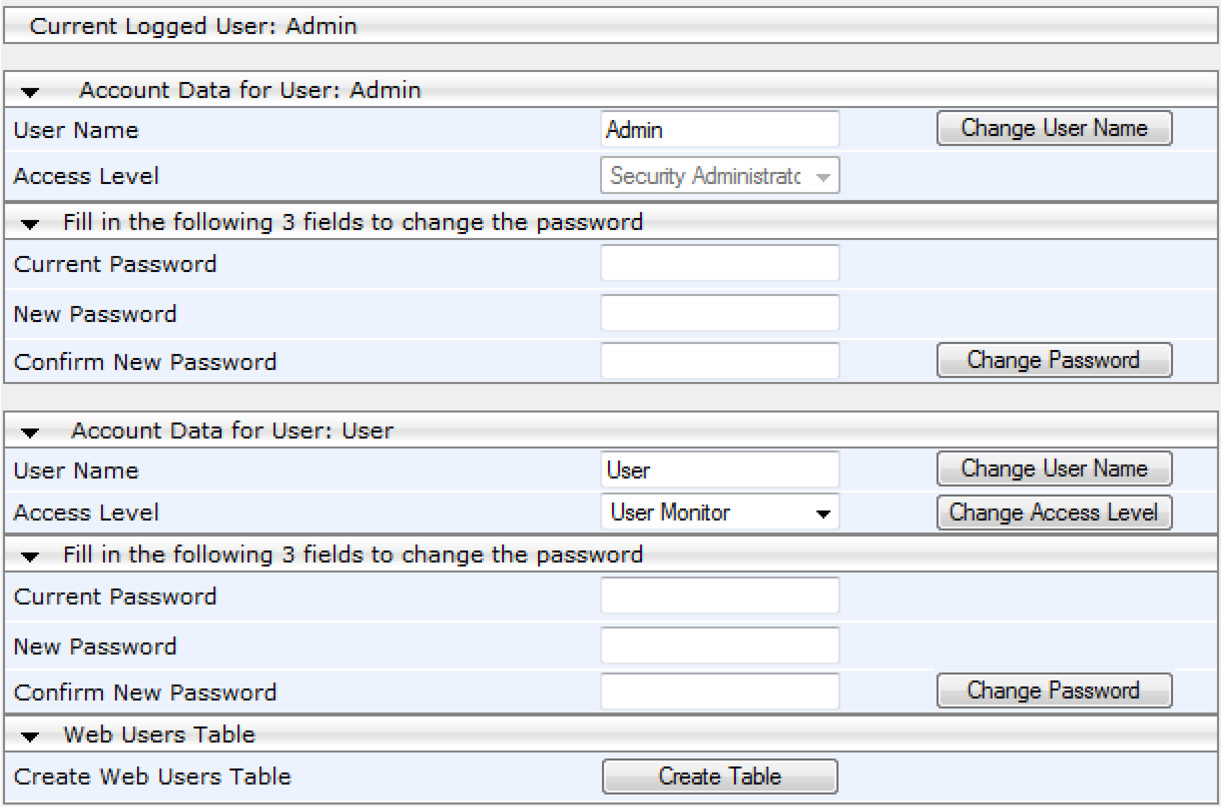
Notes:

- The username and password can be a string of up to 19 characters and are case-sensitive.
- When only the basic user accounts are being used, up to two users can be concurrently logged in to the Web interface, and they can be the same user.

➢ To configure the two pre-configured Web user accounts:

1. Open the Web User Accounts page (**Configuration** tab > **System** menu > **Web User Accounts**). If you are logged in as Security Administrator, both Web user accounts are displayed (as shown below). If you are logged in with the second user account, only the details of this user account are displayed.

Figure 16: WEB User Accounts Page (for Users with 'Security Administrator' Privileges)



Current Logged User: Admin

Account Data for User: Admin

User Name	Admin	Change User Name
Access Level	Security Administratc	

Fill in the following 3 fields to change the password

Current Password		
New Password		
Confirm New Password		Change Password

Account Data for User: User

User Name	User	Change User Name
Access Level	User Monitor	Change Access Level

Fill in the following 3 fields to change the password

Current Password		
New Password		
Confirm New Password		Change Password

Web Users Table

Create Web Users Table	Create Table

2. To change the username of an account:
   a. In the 'User Name' field, enter the new user name.
   b. Click **Change User Name**; if you are currently logged in to the Web interface with this account, the 'Web Login' dialog box appears.
   c. Log in with your new user name.
3. To change the password of an account:
   a. In the 'Current Password' field, enter the current password.
   b. In the 'New Password' and 'Confirm New Password' fields, enter the new password.
   c. Click **Change Password**; if you are currently logged in to the Web interface with this account, the 'Web Login' dialog box appears.
   d. Log in with your new password.
4. To change the access level of the optional, second account:
   a. Under the **Account Data for User: User** group, from the 'Access Level' drop-down list, select a new access level user.
   b. Click **Change Access Level**; the new access level is applied immediately.

7.3.2 Advanced User Accounts Configuration

The Web Users table lets you configure advanced Web user accounts. This configuration is relevant only if you need the following management schemes:

- Enhanced security settings per Web user (e.g., limit session duration)
- More than two Web user accounts (up to 10 Web user accounts)
- Master users

Notes:

- Only the Security Administrator user can **initially** access the Web Users table.
- Only Security Administrator and Master users can add, edit, or delete users.
- Admin users have read-only privileges in the Web Users table; Monitor users have no access to this table.
- For advanced user accounts, up to five users can be concurrently logged in to the Web interface, and they can be the same user.
- If you delete a user who is currently in an active Web session, the user is immediately logged off by the device.
- All users can change their own passwords. This is done in the WEB Security Settings page (see "Configuring Web Security Settings" on page 49).
- To remove the Web Users table and revert to the Web User Accounts page with the pre-configured, default Web user accounts, set the ResetWebPassword *ini* file parameter to 1. This also deletes all other Web users.
- Once the Web Users table is accessed, Monitor users and Admin users can only change their passwords in the Web Security Settings page (see "Configuring Web Security Settings" on page 49). The new password must have at least four different characters than the previous password. (The Security Administrator users and Master users can change their passwords in the Web Users table and in the Web Security Settings page.)

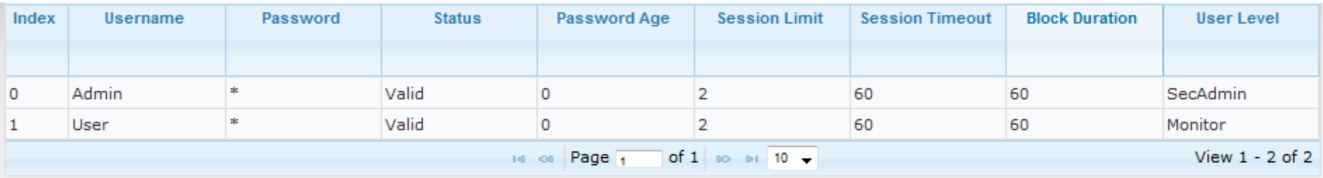
The procedure below describes how to configure Web users in the Web interface. You can also configure this using the CLI command web-users.

➢ To add Web user accounts with advanced settings:

1. Open the Web Users Table page:
   - Upon initial access:
     a. Open the Web User Accounts page (**Configuration** tab > **System** menu > **Web User Accounts**).
     b. Under the **Web Users Table** group, click the **Create Table** button.
   - Subsequent access: **Configuration** tab > **System** menu > **Web User Accounts**.

   The Web Users table appears, listing the two default, pre-configured Web use accounts - Security Administrator ("Admin") and Monitor ("User"):

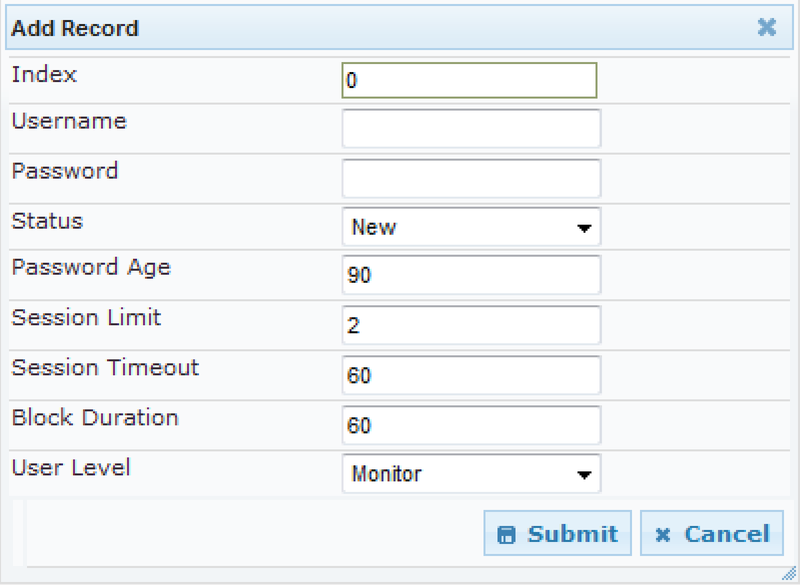
Figure 17: Web Users Table Page



Index	Username	Password	Status	Password Age	Session Limit	Session Timeout	Block Duration	User Level
0	Admin	*	Valid	0	2	60	60	SecAdmin
1	User	*	Valid	0	2	60	60	Monitor

2. Click **Add**; the following dialog box is displayed:

Figure 18: Web Users Table - Add Record Dialog Box



Add Record

Index	0
Username	
Password	
Status	New
Password Age	90
Session Limit	2
Session Timeout	60
Block Duration	60
User Level	Monitor

3. Configure a Web user according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 12: Web User Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Web: Username CLI: user-name	Defines the Web user's username. The valid value is a string of up to 40 alphanumeric characters, including the period ".", underscore "_", and hyphen "-" signs.
Web: Password CLI: password	Defines the Web user's password. The valid value is a string of 8 to 40 ASCII characters, which must include the following: ▪ At least eight characters ▪ At least two letters that are upper case (e.g., "AA") ▪ At least two letters that are lower case (e.g., "aa") ▪ At least two numbers ▪ At least two signs (e.g., the dollar "$" sign) ▪ No spaces in the string ▪ At least four characters different to the previous password

Parameter	Description
Web: Status CLI: status	Defines the status of the Web user. - New = (Default) User is required to change its password on the next login. When the user logs in to the Web interface, the user is immediately prompted to change the current password. - Valid = User can log in to the Web interface as normal. - Failed Access = This state is automatically set for users that exceed a user-defined number of failed login attempts, set by the 'Deny Access on Fail Count' parameter (see "Configuring Web Security Settings" on page 49). These users can log in only after a user-defined timeout configured by the 'Block Duration' parameter (see below) or if their status is changed (to New or Valid) by a System Administrator or Master. - Old Account = This state is automatically set for users that have not accessed the Web interface for a user-defined number of days, set by the 'User Inactivity Timer' (see "Configuring Web Security Settings" on page 49). These users can only log in to the Web interface if their status is changed (to New or Valid) by a System Administrator or Master. **Notes:** - The Old Account status is applicable only to Admin and Monitor users; System Administrator and Master users can be inactive indefinitely. - For security, it is recommended to set the status of a newly added user to New in order to enforce password change.
Web: Password Age CLI: pw-age-interval	Defines the duration (in days) of the validity of the password. When this duration elapses, the user is prompted to change the password; otherwise, access to the Web interface is blocked. The valid value is 0 to 10000, where 0 means that the password is always valid. The default is 90.
Web: Session Limit CLI: session-limit	Defines the maximum number of Web interface sessions allowed for the user. In other words, this allows the same user account to log in to the device from different sources (i.e., IP addresses). The valid value is 0 to 5. The default is 2. **Note:** Up to 5 users can be logged in to the Web interface at any given.
Web: Session Timeout CLI: session-timeout	Defines the duration (in minutes) of Web inactivity of a logged-in user, after which the user is automatically logged off the Web interface. The valid value is 0 to 100000. The default value is according to the settings of the 'Session Timeout' global parameter (see "Configuring Web Security Settings" on page 49).
Web: Block Duration CLI: block-time	Defines the duration (in seconds) for which the user is blocked when the user exceeds a user-defined number of failed login attempts. This is configured by the 'Deny Access On Fail Count' parameter (see "Configuring Web Security Settings" on page 49). The valid value is 0 to 100000, where 0 means that the user can do as many login failures without getting blocked. The default is according to the settings of the 'Deny Authentication Timer' parameter (see "Configuring Web Security Settings" on page 49). **Note:** The 'Deny Authentication Timer' parameter relates to failed Web logins from specific IP addresses.

Parameter	Description
Web: User Level CLI: user-level	Defines the user's access level. ▪ Monitor = (Default) Read-only user. This user can only view Web pages and access to security-related pages is denied. ▪ Administrator = Read/write privileges for all pages, except security-related pages including the Web Users table where this user has only read-only privileges. ▪ Security Administrator = Read/write privileges for all pages. This user is the Security Administrator. ▪ Master = Read/write privileges for all pages. This user also functions as a security administrator. **Notes:** ▪ At least one Security Administrator must exist. The last remaining Security Administrator cannot be deleted. ▪ The first Master user can be added only by a Security Administrator user. ▪ Additional Master users can be added, edited and deleted only by Master users. ▪ If only one Master user exists, it can be deleted only by itself. ▪ Master users can add, edit, and delete Security Administrators (but cannot delete the last Security Administrator). ▪ Only Security Administrator and Master users can add, edit, and delete Administrator and Monitor users.

7.4 Displaying Login Information upon Login

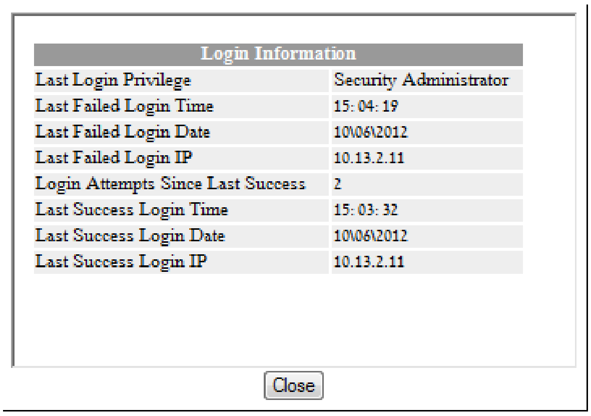
The device can display login information immediately upon Web login.

➤ To enable display of user login information upon a successful login:

1. Open the WEB Security Settings page (**Configuration** tab > **System** menu > **Management** > **WEB Security Settings**).
2. From the 'Display Login Information' drop-down list, select **Yes**.
3. Click **Submit**.

Once enabled, the Login Information window is displayed upon a successful login, as shown in the example below:

Figure 19: Login Information Window



Login Information	
Last Login Privilege	Security Administrator
Last Failed Login Time	15: 04: 19
Last Failed Login Date	10\06\2012
Last Failed Login IP	10.13.2.11
Login Attempts Since Last Success	2
Last Success Login Time	15: 03: 32
Last Success Login Date	10\06\2012
Last Success Login IP	10.13.2.11

7.5 Configuring Web Security Settings

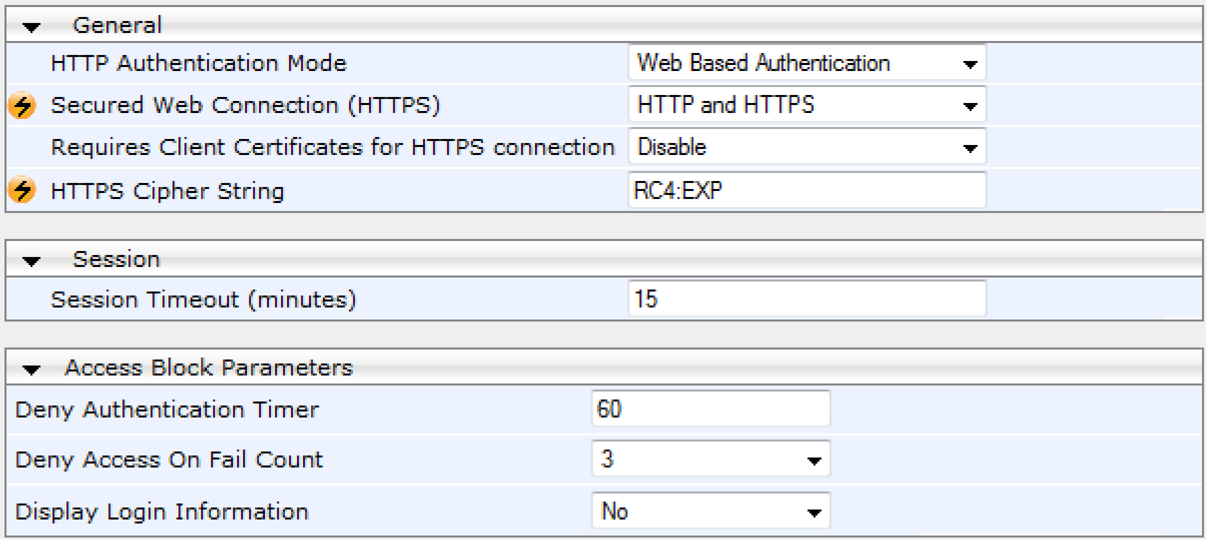
The WEB Security Settings page is used to configure security for the device's Web interface.

By default, the device accepts HTTP and HTTPS access. However, you can enforce secure Web access communication method by configuring the device to accept only HTTPS.

For a description of these parameters, see "Web and Telnet Parameters" on page 381.

➢ To define Web access security:

1. Open the WEB Security Settings page (**Configuration** tab > **System** menu > **Management** > **WEB Security Settings**).



General

HTTP Authentication Mode	Web Based Authentication
Secured Web Connection (HTTPS)	HTTP and HTTPS
Requires Client Certificates for HTTPS connection	Disable
HTTPS Cipher String	RC4:EXP

Session

Session Timeout (minutes)	15

Access Block Parameters

Deny Authentication Timer	60
Deny Access On Fail Count	3
Display Login Information	No

2. Set the 'Secured Web Connection (HTTPS)' parameter to **HTTPS Only**.
3. Configure the parameters as required.
4. Click **Submit**.
5. To save the changes to flash memory, see "Saving Configuration" on page 295.

7.6 Web Login Authentication using Smart Cards

You can enable Web login authentication using certificates from a third-party, common access card (CAC) with user identification. When a user attempts to access the device through the Web browser (HTTPS), the device retrieves the Web user's login username (and other information, if required) from the CAC. The user attempting to access the device is only required to provide the login password. Typically, a TLS connection is established between the CAC and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username. Therefore, this feature implements a two-factor authentication - what the user has (i.e., the physical card) and what the user knows (i.e., the login password).

This feature is enabled using the EnableMgmtTwoFactorAuthentication parameter.

Note: For specific integration requirements for implementing a third-party smart card for Web login authentication, contact your AudioCodes representative.

➢ To log in to the Web interface using CAC:

1. Insert the Common Access Card into the card reader.
2. Access the device using the following URL: https://<host name or IP address>; the device prompts for a username and password.
3. Enter the password only. As some browsers require that the username be provided, it's recommended to enter the username with an arbitrary value.

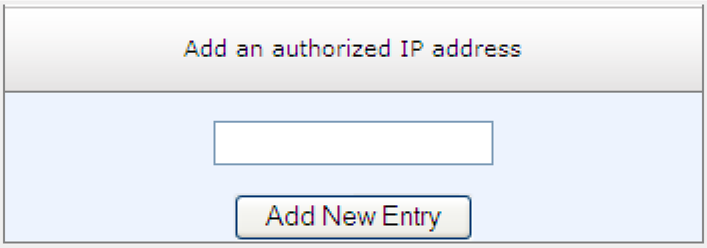
7.7 Configuring Web and Telnet Access List

The Web & Telnet Access List page is used to define IP addresses (up to ten) that are permitted to access the device's Web, Telnet, and SSH interfaces. Access from an undefined IP address is denied. If no IP addresses are defined, this security feature is inactive and the device can be accessed from any IP address. The Web and Telnet Access List can also be defined using the *ini* file parameter WebAccessList_x (see "Web and Telnet Parameters" on page 381).

➢ To add authorized IP addresses for Web, Telnet, and SSH interfaces access:

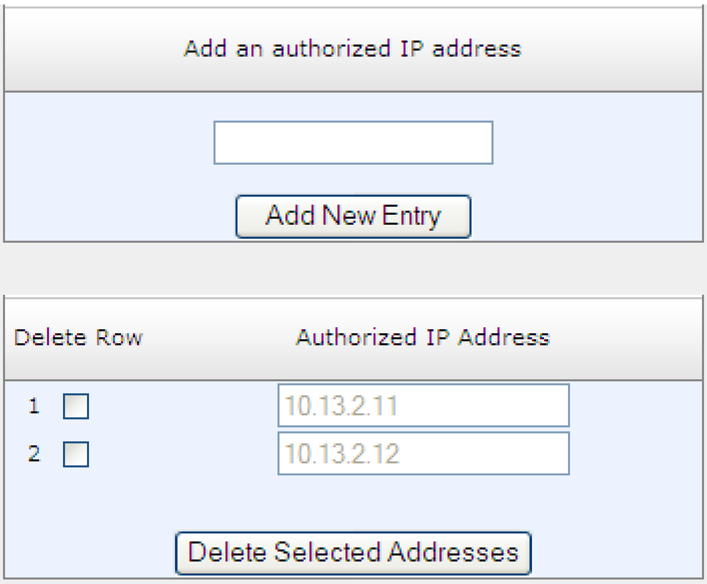
1. Open the Web & Telnet Access List page (**Configuration** tab > **System** menu > **Management** > **Web & Telnet Access List**).

Figure 20: Web & Telnet Access List Page - Add New Entry



2. To add an authorized IP address, in the 'Add an authorized IP address' field, enter the required IP address, and then click **Add New Entry**; the IP address you entered is added as a new entry to the Web & Telnet Access List table.

Figure 21: Web & Telnet Access List Table



Delete Row		Authorized IP Address
1	□	10.13.2.11
2	□	10.13.2.12

3. To delete authorized IP addresses, select the Delete Row check boxes corresponding to the IP addresses that you want to delete, and then click **Delete Selected Addresses**; the IP addresses are removed from the table and these IP addresses can no longer access the Web and Telnet interfaces.
4. To save the changes to flash memory, see "Saving Configuration" on page 295.

Notes:

- The first authorized IP address in the list must be your PC's (terminal) IP address; otherwise, access from your PC is denied.
- Delete your PC's IP address last from the 'Web & Telnet Access List page. If it is deleted before the last, subsequent access to the device from your PC is denied.

7.8 RADIUS Authentication

You can enhance security for your device by implementing Remote Authentication Dial-In User Service (RADIUS - RFC 2865) for authenticating multiple login user accounts of the device's embedded Web and Telnet servers. Thus, RADIUS also prevents unauthorized access to your device.

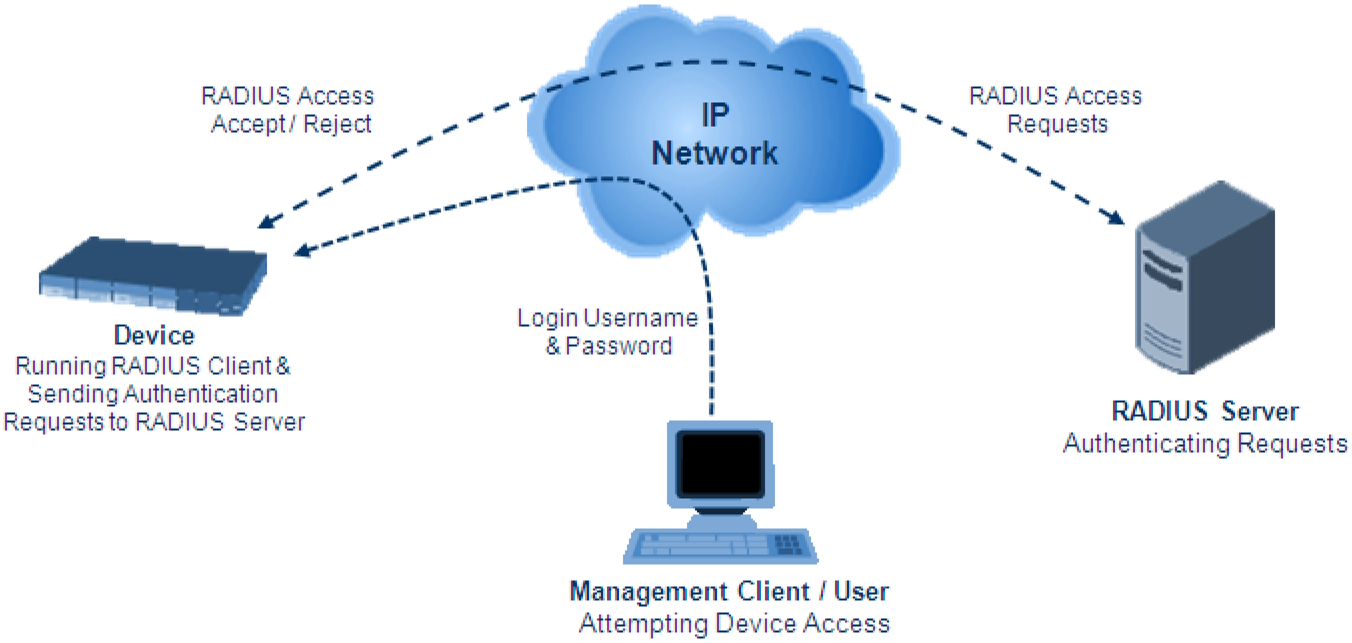
When RADIUS authentication is not used, the login username and password are locally authenticated by the device with the Web interface's local usernames and passwords or with the Telnet server's usernames and passwords.

When RADIUS authentication is used, the RADIUS server stores the device's login usernames, passwords, and access (authorization) levels (Web only). When a management client tries to access the device, the device sends the RADIUS server the client's username and password for authentication. The RADIUS server replies with an acceptance or a rejection notification. During the RADIUS authentication process, the device's Web interface is blocked until an acceptance response is received from the RADIUS server.

The local Web and Telnet usernames and passwords can be used as a fallback mechanism in case the RADIUS server does not respond.

Note that communication between the device and the RADIUS server is done by using a Shared Secret, which is not transmitted over the network.

Figure 22: RADIUS Login Authentication for Management



For using RADIUS, you need to do the following:

- Set up a RADIUS server (third-party) to communicate with the device - see "Setting Up a Third-Party RADIUS Server" on page 52
- Configure the device as a RADIUS client for communication with the RADIUS server - see "Configuring RADIUS Authentication" on page 53

7.8.1 Setting Up a Third-Party RADIUS Server

The procedure below provides an example for setting up the third-party RADIUS sever, *FreeRADIUS*, which can be downloaded from www.freeradius.org. Follow the instructions on this Web site for installing and configuring the server. If you use a RADIUS server from a different vendor, refer to its appropriate documentation.

➢ To set up a third-party RADIUS server (e.g., *FreeRADIUS)*:

1. Define the AudioCodes device as an authorized client of the RADIUS server, with the following:

- Predefined *shared secret* (password used to secure communication between the device and the RADIUS server)
- Vendor ID

Below is an example of the *clients.conf* file (FreeRADIUS client configuration):

```
#
# clients.conf - client configuration directives
#
client 10.31.4.47 {
        secret          = FutureRADIUS
        shortname       = audc_device
}
```

2. If access levels are required, set up a Vendor-Specific Attributes (VSA) dictionary for the RADIUS server and select an attribute ID that represents each user's access level. The example below shows a dictionary file for FreeRADIUS that defines the attribute "ACL-Auth-Level" with "ID=35". For the device's user access levels and their corresponding numeric representation in RADIUS servers, see "Configuring Web User Accounts" on page 42.

```
#
# AudioCodes VSA dictionary
#
VENDOR AudioCodes 5003
ATTRIBUTE ACL-Auth-Level 35 integer AudioCodes
VALUE ACL-Auth-Level ACL-Auth-UserLevel 50
VALUE ACL-Auth-Level ACL-Auth-AdminLevel 100
VALUE ACL-Auth-Level ACL-Auth-SecurityAdminLevel 200
```

3. Define the list of users authorized to use the device, using one of the password authentication methods supported by the server implementation. The example below shows a user configuration file for FreeRADIUS using a plain-text password:

```
# users - local user configuration database

john    Auth-Type := Local, User-Password == "qwerty"
        Service-Type = Login-User,
        ACL-Auth-Level = ACL-Auth-SecurityAdminLevel

sue   Auth-Type := Local, User-Password == "123456"
        Service-Type = Login-User,
        ACL-Auth-Level = ACL-Auth-UserLevel
```

4. Record and retain the IP address, port number, shared secret code, vendor ID, and VSA access level identifier (if access levels are implemented) used by the RADIUS server.

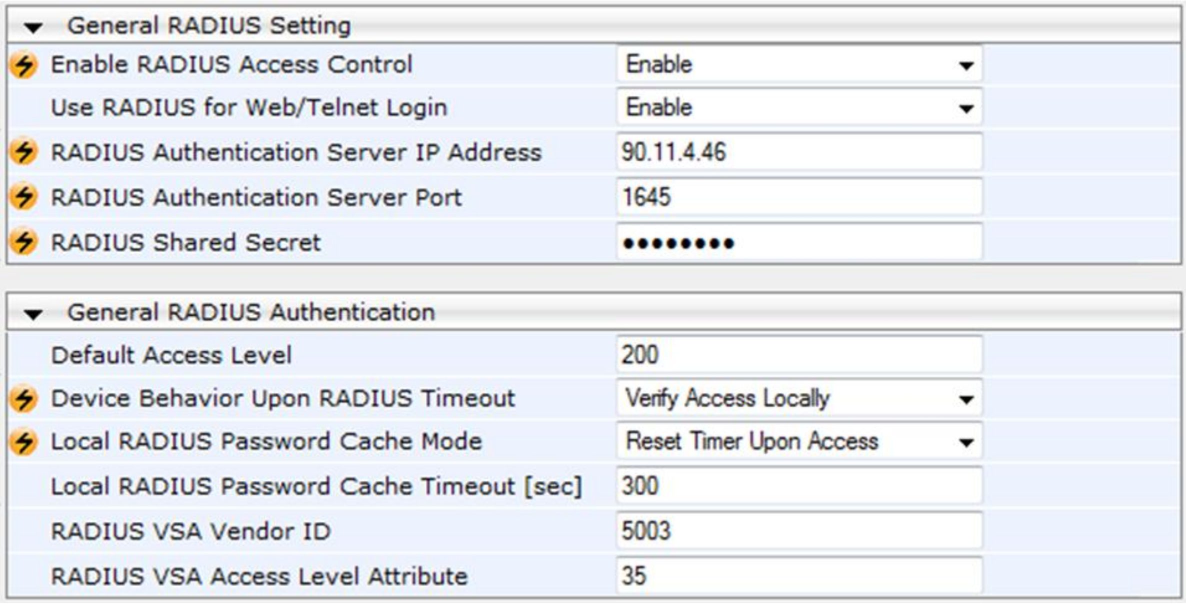
7.8.2 Configuring RADIUS Authentication

The procedure below describes how to configure the RADIUS feature. For a detailed description of the RADIUS parameters, see "RADIUS Parameters" on page 414.

➢ To configure RADIUS:

1. Open the RADIUS Settings page (**Configuration** tab > **System** menu > **Management** > **RADIUS Settings**).

Figure 23: RADIUS Parameters Page



General RADIUS Setting

⚡ Enable RADIUS Access Control	Enable
Use RADIUS for Web/Telnet Login	Enable
⚡ RADIUS Authentication Server IP Address	90.11.4.46
⚡ RADIUS Authentication Server Port	1645
⚡ RADIUS Shared Secret	●●●●●●●●

General RADIUS Authentication

Default Access Level	200
⚡ Device Behavior Upon RADIUS Timeout	Verify Access Locally
⚡ Local RADIUS Password Cache Mode	Reset Timer Upon Access
Local RADIUS Password Cache Timeout [sec]	300
RADIUS VSA Vendor ID	5003
RADIUS VSA Access Level Attribute	35

2. Set the 'Enable RADIUS Access Control' parameter to **Enable** to enable the RADIUS application.
3. Set the 'Use RADIUS for Web/Telnet Login' parameter to **Enable** to enable RADIUS authentication for Web and Telnet login.
4. Define the RADIUS server:
   a. In the 'RADIUS Authentication Server IP Address' field, enter the RADIUS server's IP address.
   b. In the 'RADIUS Authentication Server Port' field, enter the RADIUS server's port number.
   c. In the 'RADIUS Shared Secret' field, enter the shared secret code used to authenticate the device to the RADIUS server.
5. In the 'RADIUS VSA Vendor ID' field, enter the device's vendor ID. This must be the same one as configured in the RADIUS server.

6. When implementing Web user access levels, do one of the following:
    - **If the RADIUS server response includes the access level attribute:** In the 'RADIUS VSA Access Level Attribute' field, enter the code that indicates the access level attribute in the VSA section of the received RADIUS packet. For defining the RADIUS server with access levels, see "Setting Up a Third-Party RADIUS Server" on page 52.
    - **If the RADIUS server responses exclude the access level attribute:**
      In the 'Default Access Level' field, enter the default access level that is applied to all users authenticated by the RADIUS server.
7. Define RADIUS timeout handling:
    a. From the 'Device Behavior Upon RADIUS Timeout' drop-down list, select the option if the RADIUS server does not respond within five seconds:
        - ♦ **Deny Access**: device denies access to the Web and Telnet interfaces.
        - ♦ **Verify Access Locally**: device checks the user name and password defined locally for the device (in the Web User Accounts page) and if correct, it allows access.

    b. In the 'Local RADIUS Password Cache Timeout' field, enter a time limit (in seconds) after which the user name and password verified by the RADIUS server becomes invalid and a user name and password must be re-validated with the RADIUS server.

    c. From the 'Local RADIUS Password Cache Mode' drop-down list, select the option for the local RADIUS password cache timer:
        - ♦ **Reset Timer Upon Access**: upon each access to a Web page, the timer resets (reverts to the initial value configured in the previous step).
        - ♦ **Absolute Expiry Timer**: when you access a Web page, the timer doesn't reset, but continues its count down.
8. Click **Submit** and save the changes to flash memory, see "Saving Configuration" on page 295.

7.8.3 Securing RADIUS Communication

RADIUS authentication requires HTTP basic authentication (according to RFC 2617). However, this is insecure as the usernames and passwords are transmitted in clear text over plain HTTP. Thus, as digest authentication is not supported with RADIUS, it is recommended that you use HTTPS with RADIUS so that the usernames and passwords are encrypted.

To configure the device to use HTTPS, set the 'Secured Web Connection (HTTPS)' parameter to **HTTPS Only**, in the WEB Security Settings page (**Configuration** tab > **System** menu > **Management** > **WEB Security Settings**).

7.8.4 Authenticating RADIUS in the URL

RADIUS authentication is typically done after the user accesses the Web interface by entering only the device's IP address in the Web browser's URL field (for example, http://10.13.4.12/), and then entering the username and password credentials in the Web interface login screen. However, authentication with the RADIUS server can also be done immediately after the user enters the URL, if the URL also contains the login credentials, for example: http://10.4.4.112/Forms/RadiusAuthentication?WSBackUserName=John&WSBackPassword=1234

Note: This feature allows up to five simultaneous users only.

8 CLI-Based Management

This chapter provides an overview of the CLI-based management and provides configuration relating to CLI management.

Notes:

- For security, CLI is disabled by default.
- Only Security Administrator and Master user levels can access the CLI interface. CLI and Web interface users are configured in "Configuring Web User Accounts" on page 42.
- For a description of the CLI commands, refer to the CLI Reference Guide.

8.1 Enabling CLI

Access to the device's CLI through Telnet and SSH is disabled by default. This section describes how to enable these protocols.

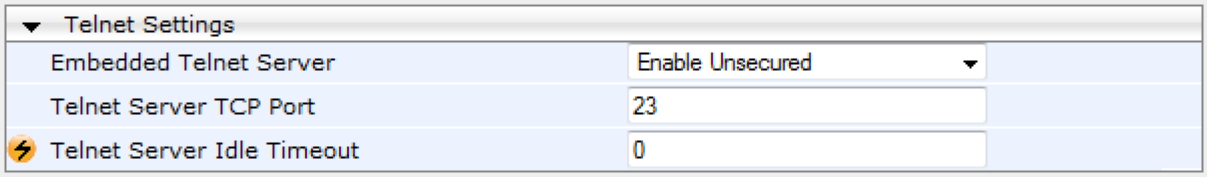
8.1.1 Enabling Telnet for CLI

The procedure below describes how to enable Telnet. You can enable a secured Telnet that uses Secure Socket Layer (SSL) where information is not transmitted in the clear. If SSL is used, a special Telnet client is required on your PC to connect to the Telnet interface over a secured connection; examples include C-Kermit for UNIX and Kermit-95 for Windows.

For security, some organizations require the display of a proprietary notice upon starting a Telnet session. You can use the configuration ini file parameter, WelcomeMessage to configure such a message (see "Creating a Login Welcome Message" on page 38).

➢ To enable Telnet:

1. Open the Telnet/SSH Settings page (**Configuration** tab > **System** menu > **Management** > **Telnet/SSH Settings**).



Telnet Settings

Embedded Telnet Server	Enable Unsecured
Telnet Server TCP Port	23
Telnet Server Idle Timeout	0

2. Set the 'Embedded Telnet Server' parameter to **Enable Unsecured** or **Enable Secured** (i.e, SSL).
3. Configure the other Tenet parameters as required. For a description of these parameters, see "Telnet Parameters" on page 384.
4. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

8.1.2 Enabling SSH with RSA Public Key for CLI

Unless configured for TLS, Telnet is not secure as it requires passwords to be transmitted in clear text. To overcome this, Secure SHell (SSH) is used, which is the de-facto standard for secure CLI. SSH 2.0 is a protocol built above TCP, providing methods for key exchange, authentication, encryption, and authorization.

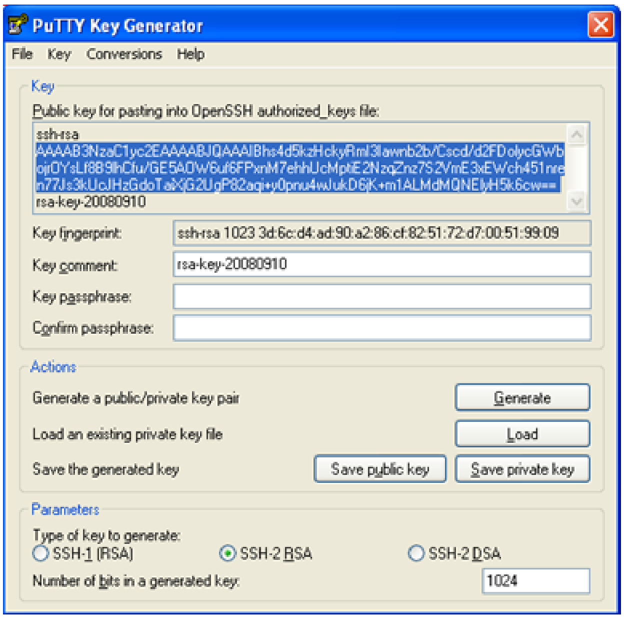
SSH requires appropriate client software for the management PC. Most Linux distributions have OpenSSH pre-installed; Windows-based PCs require an SSH client software such as PuTTY, which can be downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty/.

By default, SSH uses the same username and password as the Telnet and Web server. SSH supports 1024/2048-bit RSA public keys, providing carrier-grade security. Follow the instructions below to configure the device with an administrator RSA key as a means of strong authentication.

➢ To enable SSH and configure RSA public keys for Windows (using PuTTY SSH software):

1. Start the PuTTY Key Generator program, and then do the following:
   a. Under the 'Parameters' group, do the following:
      - Select the **SSH-2 RSA** option.
      - In the 'Number of bits in a generated key' field, enter "1024" bits.
   b. Under the 'Actions' group, click **Generate** and then follow the on-screen instructions.
   c. Under the 'Actions' group, click **Save private key** to save the new private key to a file (*.ppk) on your PC.
   d. Under the 'Key' group, select the displayed encoded text between "ssh-rsa" and "rsa-key-….", as shown in the example below:

Figure 24: Selecting Public RSA Key in PuTTY



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAIBhs4d5kzHckyRml3lawnb2b/Cscd/d2FDolycGWb
ojrOYsLf8B9hCfu/GE5A0W6uf6FPxnM7ehhUcMptE2NzqZnz7S2VmE3xEWch451nre
n77Js3kUcJHzGdoTaXjG2UgP82aqi+y0pnu4wJukD6jK+m1ALMdMQNElyH5k6cw==
rsa-key-20080910
```

Key fingerprint: ssh-rsa 1023 3d:6c:d4:ad:90:a2:86:cf:82:51:72:d7:00:51:99:09

Key comment: rsa-key-20080910

Key passphrase:

Confirm passphrase:

Generate a public/private key pair [Generate]

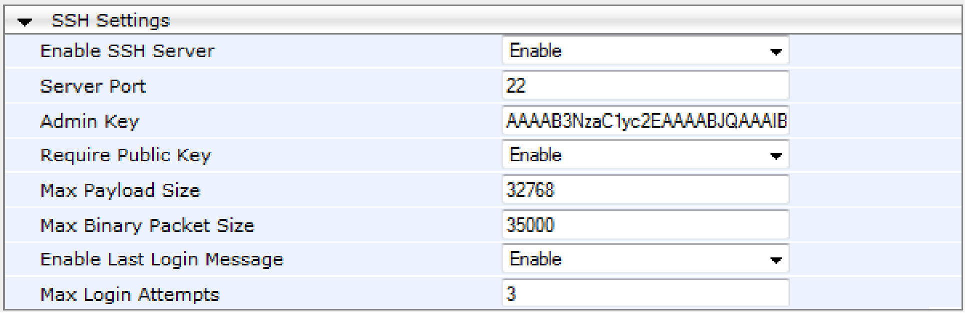
Load an existing private key file [Load]

Save the generated key [Save public key] [Save private key]

Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024

2. Open the Telnet/SSH Settings page (**Configuration** tab > **System** menu > **Management** > **Telnet/SSH Settings**), and then do the following:
   a. Set the 'Enable SSH Server' parameter to **Enable**.
   b. Paste the public key that you copied in Step 1.d into the 'Admin Key' field, as shown below:



SSH Settings

Enable SSH Server	Enable
Server Port	22
Admin Key	AAAAB3NzaC1yc2EAAAABJQAAAIB
Require Public Key	Enable
Max Payload Size	32768
Max Binary Packet Size	35000
Enable Last Login Message	Enable
Max Login Attempts	3

   c. For additional security, you can set the 'Require Public Key' to **Enable**. This ensures that SSH access is only possible by using the RSA key and not by using user name and password.
   d.
   e. Configure the other SSH parameters as required. For a description of these parameters, see "SSH Parameters" on page 412.
   f. Click **Submit**.
3. Start the PuTTY Configuration program, and then do the following:

a. In the 'Category' tree, drill down to **Connection**, then **SSH**, and then **Auth**; the 'Options controlling SSH authentication' pane appears.

b. Under the 'Authentication parameters' group, click **Browse** and then locate the private key file that you created and saved in Step 4.

4. Connect to the device with SSH using the username "Admin"; RSA key negotiation occurs automatically and no password is required.

➢ To configure RSA public keys for Linux (using OpenSSH 4.3):

1. Run the following command to create a new key in the admin.key file and to save the public portion to the admin.key.pub file:

```
ssh-keygen -f admin.key -N "" -b 1024
```

2. Open the admin.key.pub file, and then copy the encoded string from "ssh-rsa" to the white space.
3. Open the Telnet/SSH Settings page (**Configuration** tab > **System** menu > **Management** > **Telnet/SSH Settings**), and then paste the value copied in Step 2 into the 'Admin Key' field.
4. Click **Submit**.
5. Connect to the device with SSH, using the following command:

```
ssh -i admin.key xx.xx.xx.xx
```

where *xx.xx.xx.xx* is the device's IP address. RSA-key negotiation occurs automatically and no password is required.

8.2 Establishing a CLI Session

The device's CLI can be accessed using any of the following methods:

- **RS-232:** The device can be accessed through its RS-232 serial port, by connecting a VT100 terminal to it or using a terminal emulation program (e.g., HyperTerminal) with a PC. For connecting to the CLI through RS-232, see CLI.
- **Secure SHell (SSH):** The device can be accessed through its Ethernet interface by the SSH protocol using SSH client software. A popular and freeware SSH client software is Putty, which can be downloaded from http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html
- **Telnet:** The device can be accessed through its Ethernet interface by the Telnet protocol using Telnet client software.The procedure below describes how to establish a CLI session with the device.

The procedure below describes how to access the CLI through Telnet/SSH.

Notes:

- The CLI login credentials are the same as all the device's other management interfaces (such as Web interface). The default username and password is "Admin" and "Admin" (case-sensitive), respectively. For configuring login credentials, see "Configuring Web User Accounts" on page 42.
- Only the primary User Account, which has Security Administration access level (200), can access the device using Telnet.

➢ To establish a CLI session with the device:

1. Connect the device to the network.
2. Establish a Telnet or SSH session using the device's OAMP IP address.
3. Log in to the session using the username and password assigned to the Admin user of the Web interface:

a. At the Username prompt, type the username, and then press Enter:

```
Username: Admin
```

b. At the Password prompt, type the password, and then press Enter:

```
Password: Admin
```

c. At the prompt, type the following, and then press Enter:

```
> enable
```

d. At the prompt, type the password again, and then press Enter:

```
Password: Admin
```

8.3 Understanding Configuration Modes

Before you begin your CLI session, you should familiarize yourself with the CLI command modes. Each command mode provides different levels of access to commands, as described below:

- **Basic command mode:** This is the initial mode that is accessed upon a successful CLI login authentication. Any user level can access this mode and thus, the commands supported by this command tier are limited, as is interaction with the device itself. This mode allows you to view various information (using the show commands) and activate various debugging capabilities.

```
Welcome to AudioCodes CLI
Username: Admin
Password:
>
```

The Basic mode prompt is ">".

- **Enable command mode:** This mode is the high-level tier in the command hierarchy, one step up from the Basic Mode. A password ("Admin", by default) is required to access this mode after you have accessed the Basic mode. This mode allows you to configure all the device's settings. The Enable mode is accessed by typing the following commands:

```
> enable
Password: <password>
#
```

The Enable mode prompt is "#".

The Enable mode groups the configuration commands under the following command sets:

- **config-system:** Provides the general and system related configuration commands, for example, Syslog configuration. This set is accessed by typing the following command:

```
# configure system
(config-system)#
```

- **config-voip:** Provides the VoIP-related configuration commands, for example, SIP and media parameters, and VoIP network interface configuration. This set is accessed by typing the following command:

```
# configure voip
(config-voip)#
```

8.4 Using CLI Shortcuts

The CLI provides several editing shortcut keys to help you configure your device more easily, as listed in the table below.

Table 13: CLI Editing Shortcut keys

Shortcut Key	Description
Up arrow key	Retypes the previously entered command. Continuing to press the **Up** arrow key cycles through all commands entered, starting with the most recent command.
<Tab> key	Pressing the **<Tab>** key after entering a partial (but unique) command automatically completes the command, displays it on the command prompt line, and waits for further input. Pressing the **<Tab>** key after entering a partial and not unique command displays all completing options.

Shortcut Key	Description

? (question mark)	▪ Displays a list of all subcommands in the current mode, for example:
	`(config-voip)# voip-network ?`
	`  dns             Enter voip-network dns`
	`  ip-group IP Group table`
	`  nat-translation      NATTranslationtable`
	`...`
	▪ Displays a list of available commands beginning with certain letter(s), for example:
	`(config)# voip-network d?`
	`  dns             Enter voip-network dns`
	▪ Displays syntax help for a specific command by entering the command, a space, and then a question mark (?). This includes the range of valid values and a brief description of the next parameter expected for that particular command. For example:
	`(config)# voip-network dns srv2ip ?`
	`  [0-9]        index`
	If a command can be invoked (i.e., all its arguments have been entered), the question mark at its end displays "<cr>" to indicate that a carriage return (Enter) can now be entered to run the command, for example:
	`(config)# logging host 10.1.1.1 ?`
	`  <cr>`

<**Ctrl + A**>	Moves the cursor to the beginning of the command line.
<**Ctrl + E**>	Moves the cursor to the end of the command line.
<**Ctrl + U**>	Deletes all the characters on the command line.
auto finish	You need only enter enough letters to identify a command as unique. For example, entering "int G 0/0" at the configuration prompt provides you access to the configuration parameters for the specified Gigabit-Ethernet interface. Entering "interface GigabitEthernet 0/0" would work as well, but is not necessary.
Space Bar at the --More-- prompt	Displays the next screen of output. You can configure the size of the displayed output, as described in "Configuring Displayed Output Lines in CLI Terminal Window" on page 63.

8.5 Common CLI Commands

The following table contains descriptions of common CLI commands.

Table 14: Common CLI Commands

Command	Description

do	Provides a way to execute commands in other command sets without taking the time to exit the current command set. The following example shows the **do** command, used to view the GigabitEthernet interface configuration while in the virtual-LAN interface command set:
	`(config)# interface vlan 1`
	`(conf-if-VLAN 1)# do show interfaces GigabitEthernet 0/0`

no	Undoes an issued command or disables a feature. Enter **no** before the command: `# no debug log`
activate	Activates a command. When you enter a configuration command in the CLI, the command is not applied until you enter the **activate** and **exit** commands. **Note:** Offline configuration changes require a reset of the device. A reset can be performed at the end of the configuration changes. A required reset is indicated

Command	Description
	by an asterisk (*) before the command prompt.

exit	Leaves the current command-set and returns one level up. If issued on the top level, the session ends.
	For online parameters, if the configuration was changed and no **activate** command was entered, the **exit** command applies the **activate** command automatically. If issued on the top level, the session will end:
	`(config)# exit`
	`# exit`
	`(session closed)`

display	Displays the configuration of current configuration set.
help	Displays a short help how-to string.
history	Displays a list of previously run commands.
list	Displays the available command list of the current command-set.

| <filter>	Applied to a command output. The filter should be typed after the command with a pipe mark (|).
	Supported filters:
	▪ **include <word>** – filter (print) lines which contain <word>
	▪ **exclude <word>** – filter lines which does not contain <word>
	▪ **grep <options>** - filter lines according to *grep* common Unix utility options
	▪ **egrep <options>** - filter lines according to e*grep* common Unix utility options
	▪ **begin <word>** – filter (print) lines which begins with <word>
	▪ **between <word1> <word2>** – filter (print) lines which are placed between <word1> and <word2>
	▪ **count** – show the output's line count
	Example:
	`# show system version | grep Number`
	`;Serial Number: 2239835;Slot Number: 1`

8.6 Configuring Tables in CLI

Throughout the CLI, many configuration elements are in table format, where each table row is represented by an index number. When you add a new row to a table, the device automatically assigns it the next consecutive, available index number. You can also specify an index number, if required. When you add a new table row, the device accesses the row's configuration mode.

Table rows are added using the **new** command:

```
# <table name> new
```

For example, if three rows are configured in the Account table (account-0, account-1, and account-2) and a new entry is subsequently added, account-3 is automatically created and its configuration mode is accessed:

```
(config-voip)# sip-definition account new
(account-3)#
```

You can also add a new table row to any specific index number, even if a row has already been configured for that index number. The row that was previously assigned that index number is subsequently incremented to the next index number, as well as all the index rows listed further down in the table.

To add a new table row to a specific index number, use the **insert** command:

```
# <table name> <index> insert
```

For example, if three rows are configured in the Account table (account-0, account-1, and account-2) and a new row is subsequently added with index 1, the previous account-1 becomes account-2 and the previous account-2 becomes account-3, and so on. The following command is run for this example:

```
(config-voip)# sip-definition account 1 insert
```

Note: This behavior when inserting table rows is applicable only to tables that do not have "child" tables (sub-tables).

8.7 Understanding CLI Error Messages

The CLI provides feedback on commands by displaying informative messages:

- Failure reason of a run command. The failure message is identical to the notification failure message sent by Syslog. For example, an invalid Syslog server IP address is displayed in the CLI as follows:

```
(logging)# syslog-ip 1111.1.1.1
Parameter 'SyslogServerIP' does NOT accept the IP-Address: 1111.1.1.1, illegal
IPAddress.
Configuration failed
Command Failed!
```

- "Invalid command" message: The command may not be valid in the current command mode, or you may not have entered sufficient characters for the command to be recognized. Use "?" to determine your error.
- "Incomplete command" message: You may not have entered all of the pertinent information required to make the command valid. Use "?" to determine your error.

8.8 Configuring Maximum Telnet/SSH Sessions

You can set the maximum (up to five) number of concurrent Telnet/SSH sessions permitted on the device.

Note: Before changing this setting, make sure that not more than this number of sessions are currently active; otherwise, the new setting will not take effect.

➢ To configure the maximum number of concurrent Telnet/SSH sessions:

1. Open the Telnet/SSH Settings page (**Configuration** tab > **System** menu > **Management** > **Telnet/SSH Settings**).
2. In the 'Maximum Telnet Sessions' field, enter the maximum number of concurrent sessions.
3. Click **Submit**.

8.9 Viewing Current CLI Sessions

You can view users that are currently logged in to the device's CLI. This applies to users logged in to the CLI through RS-232 (console), Telnet, or SSH. For each logged-in user, the following is displayed: the type of interface (console, Telnet, or SSH), user's username, remote IP address from where the user logged in, and the duration (days and time) of the session. Each user is displayed with a unique index (session ID).

➢ To view currently logged-in CLI users:

```
# show users
[0]  console     Admin      local             0d00h03m15s
[1]  telnet      John       10.4.2.1          0d01h03m47s
[2]* ssh         Alex       192.168.121.234   12d00h02m34s
```

The current session from which this show command was run is displayed with an asterisk (*).

Note: The device can display management sessions of up to 24 hours. After this time, the duration counter is reset.

8.10 Terminating a User's CLI Session

You can terminate users that are currently logged in to the device's CLI. This applies to users logged in to the CLI through RS-232 (console), Telnet, or SSH.

➢ To terminate the CLI session of a specific CLI user:

```
# clear user <session ID>
```

The *session ID* is a unique identification of each currently logged in user. You can view the session ID by running the **show users** command (see "Viewing Current CLI Sessions" on page 62).

Note: The session from which the command is run cannot be terminated.

8.11 Configuring Displayed Output Lines in CLI Terminal Window

You can configure the maximum number of lines (height) displayed in the terminal window for the output of CLI commands (Telnet and SSH). The number of displayed lines can be specified from 0 to 65,535, or determined by re-sizing the terminal window by mouse-dragging the window's border.

➢ To configure a specific number of output lines:

```
(config-system)# cli-terminal
<cli-terminal># window-height [0-65535]
```

If window-height is set to 0, the entire command output is displayed. In other words, even if the output extends beyond the visible terminal window length, the --MORE-- prompt is not displayed.

➢ To configure the number of lines according to dragged terminal window:

```
(config-system)# cli-terminal
<cli-terminal># window-height automatic
```

When this mode is configured, each time you change the height of the terminal window using your mouse (i.e., dragging one of the window's borders or corners), the number of displayed output command lines is changed accordingly.

Reader's Notes

9 SNMP-Based Management

The device provides an embedded SNMP Agent that allows it to be managed by AudioCodes Element Management System (EMS) or a third-party SNMP Manager (e.g., element management system). The SNMP Agent supports standard Management Information Base (MIBs) and proprietary MIBs, enabling a deeper probe into the interworking of the device. The SNMP Agent can also send unsolicited events (SNMP traps) towards the SNMP Manager. All supported MIB files are supplied to customers as part of the release.

AudioCodes EMS is an advanced solution for standards-based management that covers all areas vital for the efficient operation, administration, management and provisioning (OAM&P) of the device. The standards-compliant EMS uses distributed SNMP-based management software, optimized to support day-to-day Network Operation Center (NOC) activities, offering a feature-rich management framework. It supports fault management, configuration and security.

This section provides configuration relating to SNMP management.

Notes:

- SNMP-based management is enabled by default. For disabling it, see "Enabling SNMP and Configuring SNMP Community Strings" on page 65.
- For more information on the device's SNMP support (e.g., SNMP traps), refer to the *SNMP User's Guide*.
- EMS support is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 309.
- For more information on using the EMS tool, refer to the *EMS User's Manual* and *EMS Server IOM Manual*.

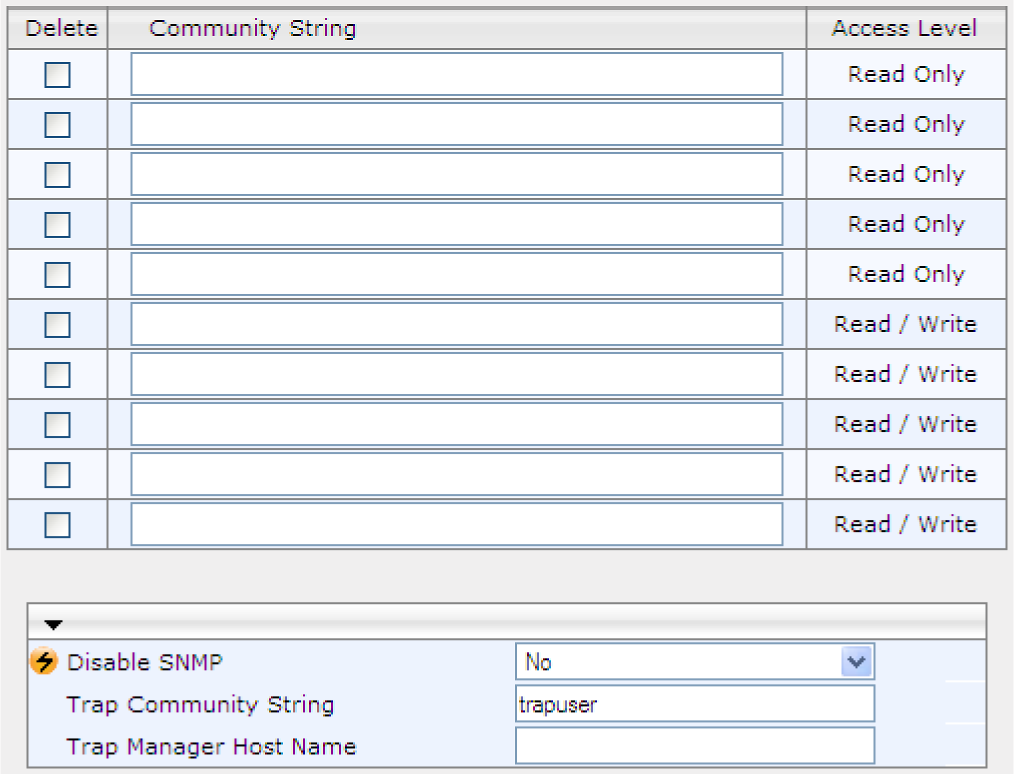
9.1 Enabling SNMP and Configuring SNMP Community Strings

The SNMP Community String page lets you configure up to five read-only and up to five read-write SNMP community strings and to configure the community string that is used for sending traps.

For detailed descriptions of the SNMP parameters, see "SNMP Parameters" on page 385.

➢ To configure SNMP community strings:

1. Open the SNMP Community String page (**Configuration** tab > **System** menu > **Management** > **SNMP** > **SNMP Community String**).



Delete	Community String	Access Level
□		Read Only
□		Read Only
□		Read Only
□		Read Only
□		Read Only
□		Read / Write
□		Read / Write
□		Read / Write
□		Read / Write
□		Read / Write

Disable SNMP	No
Trap Community String	trapuser
Trap Manager Host Name	

2. Configure SNMP community strings according to the table below.
3. Click **Submit**, and then save ("burn") your settings to flash memory.

To delete a community string, select the **Delete** check box corresponding to the community string that you want to delete, and then click **Submit**.

Table 15: SNMP Community String Parameter Descriptions

Parameter	Description
Community String	▪ Read Only **[SNMPReadOnlyCommunityString_x]:** Up to five read-only community strings (up to 19 characters each). The default string is 'public'. ▪ Read / Write **[SNMPReadWriteCommunityString_x]:** Up to five read / write community strings (up to 19 characters each). The default string is 'private'.
Trap Community String CLI: configure system > snmp trap > community-string **[SNMPTrapCommunityString]**	Community string used in traps (up to 19 characters). The default string is 'trapuser'.

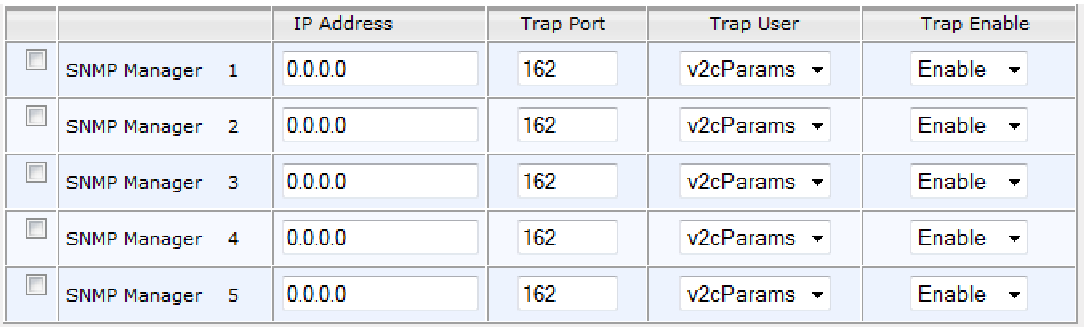
9.2 Configuring SNMP Trap Destinations

The SNMP Trap Destinations page allows you to configure up to five SNMP trap managers. You can associate a trap destination with SNMPv2 users and specific SNMPv3 users. Associating a trap destination with SNMPv3 users sends encrypted and authenticated traps to the SNMPv3 destination. By default, traps are sent unencrypted using SNMPv2.

➢ To configure SNMP trap destinations:

1. Open the SNMP Trap Destinations page (**Configuration** tab > **System** menu > **Management** > **SNMP** > **SNMP Trap Destinations**).

Figure 25: SNMP Trap Destinations Page



		IP Address	Trap Port	Trap User	Trap Enable
□	SNMP Manager 1	0.0.0.0	162	v2cParams	Enable
□	SNMP Manager 2	0.0.0.0	162	v2cParams	Enable
□	SNMP Manager 3	0.0.0.0	162	v2cParams	Enable
□	SNMP Manager 4	0.0.0.0	162	v2cParams	Enable
□	SNMP Manager 5	0.0.0.0	162	v2cParams	Enable

2. Configure the SNMP trap manager parameters according to the table below.
3. Select the check box corresponding to the SNMP Manager that you wish to enable.
4. Click **Submit**.

Note: Only row entries whose corresponding check boxes are selected are applied when clicking **Submit**; otherwise, settings revert to their defaults.

Table 16: SNMP Trap Destinations Parameters Description

Parameter	Description
Web: SNMP Manager **[SNMPManagerIsUsed_x]**	Enables the SNMP Manager to receive traps and checks the validity of the configured destination (IP address and port number). ▪ **[0]** (check box cleared) = (Default) Disables SNMP Manager ▪ **[1]** (check box selected) = Enables SNMP Manager
Web: IP Address **[SNMPManagerTableIP_x]**	Defines the IP address (in dotted-decimal notation, e.g., 108.10.1.255) of the remote host used as the SNMP Manager. The device sends SNMP traps to this IP address.
Trap Port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port. The valid value range is 100 to 4000. The default is 162.
Web: Trap User **[SNMPManagerTrapUser]**	Associates a trap user with the trap destination. This determines the trap format, authentication level, and encryption level. ▪ v2cParams (default) = SNMPv2 user community string ▪ SNMPv3 user configured in "Configuring SNMP V3 Users" on page 68
Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Activates the sending of traps to the SNMP Manager. ▪ **[0]** Disable ▪ **[1]** Enable (Default)

9.3 Configuring SNMP Trusted Managers

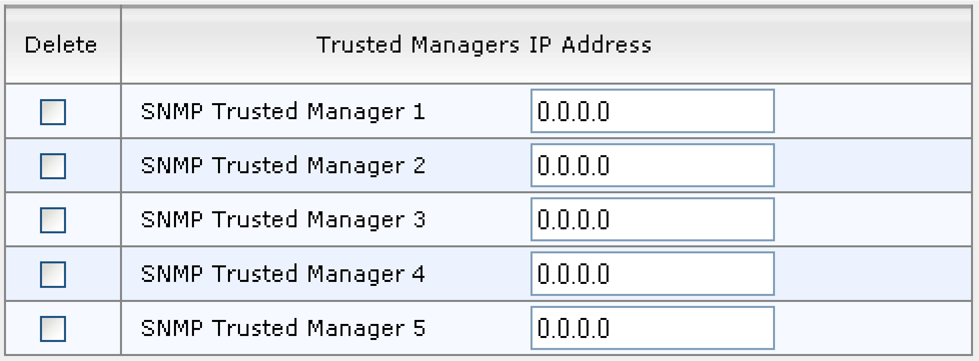
The SNMP Trusted Managers table lets you configure up to five SNMP Trusted Managers based on IP addresses. By default, the SNMP agent accepts SNMP Get and Set requests from any IP address as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests.

The procedure below describes how to configure SNMP trusted managers in the Web interface. You can also configure this using the table ini file parameter, SNMPTrustedMgr_x or CLI command, configure system > snmp > trusted-managers.

➢ To configure SNMP Trusted Managers:

1. Open the SNMP Trusted Managers page (**Configuration** tab > **System** menu > **Management** > **SNMP** > **SNMP Trusted Managers**).

Figure 26: SNMP Trusted Managers



Delete	Trusted Managers IP Address	
□	SNMP Trusted Manager 1	0.0.0.0
□	SNMP Trusted Manager 2	0.0.0.0
□	SNMP Trusted Manager 3	0.0.0.0
□	SNMP Trusted Manager 4	0.0.0.0
□	SNMP Trusted Manager 5	0.0.0.0

2. Select the check box corresponding to the SNMP Trusted Manager that you want to enable and for whom you want to define an IP address.
3. Define an IP address in dotted-decimal notation.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

9.4 Configuring SNMP V3 Users

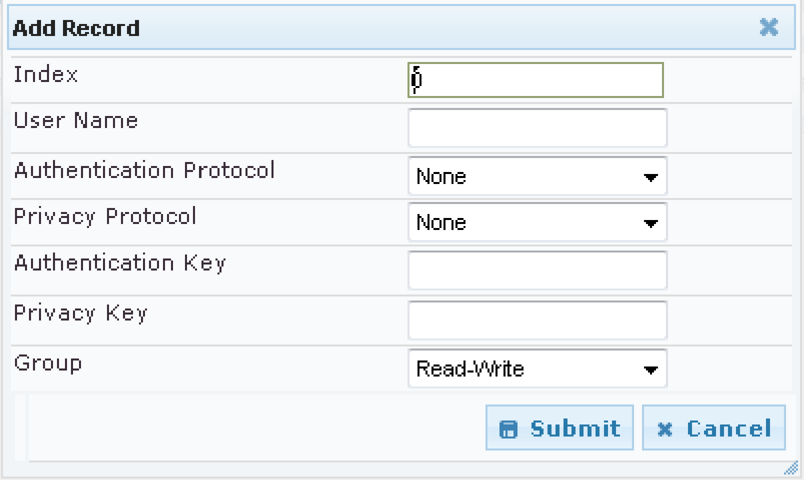
The SNMP v3 Users table lets you configure up to 10 SNMP v3 users for authentication and privacy.

The procedure below describes how to configure SNMP v3 users in the Web interface. You can also configure this using the table ini file parameter, SNMPUsers or CLI command, configure system > snmp v3-users.

➢ To configure an SNMP v3 user:

1. Open the SNMP v3 Users page (**Configuration** tab > **System** menu > **Management** > **SNMP** > **SNMP V3 Users**).
2. Click **Add**; the following dialog box appears:

Figure 27: SNMP V3 Setting Page - Add Record Dialog Box



Add Record

Index	0
User Name	
Authentication Protocol	None
Privacy Protocol	None
Authentication Key	
Privacy Key	
Group	Read-Write

3. Configure the SNMP V3 Setting parameters according to the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Note: If you delete a user that is associated with a trap destination (see "Configuring SNMP Trap Destinations" on page 66), the configured trap destination becomes disabled and the trap user reverts to default (i.e., SNMPv2).

Table 17: SNMP V3 Users Parameters

Parameter	Description
Index **[SNMPUsers_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
User Name CLI: username **[SNMPUsers_Username]**	Name of the SNMP v3 user. This name must be unique.
Authentication Protocol CLI: auth-protocol **[SNMPUsers_AuthProtocol]**	Authentication protocol of the SNMP v3 user. ▪ **[0]** None (default) ▪ **[1]** MD5 ▪ **[2]** SHA-1
Privacy Protocol CLI: priv-protocol **[SNMPUsers_PrivProtocol]**	Privacy protocol of the SNMP v3 user. ▪ **[0]** None (default) ▪ **[1]** DES ▪ **[2]** 3DES ▪ **[3]** AES-128 ▪ **[4]** AES-192 ▪ **[5]** AES-256
Authentication Key CLI: auth-key **[SNMPUsers_AuthKey]**	Authentication key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Privacy Key CLI: priv-key **[SNMPUsers_PrivKey]**	Privacy key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Group CLI: group **[SNMPUsers_Group]**	The group with which the SNMP v3 user is associated. ▪ **[0]** Read-Only (default) ▪ **[1]** Read-Write ▪ **[2]** Trap **Note:** All groups can be used to send traps.

10 INI File-Based Management

The device can be configured using an ini file, which is a text-based file with an *ini* file extension name that can be created using any standard text-based editor such as Notepad. Each configuration element of the device has a corresponding ini file parameter that you can use in the ini file for configuring the device. When you have created the ini file with your ini file parameter settings, you apply these settings to the device by installing (loading) the ini file to the device.

Notes:

- For a list and description of the *ini* file parameters, see "Configuration Parameters Reference" on page 380.
- To restore the device to default settings using the *ini* file, see "Restoring Factory Defaults" on page 327.

10.1 INI File Format

The *ini* file can be configured with any number of parameters. These *ini* file parameters can be one of the following types:

- Individual parameters - see "Configuring Individual ini File Parameters" on page 70
- Table parameters - see "Configuring Table ini File Parameters" on page 70

10.1.1 Configuring Individual ini File Parameters

The syntax for configuring individual *ini* file parameters in the ini file is as follows:

- An optional, subsection name (or group name) enclosed in square brackets "[...]". This is used to conveniently group similar parameters by their functionality.
- Parameter name, followed by an equal "=" sign and then its value.
- Comments must be preceded by a semicolon ";".

```
[subsection name]
parameter name = value
parameter name = value
; this is a comment line
; for example:
[System Parameters]
SyslogServerIP = 10.13.2.69
EnableSyslog = 1
```

For general *ini* file formatting rules, see "General ini File Formatting Rules" on page 72.

10.1.2 Configuring Table ini File Parameters

The table ini file parameters allow you to configure tables, which include multiple parameters (*columns*) and row entries (*indices*). When loading an *ini* file to the device, it's recommended to include only tables that belong to applications that are to be configured (dynamic tables of other applications are empty, but static tables are not).

The table ini file parameter is composed of the following elements:

- **Title of the table:** The name of the table in square brackets, e.g., [MY_TABLE_NAME].
- **Format line:** Specifies the columns of the table (by their string names) that are to be configured.
  - The first word of the Format line must be "FORMAT", followed by the Index field name and then an equal "=" sign. After the equal sign, the names of the columns are listed.

    - Columns must be separated by a comma ",".
    - The Format line must only include columns that can be modified (i.e., parameters that are not specified as read-only). An exception is Index fields, which are mandatory.
    - The Format line must end with a semicolon ";".

- **Data line(s):** Contain the actual values of the columns (parameters). The values are interpreted according to the Format line.
    - The first word of the Data line must be the table's string name followed by the Index field.
    - Columns must be separated by a comma ",".
    - A Data line must end with a semicolon ";".
- **End-of-Table Mark:** Indicates the end of the table. The same string used for the table's title, preceded by a backslash "\", e.g., [\MY_TABLE_NAME].

The following displays an example of the structure of a table ini file parameter.

```
[Table Title]
; This is the title of the table.
FORMAT Index = Column Name1, Column Name2, Column Name3;
; This is the Format line.
Index 0 = value1, value2, value3;
Index 1 = value1, $$, value3;
; These are the Data lines.
[\Table Title]
; This is the end-of-the-table-mark.
```

The table ini file parameter formatting rules are listed below:

- Indices (in both the Format and the Data lines) must appear in the same order. The Index field must never be omitted.
- The Format line can include a subset of the configurable fields in a table. In this case, all other fields are assigned with the pre-defined default values for each configured line.
- The order of the fields in the Format line isn't significant (as opposed to the Index fields). The fields in the Data lines are interpreted according to the order specified in the Format line.
- The double dollar sign ($$) in a Data line indicates the default value for the parameter.
- The order of the Data lines is insignificant.
- Data lines must match the Format line, i.e., it must contain exactly the same number of Indices and Data fields and must be in exactly the same order.
- A row in a table is identified by its table name and Index field. Each such row may appear only once in the *ini* file.
- Table dependencies: Certain tables may depend on other tables. For example, one table may include a field that specifies an entry in another table. This method is used to specify additional attributes of an entity, or to specify that a given entity is part of a larger entity. The tables must appear in the order of their dependency (i.e., if Table X is referred to by Table Y, Table X must appear in the *ini* file before Table Y).

For general *ini* file formatting rules, see "General ini File Formatting Rules" on page 72.

The table below displays an example of a table ini file parameter:

```
[ CodersGroup0 ]
FORMAT CodersGroup0 Index = CodersGroup0 Name, CodersGroup0 pTime,
CodersGroup0 rate, CodersGroup0 PayloadType, CodersGroup0 Sce;
CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0;
[ \CodersGroup0 ]
```

Note: Do not include read-only parameters in the table ini file parameter as this can cause an error when attempting to load the file to the device.

10.1.3 General ini File Formatting Rules

The *ini* file must adhere to the following formatting rules:

- The *ini* file name must not include hyphens "-" or spaces; if necessary, use an underscore "_" instead.
- Lines beginning with a semi-colon ";" are ignored. These can be used for adding remarks in the *ini* file.
- A carriage return (i.e., Enter) must be done at the end of each line.
- The number of spaces before and after the equals sign "=" is irrelevant.
- Subsection names for grouping parameters are optional.
- If there is a syntax error in the parameter name, the value is ignored.
- Syntax errors in the parameter's value can cause unexpected errors (parameters may be set to the incorrect values).
- Parameter string values that denote file names (e.g., CallProgressTonesFileName) must be enclosed with inverted commas, e.g., CallProgressTonesFileName = 'cpt_usa.dat'.
- The parameter name is not case-sensitive.
- The parameter value is not case-sensitive, except for coder names.
- The *ini* file must end with at least one carriage return.

10.2 Configuring an ini File

There are different methods that you can use for configuring the ini file before you load it to the device.

- Modifying the device's current ini file. This method is recommended if you mainly need to change the settings of parameters that you have previously configured.
  1. Save the device's current configuration as an *ini* file on your computer, using the Web interface (see "Saving Configuration" on page 295).
  2. Open the file using a text file editor, and then modify the *ini* file as required.
  3. Save and close the file.
  4. Load the file to the device.
- Creating a new ini file that includes only updated configuration:
  1. Open a text file editor such as Notepad.
  2. Add only the required parameters and their settings.
  3. Save the file with the ini file extension name (e.g., myconfiguration.ini).
  4. Load the file to the device.

For loading the ini file to the device, see "Loading an ini File to the Device" on page 72.

Note: To restore the device to default settings using the *ini* file, see "Restoring Factory Defaults" on page 327.

10.3 Loading an ini File to the Device

You can load an *ini* file to the device using the following methods:

- CLI:

- Voice Configuration: # copy voice-configuration from <URL>

■ Web interface:

- Load Auxiliary Files page (see "Loading Auxiliary Files" on page 299): The device updates its configuration according to the loaded ini file, while preserving the remaining current configuration.
- Configuration File page (see "Backing Up and Loading Configuration File" on page 316): The device updates its configuration according to the loaded ini file, and applies default values to parameters that were not included in the loaded ini file. Thus, all previous configuration is overridden.

When you load an ini file to the device, its configuration settings are saved to the device's non-volatile memory.

Note: Before you load an *ini* file to the device, make sure that the file extension name is *.ini*.

10.4 Secured Encoded ini File

The *ini* file contains sensitive information that is required for the functioning of the device. The file may be loaded to the device using HTTP. These protocols are not secure and are vulnerable to potential hackers. To overcome this security threat, the AudioCodes DConvert utility allows you to binary-encode (encrypt) the *ini* file before loading it to the device. For more information, refer to the *DConvert Utility User's Guide*.

Note: If you save an ini file from the device to a folder on your PC, an *ini* file that was loaded to the device encoded is saved as a regular *ini* file (i.e., unencoded).

10.5 Configuring Password Display in ini File

Passwords can be displayed in the ini file in one of the following formats, configured by the INIPasswordsDisplayType ini file parameter:

- Obscured: The password characters are concealed and displayed as encoded. The password is displayed using the syntax, *$1$<obscured password>*, for example, $1$S3p+fno=.
- Hidden: the password is replaced with an asterisk (*).

When you save an ini file from the device to a PC, the passwords are displayed according to the enabled format. When you load an ini file to the device, obscured passwords are parsed and applied to the device; hidden passwords are ignored.

By default, the enabled format is obscured passwords, thus enabling their full recovery in case of configuration restore or copy to another device.

When obscured password mode is enabled, you can enter a password in the ini file using any of the following formats:

- $1$<obscured password>: Password in obscured format as generated by the device; useful for restoring device configuration and copying configuration from one device to another.
- $0$<plain text>: Password can be entered in plain text; useful for configuring a new password. When the ini file is loaded to the device and then later saved from the device to a PC, the password is displayed obscured (i.e., $1$<obscured password>).

10.6 INI Viewer and Editor Utility

AudioCodes INI Viewer & Editor utility provides a user-friendly graphical user interface (GUI) that lets you easily view and modify the device's ini file. This utility is available from AudioCodes Web site at www.AudioCodes.com/downloads, and can be installed on any Windows-based PC.

For more information, refer to the *INI Viewer & Editor User's Guide*.

Part III

General System Settings

11 Configuring Certificates

The Certificates page allows you to configure X.509 certificates, which are used for secure management of the device, secure SIP transactions, and other security applications.

> **Note:** The device is shipped with an active TLS setup. Thus, configure certificates only if required.

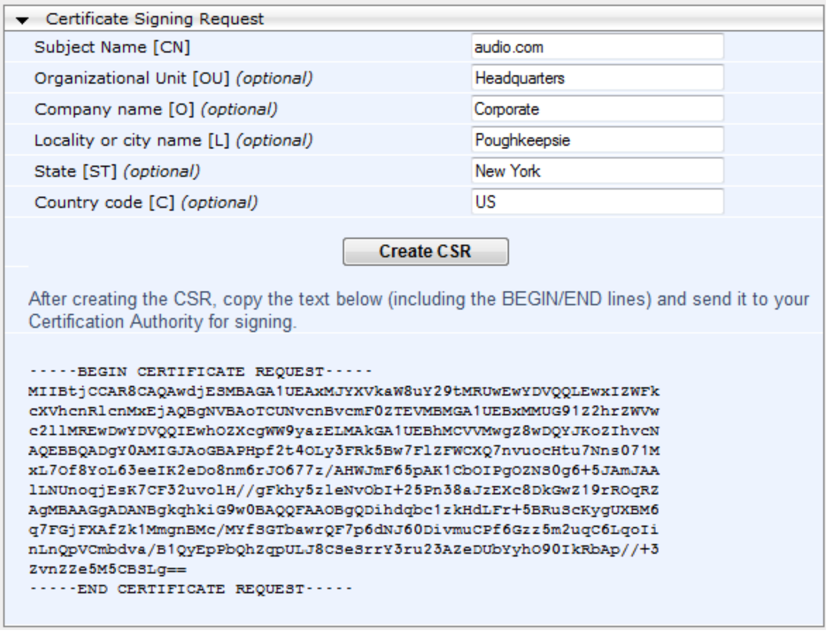
11.1 Replacing the Device's Certificate

The device is supplied with a working TLS configuration consisting of a unique self-signed server certificate. If an organizational Public Key Infrastructure (PKI) is used, you may wish to replace this certificate with one provided by your security administrator.

➢ To replace the device's certificate:

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This DNS name is used to access the device and therefore, must be listed in the server certificate.
2. If the device is operating in HTTPS mode, then set the 'Secured Web Connection (HTTPS)' parameter (HTTPSOnly) to **HTTP and HTTPS** (see "Configuring Web Security Settings" on page 49). This ensures that you have a method for accessing the device in case the new certificate does not work. Restore the previous setting after testing the configuration.
3. Open the Certificates page (**Configuration** tab > **System** menu > **Certificates**).
4. Under the **Certificate Signing Request** group, do the following:
   a. In the 'Subject Name [CN]' field, enter the DNS name.
   b. Fill in the rest of the request fields according to your security provider's instructions.
   c. Click the **Create CSR** button; a textual certificate signing request is displayed in the area below the button:

Figure 28: Certificate Signing Request Group



Subject Name [CN]	audio.com
Organizational Unit [OU] *(optional)*	Headquarters
Company name [O] *(optional)*	Corporate
Locality or city name [L] *(optional)*	Poughkeepsie
State [ST] *(optional)*	New York
Country code [C] *(optional)*	US

After creating the CSR, copy the text below (including the BEGIN/END lines) and send it to your Certification Authority for signing.

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBtjCCAR8CAQAwdjESMBAGA1UEAxMJYXVkaW8uY29tMRUwEwYDVQQLEwxIZWFk
cXVhcnRlcnMxEjAQBgNVBAoTCUNvcnBvcmF0ZTEVMBMGA1UEBxMMUG91z2hrZWVw
c2llMREwDwYDVQQIEwhOZXcgWW9yazELMAkGA1UEBhMCVVMwgZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBAPHpf2t4OLy3FRk5Bw7FlZFWCXQ7nvuocHtu7Nns071M
xL7Of8YoL63eeIK2eDo8nm6rJO677z/AHWJmF65pAK1CbOIPgOZNS0g6+5JAmJAA
lLNUnoqjEsK7CF32uvolH//gFkhy5zleNvObI+25Pn38aJzEXc8DkGwZ19rROqRZ
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQDihdqbc1zkHdLFr+5BRuScKygUXBM6
q7FGjFXAfZk1MmgnBMc/MYfSGTbawrQF7p6dNJ60DivmuCPf6Gzz5m2uqC6LqoIi
nLnQpVCmbdva/B1QyEpPbQhZqpULJ8CSeSrrY3ru23AZeDUbYyhO90IkRbAp//+3
ZvnZZe5M5CBSLg==
-----END CERTIFICATE REQUEST-----
```

5. Copy the text and send it to your security provider. The security provider, also known as Certification Authority or CA, signs this request and then sends you a server certificate for the device.
6. Save the certificate to a file (e.g., cert.txt). Ensure that the file is a plain-text file containing

the"'BEGIN CERTIFICATE" header, as shown in the example of a Base64-Encoded X.509 Certificate below:

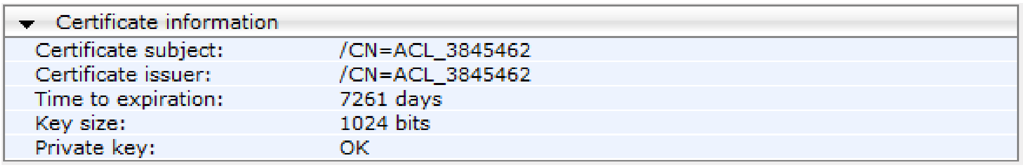
```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEwJGUjETMBEGA1UEChMKQ2Vy
dGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBTZXJ2ZXVyMB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4
MDAwMFowPzELMAkGA1UEBhMCRlIxEzARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9zdGUg
U2VydmV1cjCCASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4MziR4spWldGRx8bQrhZkonWnNm`+
Yhb7+4Q67ecf1janH7GcN/SXsfx7jJpreWULf7v7Cvpr4R7qIJcmdHIntmf7JPM5n6cDBv17uSW63er7NkVn
MFHwK1QaGFLMybFkzaeGrvFm4k3lRefiXDmuOe+FhJgHYezYHf44LvPRPwhSrzi9+Aq3o8pWDguJuZDIUP1F
1jMa+LPwvREXfFcUW+w==
-----END CERTIFICATE-----
```

7. Scroll down to the **Upload certificates files from your computer** group, click the **Browse** button corresponding to the 'Send Device Certificate...' field, navigate to the cert.txt file, and then click **Send File**.

8. After the certificate successfully loads to the device, save the configuration with a device reset (see "Saving Configuration" on page 295); the Web interface uses the provided certificate.

9. Open the Certificates page again and verify that under the **Certificate information** group (at the top of the page), the 'Private key' read-only field displays "OK"; otherwise, consult your security administrator:

Figure 29: Private key "OK" in Certificate Information Group



Certificate information

Certificate subject:	/CN=ACL_3845462
Certificate issuer:	/CN=ACL_3845462
Time to expiration:	7261 days
Key size:	1024 bits
Private key:	OK

10. If the device was originally operating in HTTPS mode and you disabled it in Step 2, then return it to HTTPS by setting the 'Secured Web Connection (HTTPS)' parameter to **HTTPS Only**, and then reset the device with a flash burn.

Notes:

- The certificate replacement process can be repeated when necessary (e.g., the new certificate expires).
- It is possible to use the IP address of the device (e.g., 10.3.3.1) instead of a qualified DNS name in the Subject Name. This is not recommended since the IP address is subject to change and may not uniquely identify the device.
- The device certificate can also be loaded via the Automatic Update Facility by using the HTTPSCertFileName *ini* file parameter.

11.2 Loading a Private Key

The device is shipped with a self-generated random private key, which cannot be extracted from the device. However, some security administrators require that the private key be generated externally at a secure facility and then loaded to the device through configuration. Since private keys are sensitive security parameters, take precautions to load them over a physically-secure connection such as a back-to-back Ethernet cable connected directly to the managing computer.

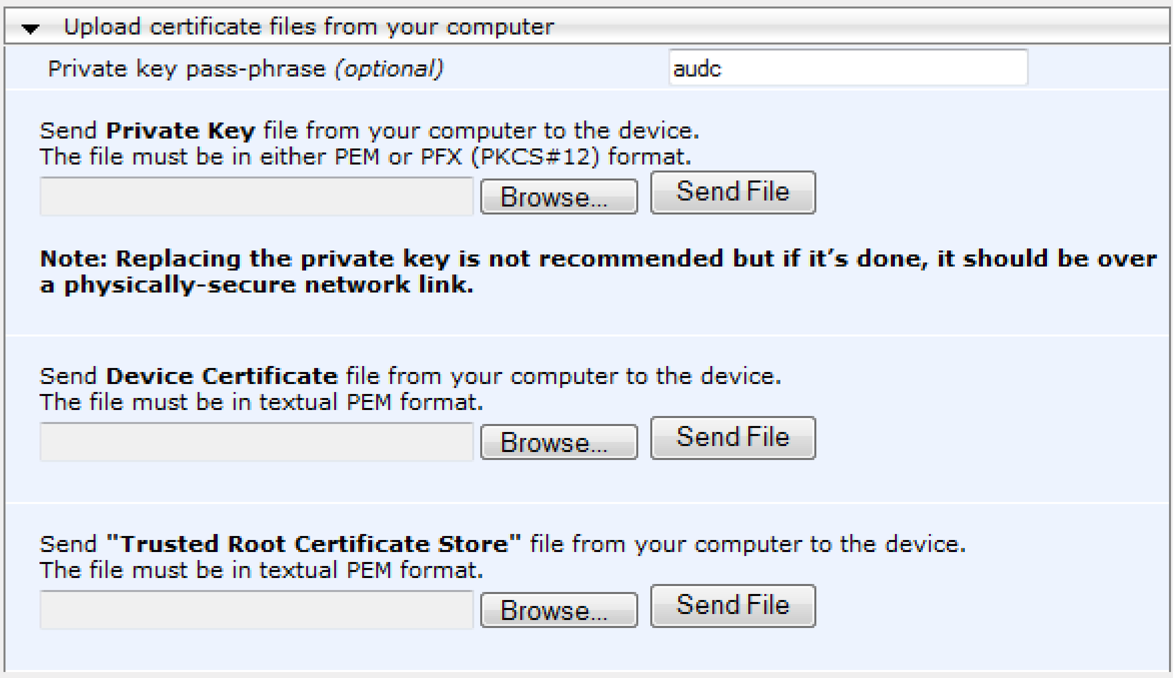
➢ To replace the device's private key:

1. Your security administrator should provide you with a private key in either textual PEM (PKCS #7) or PFX (PKCS #12) format. The file may be encrypted with a short pass-phrase, which should be provided by your security administrator.

2. If the device is operating in HTTPS mode, then set the 'Secured Web Connection (HTTPS)' field (HTTPSOnly) to **HTTP and HTTPS** (see "Configuring Web Security Settings" on page 49). This ensures that you have a method for accessing the device in case the new configuration does not work. Restore the previous setting after testing the configuration.

3. Open the Certificates page (**Configuration** tab > **System** menu > **Certificates**) and scroll down to the **Upload certificate files from your computer** group.

Figure 30: Upload Certificate Files from your Computer Group



▼ Upload certificate files from your computer

Private key pass-phrase *(optional)*: audc

Send **Private Key** file from your computer to the device.
The file must be in either PEM or PFX (PKCS#12) format.

Note: Replacing the private key is not recommended but if it's done, it should be over a physically-secure network link.

Send **Device Certificate** file from your computer to the device.
The file must be in textual PEM format.

Send **"Trusted Root Certificate Store"** file from your computer to the device.
The file must be in textual PEM format.

4. Fill in the 'Private key pass-phrase' field, if required.
5. Click the **Browse** button corresponding to the 'Send Private Key' field, navigate to the key file, and then click **Send File**.
6. If the security administrator has provided you with a device certificate file, load it using the 'Send Device Certificate' field.
7. After the files successfully load to the device, save the configuration with a device reset (see "Saving Configuration" on page 295); the Web interface uses the new configuration.
8. Open the Certificates page again, and verify that under the **Certificate information** group (at the top of the page) the 'Private key' read-only field displays "OK"; otherwise, consult your security administrator.
9. If the device was originally operating in HTTPS mode and you disabled it in Step 2, then enable it by setting the 'Secured Web Connection (HTTPS)' field to **HTTPS Only**.

11.3 Mutual TLS Authentication

By default, servers using TLS provide one-way authentication. The client is certain that the identity of the server is authentic. When an organizational PKI is used, two-way authentication may be desired - both client and server should be authenticated using X.509 certificates. This is achieved by installing a client certificate on the managing PC and loading the root CA's certificate to the device's Trusted Root Certificate Store. The Trusted Root Certificate file may contain more than one CA certificate combined, using a text editor.

Since X.509 certificates have an expiration date and time, the device must be configured to use NTP (see "Simple Network Time Protocol Support" on page 82) to obtain the current date and time. Without the correct date and time, client certificates cannot work.

➢ To enable mutual TLS authentication for HTTPS:

1. Set the 'Secured Web Connection (HTTPS)' field to **HTTPS Only** (see "Configuring Web Security Settings" on page 49) to ensure you have a method for accessing the device in case the client certificate does not work. Restore the previous setting after testing the configuration.
2. Open the Certificates page (see "Replacing the Device's Certificate" on page 76).
3. In the **Upload certificate files from your computer** group, click the **Browse** button corresponding to

the 'Send Trusted Root Certificate Store ...' field, navigate to the file, and then click **Send File**.

4. When the operation is complete, set the 'Requires Client Certificates for HTTPS connection' field to **Enable** (see "Configuring Web Security Settings" on page 49).
5. Save the configuration with a device reset (see "Saving Configuration" on page 295).

When a user connects to the secured Web interface of the device:

- If the user has a client certificate from a CA that is listed in the Trusted Root Certificate file, the connection is accepted and the user is prompted for the system password.
- If both the CA certificate and the client certificate appear in the Trusted Root Certificate file, the user is not prompted for a password (thus, providing a single-sign-on experience - the authentication is performed using the X.509 digital signature).
- If the user does not have a client certificate from a listed CA or does not have a client certificate, the connection is rejected.

Notes:

- The process of installing a client certificate on your PC is beyond the scope of this document. For more information, refer to your operating system documentation, and/or consult your security administrator.
- The root certificate can also be loaded via the Automatic Update facility, using the HTTPSRootFileName *ini* file parameter.
- You can enable the device to check whether a peer's certificate has been revoked by an Online Certificate Status Protocol (OCSP) server (see "Configuring Certificate Revocation Checking (OCSP)" on page 80).

11.4 Self-Signed Certificates

The device is shipped with an operational, self-signed server certificate. The subject name for this default certificate is 'ACL_nnnnnnn', where *nnnnnnn* denotes the serial number of the device. However, this subject name may not be appropriate for production and can be changed while still using self-signed certificates.

➢ To change the subject name and regenerate the self-signed certificate:

1. Before you begin, ensure the following:
   - You have a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device and should therefore, be listed in the server certificate.
   - No traffic is running on the device. The certificate generation process is disruptive to traffic and should be executed during maintenance time.
2. Open the Certificates page (see "Replacing the Device's Certificate" on page 76).
3. In the 'Subject Name [CN]' field, enter the fully-qualified DNS name (FQDN) as the certificate subject, select the desired private key size (in bits), and then click **Generate self-signed**; after a few seconds, a message appears displaying the new subject name.
4. Save the configuration with a device reset (see "Saving Configuration" on page 295) for the new certificate to take effect.

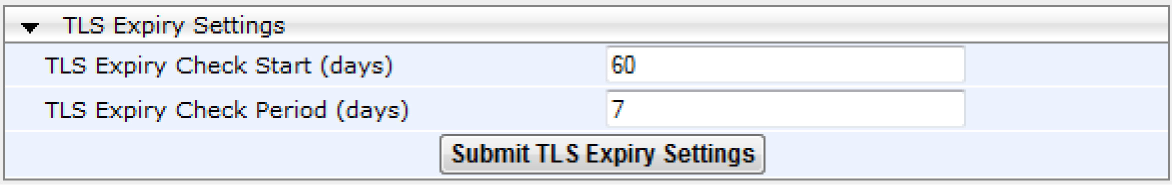
11.5 TLS Server Certificate Expiry Check

The device can periodically check the validation date of the installed TLS server certificate. This periodic check interval is user-defined. In addition, within a user-defined number of days before the installed TLS server certificate expires, the device can be configured to send the SNMP trap, acCertificateExpiryNotifiaction to notify of the impending certificate expiration.

➢ To configure TLS certificate expiry checks and notification:

1. Open the Certificates page (see "Replacing the Device's Certificate" on page 76).
2. In the 'TLS Expiry Check Start' field, enter the number of days before the installed TLS server certificate is to expire at which the device must send a trap to notify of this.

Figure 31: TLS Expiry Settings Group



▾ TLS Expiry Settings

TLS Expiry Check Start (days)	60
TLS Expiry Check Period (days)	7

3. In the 'TLS Expiry Check Period' field, enter the periodical interval (in days) for checking the TLS server certificate expiry date. By default, the device checks the certificate every 7 days.
4. Click the **Submit TLS Expiry Settings** button.

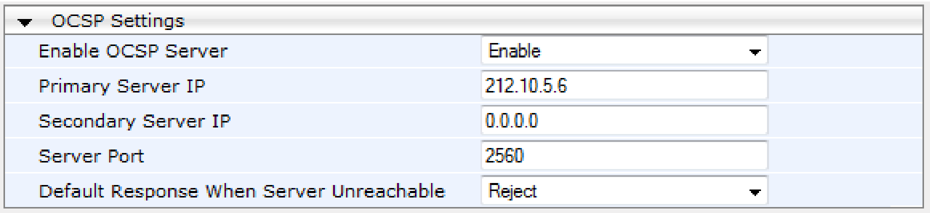
11.6 Configuring Certificate Revocation Checking (OCSP)

Some Public-Key Infrastructures (PKI) can revoke a certificate after it has been issued. You can configure the device to check whether a peer's certificate has been revoked, using the Online Certificate Status Protocol (OCSP). When OCSP is enabled, the device queries the OCSP server for revocation information whenever a peer certificate is received (IPSec, TLS client mode, or TLS server mode with mutual authentication).

➢ To configure OCSP:

1. Open the General Security Settings page (**Configuration** tab > **VoIP** menu > **Security** > **General Security Settings**).

Figure 32: OCSP Parameters



▾ OCSP Settings

Enable OCSP Server	Enable ▾
Primary Server IP	212.10.5.6
Secondary Server IP	0.0.0.0
Server Port	2560
Default Response When Server Unreachable	Reject ▾

2. Configure the OCSP parameters as required. For a description of these parameters, see "OCSP Parameters" on page 414.
3. Click **Submit**.

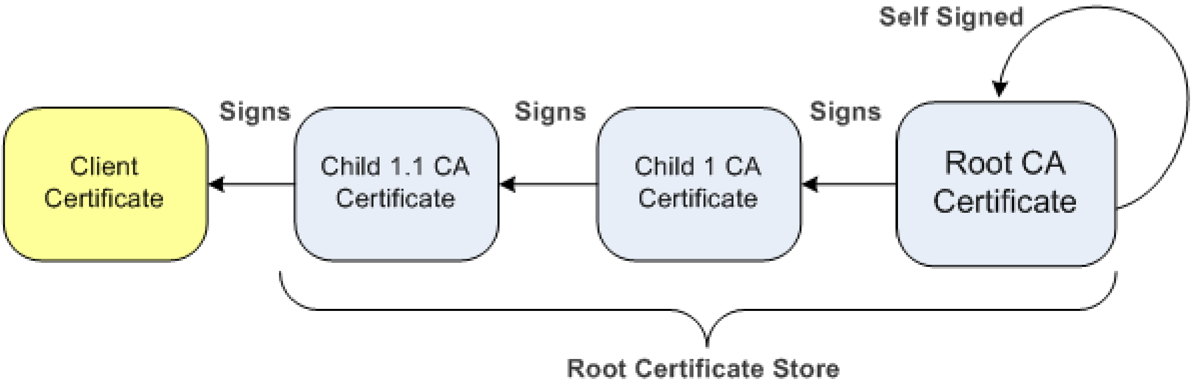
Notes:

- The device does not query OCSP for its own certificate.
- Some PKIs do not support OCSP but generate Certificate Revocation Lists (CRLs). For such cases, set up an OCSP server such as OCSPD.

11.7 Loading Certificate Chain for Trusted Root

A certificate chain is a sequence of certificates where each certificate in the chain is signed by the subsequent certificate. The last certificate in the list of certificates is the Root CA certificate, which is self-signed. The purpose of a certificate chain is to establish a chain of trust from a child certificate to the trusted root CA certificate. The CA vouches for the identity of the child certificate by signing it. A client certificate is considered trusted if one of the CA certificates up the certificate chain is found in the server certificate directory.

Figure 33: Certificate Chain Hierarchy



For the device to trust a whole chain of certificates, you need to combine the certificates into one text file (using a text editor). Once done, upload the file using the 'Trusted Root Certificate Store' field in the Certificates page.

Note: The maximum supported size of the combined file of trusted chain of certificates is 100,000 bytes (including the certificate's headers).

12 Date and Time

The date and time of the device can be configured manually or it can be obtained automatically from a Simple Network Time Protocol (SNTP) server.

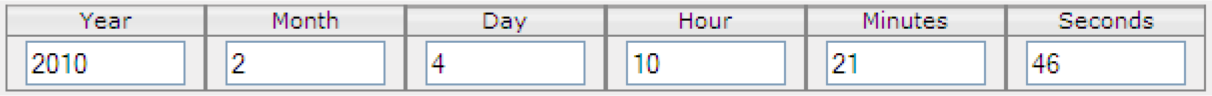
12.1 Configuring Date and Time Manually

The date and time of the device can be configured manually.

➢ To manually configure the device's date and time, using the Web interface:

1. Open the Regional Settings page (**Configuration** tab > **System** menu > **Regional Settings**).

Figure 34: Regional Settings Page



Year	Month	Day	Hour	Minutes	Seconds
2010	2	4	10	21	46

2. Enter the current date and time of the geographical location in which the device is installed.
3. Click **Submit**.

Notes:

- If the device is configured to obtain the date and time from an SNTP server, the fields on this page are read-only, displaying the received date and time.
- After performing a hardware reset, the date and time are returned to their defaults and thus, should be updated.

12.2 Automatic Date and Time through SNTP Server

The Simple Network Time Protocol (SNTP) client functionality generates requests and reacts to the resulting responses using the NTP version 3 protocol definitions (according to RFC 1305). Through these requests and responses, the NTP client synchronizes the system time to a time source within the network, thereby eliminating any potential issues should the local system clock 'drift' during operation. By synchronizing time to a network time source, traffic handling, maintenance, and debugging become simplified for the network administrator.

The NTP client follows a simple process in managing system time: the NTP client requests an NTP update, receives an NTP response, and then updates the local system clock based on a configured NTP server within the network.

The client requests a time update from a specified NTP server at a specified update interval. In most situations, this update interval is every 24 hours based on when the system was restarted. The NTP server identity (as an IP address or FQDN) and the update interval are user-defined, or an SNMP MIB object.

When the client receives a response to its request from the identified NTP server, it must be interpreted based on time zone or location offset that the system is to a standard point of reference called the Universal Time Coordinate (UTC). The time offset that the NTP client uses is configurable.

If required, the clock update is performed by the client as the final step of the update process. The update is performed in such a way as to be transparent to the end users. For instance, the response of the server may indicate that the clock is running too fast on the client. The client slowly robs bits from the clock counter to update the clock to the correct time. If the clock is running too slow, then in an effort to catch the clock up, bits are added to the counter, causing the clock to update quicker and catch up to the correct time. The advantage of this method is that it does not introduce any disparity in the system time that is noticeable to an end user or that could corrupt call timeouts and timestamps.

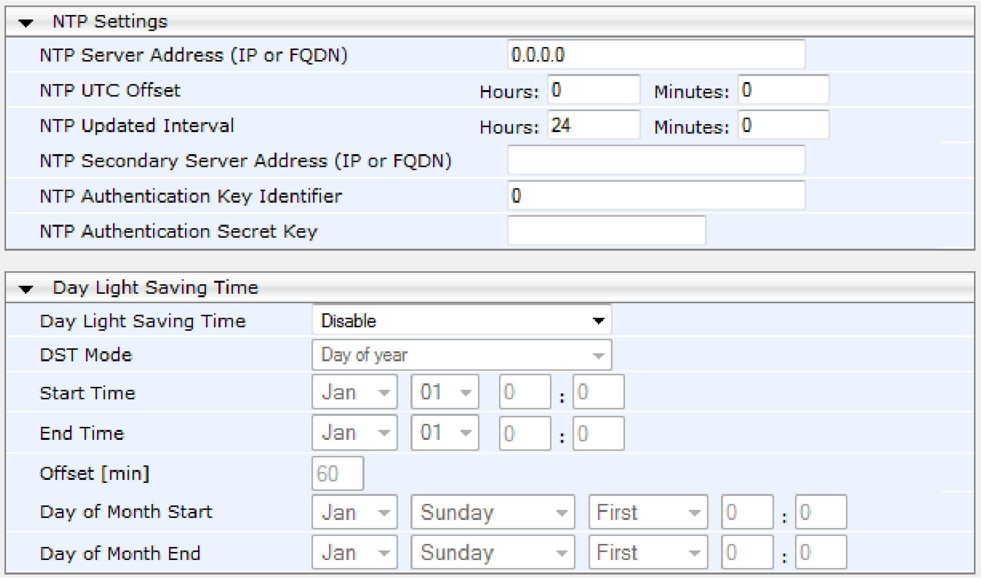
You can also configure the device to authenticate and validate the NTP messages received from the NTP server. Authentication is done using an authentication key with the MD5 cryptographic hash algorithm. When this feature is enabled, NTP messages received without authentication are ignored.

The procedure below describes how to configure SNTP. For detailed descriptions of the configuration parameters, see "NTP and Daylight Saving Time Parameters" on page 397.

➢ To configure SNTP using the Web interface:

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).

Figure 35: SNTP Configuration in Application Settings Page



▼ NTP Settings

NTP Server Address (IP or FQDN)	0.0.0.0
NTP UTC Offset	Hours: 0 Minutes: 0
NTP Updated Interval	Hours: 24 Minutes: 0
NTP Secondary Server Address (IP or FQDN)	
NTP Authentication Key Identifier	0
NTP Authentication Secret Key	

▼ Day Light Saving Time

Day Light Saving Time	Disable
DST Mode	Day of year
Start Time	Jan 01 0 : 0
End Time	Jan 01 0 : 0
Offset [min]	60
Day of Month Start	Jan Sunday First 0 : 0
Day of Month End	Jan Sunday First 0 : 0

2. Configure the NTP parameters:
   - 'NTP Server Address' (NTPServerIP) - defines the NTP server's address (IP or FQDN).
   - 'NTP UTC Offset' (NTPServerUTCOffset) - defines the time offset in relation to the UTC. For example, if your region is 2 hours ahead of the UTC, enter "2".
   - 'NTP Updated Interval' (NTPUpdateInterval) - defines the period after which the date and time of the device is updated.
   - 'NTP Secondary Server Address' (NTPSecondaryServerIP) - defines the secondary NTP server.
3. Configure NTP message authentication:
   - 'NTP Authentication Key Identifier' - defines the NTP authentication key identifier.
   - 'NTP Authentication Secret Key' - defines the secret authentication key shared between the device (client) and the NTP server.

4. Configure daylight saving, if required:
   - 'Day Light Saving Time' (DayLightSavingTimeEnable) - enables daylight saving time.
   - 'DST Mode' - Determines the range type for configuring the start and end date for daylight saving:
     - ♦ **Day of Year:** The range is configured by date of month, for example, from January 4 to August 31.
     - ♦ **Day of month:** The range is configured by day of month, for example, from the second Sunday of May January to the last Sunday of August.
   - 'Start Time' (DayLightSavingTimeStart) and 'End Time' (DayLightSavingTimeEnd) - defines the period for which daylight saving time is relevant.
   - 'Offset' (DayLightSavingTimeOffset) - defines the offset in minutes to add to the time for daylight saving. For example, if your region has daylight saving of one hour, the time received from the NTP server is 11:00, and the UTC offset for your region is +2 (i.e., 13:00), you need to enter "60" to change the local time to 14:00.
5. Verify that the device is set to the correct date and time. You can do this by viewing the date and time in the Regional Settings page, as described in ''Configuring Date and Time Manually'' on page 82.

Reader's Notes

Part IV

General VoIP Configuration

13 Network

This section describes the network-related configuration.

13.1 Configuring Physical Ethernet Ports

The device's physical Ethernet ports are grouped into pairs, termed *Group Members*, where each group consists of an active port and a standby port. This provides port redundancy within a group whereby if an active port is disconnected, the device switches over to the standby port.

These port groups can be assigned to IP network interfaces in the Interface table (see "Configuring IP Network Interfaces" on page 91). This enables physical separation of network interfaces, providing a higher level of segregation of sub-networks. Equipment connected to different physical ports is not accessible to one another. The only connection between them can be established by cross connecting them with media streams (VoIP calls).

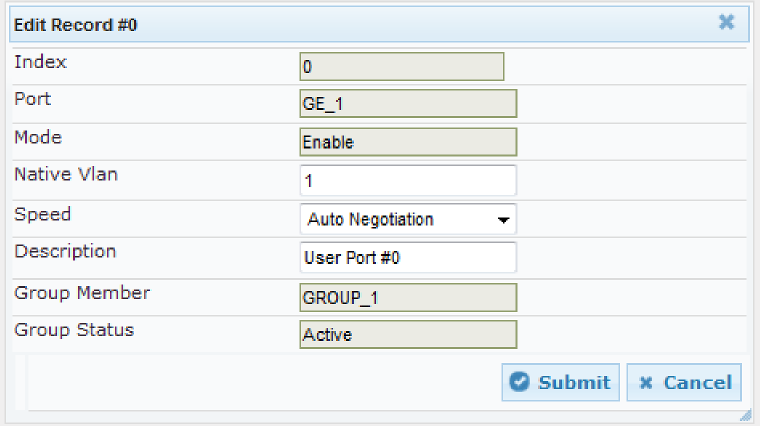
For each port, you can configure the speed, duplex mode, native VLAN (PVID), and provide a brief description. The device supports up to twoport-pair redundancy groups or up to four port groups, where each group is assigned only one port; a combination of port-pair redundancy groups and single-port groups can be configured.

Note: To configure the transmit (Tx) and receive (Rx) settings per port group and to assign ports to port groups, see "Configuring Tx/Rx for Ethernet Port-Pair Groups" on page 88.

The procedure below describes how to configure the Ethernet ports in the Web interface. You can also configure these ports using the table ini file parameter, PhysicalPortsTable or CLI command, configure voip/physical-port.

➢ To configure the physical Ethernet ports:

1. Open the Physical Ports Settings page (**Configuration** tab > **VoIP** menu > **Network** > **Physical Ports Table**).
2. Select a port that you want to configure by clicking its table row, and then click **Edit**; the following dialog box appears:



Edit Record #0

Index	0
Port	GE_1
Mode	Enable
Native Vlan	1
Speed	Auto Negotiation
Description	User Port #0
Group Member	GROUP_1
Group Status	Active

3. Configure the port according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 18: Physical Port Settings Parameter Descriptions

Parameter	Description

Parameter	Description
Port CLI: port **[PhysicalPortsTable_Port]**	(Read-only) Displays the port number.
Mode CLI: mode **[PhysicalPortsTable_Mode]**	(Read-only) Displays the mode of the port: ▪ **[0]** Disable ▪ **[1]** Enable (default)
Native Vlan CLI: native-vlan **[PhysicalPortsTable_NativeVlan]**	Defines the Native VLAN or PVID of the port. Incoming packets without a VLAN ID are tagged with this VLAN. For outgoing packets, if the VLAN ID as defined in the Interface table is the same as the Native VLAN ID, the device sends the packet without a VLAN; otherwise, the VLAN ID as defined in the Interface table takes precedence. The valid value range is 1 to 4096. The default is 1.
Speed & Duplex CLI: speed-duplex **[PhysicalPortsTable_SpeedDuplex]**	Defines the speed and duplex mode of the port. ▪ **[0]** 10BaseT Half Duplex ▪ **[1]** 10BaseT Full Duplex ▪ **[2]** 100BaseT Half Duplex ▪ **[3]** 100BaseT Full Duplex ▪ **[4]** Auto Negotiation (default) ▪ **[6]** 1000BaseT Half Duplex ▪ **[7]** 1000BaseT Full Duplex
Description CLI: port-description **[PhysicalPortsTable_PortDescription]**	Defines an arbitrary description of the port.
Group Member CLI: group-member **[PhysicalPortsTable_GroupMember]**	(Read-only) Displays the group to which the port belongs.
Group Status CLI: group-status **[PhysicalPortsTable_GroupStatus]**	(Read-only) Displays the status of the port: ▪ "Active" - the active port ▪ "Redundant" - the standby (redundant) port

13.2 Configuring Tx/Rx for Ethernet Port-Pair Groups

The Ethernet Group Settings table lets you configure the transmit (Tx) and receive (Rx) settings for the physical Ethernet ports belonging to a port-pair group for 1+1 physical port redundancy. You can also assign ports to each port group, where the group can be assigned a single port or two ports for 1+1 redundancy. If an Ethernet Group has a single port, it will operate as a single port (i.e., without 1+1 redundancy). You can setup the device with a combination of Ethernet Groups, where one group has only one physical port (i.e., no redundancy) and another group has two ports for port-pair redundancy.

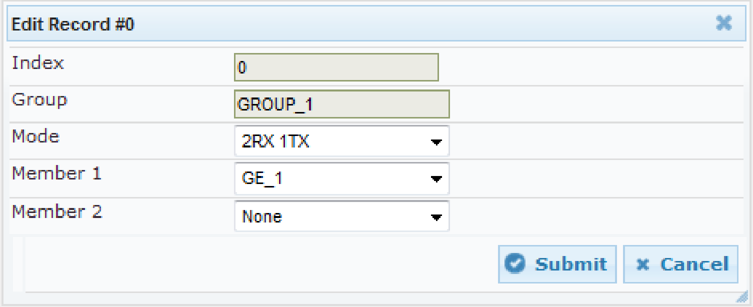
To view the mapping of physical ports to logical ports (strings) used in the device's management tools (e.g., Web interface), use the CLI command, show voip ports. This displays the MAC address and port status (up or down) of the physical port, and its corresponding logical port. Below shows an example of the mapping results from running this command:

```
# show voip ports
Port Num   Port Name      MAC Address      Speed       Duplexity  Link Status Native
VLAN
----------         -----------           -----------------        --------        -
----------     ------------    -----------------
 1                GE_1         00:1e:67:11:7c:28    100Mbps     FULL          UP
1
 2                GE_2         00:1e:67:11:7c:29    100Mbps     FULL          DOWN
1
```

The procedure below describes how to configure Tx/Rx mode in the Web interface. You can also configure this using the table ini file parameter, EtherGroupTable or CLI command, configure voip/ether-group.

➢ To configure the Tx/Rx mode of and assign ports to the Ethernet port-pair:

1. Open the Ethernet Group Settings page (**Configuration** tab > **VoIP** menu > **Network** > **Ethernet Groups Table**).
2. Select an Ethernet Group that you want to configure by clicking its table row, and then click **Edit**; the following dialog box appears:



Edit Record #0

Index	0
Group	GROUP_1
Mode	2RX 1TX
Member 1	GE_1
Member 2	None

3. Configure the Ethernet Group according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 19: Ethernet Group Settings Parameter Descriptions

Parameter	Description
Group CLI: group **[EtherGroupTable_Group]**	(Read-only) Displays the Ethernet port-pair group number.
Mode CLI: mode **[EtherGroupTable_Mode]**	Defines the mode of operation of the ports in the group: ▪ [2] 1RX/1TX = (Default) At any given time, only a single port in the group can transmit and receive packets. If a link exists on both ports, then the active one is either the first to have a link up or the lower-numbered port if both have the same link up from start. ▪ **[3]** 2RX/1TX = Both ports in the group can receive packets, but only one port can transmit. The transmitting port is determined arbitrarily by the device. If the selected port fails at a later stage, a switchover to the redundant port is done, which begins to transmit as well as receive. ▪ **[4]** 2RX/2TX = Both ports in the group can receive and transmit packets. **Notes:** ▪ It is recommended to use the 2RX/1TX option when implementing 1+1 Ethernet port redundancy. In such a setup, the ports can be connected to the same LAN switch or each to a different switch where both are in the same subnet. If connecting each port to a different switch, the 2RX/2TX option can be used but

Parameter	Description
	only if the port group is associated with OAMP and/or Control application types, not media. ▪ For Ethernet port settings and connections of the Maintenance interface when implementing High Availability, see Initial HA Configuration on page 279.
Member 1 CLI: member1 **[EtherGroupTable_Member1]**	Defines the first port in the Ethernet Group.
Member 2 CLI: member2 **[EtherGroupTable_Member2]**	Defines the second port in the Ethernet Group.

13.3 Configuring Underlying Ethernet Devices

The Ethernet Device table lets you configure *Ethernet Devices* (underlying devices). An Ethernet Device represents a Layer-2 bridging device and is assigned with a VLAN ID. An Ethernet Device is associated with an IP network interface in the Interface table ('Underlying Device' field) and/or with a static route in the Static Route table ('Device Name' field). Multiple IP interfaces can be associated with the same Ethernet Device and thereby, implement mutihoming (multiple addresses on the same interface/VLAN).

The Ethernet Device table lets you configure Ethernet Devices by defining a VLAN ID assigning it an arbitrary name for future reference to other configuration items, and associating it with an Ethernet Port Group.

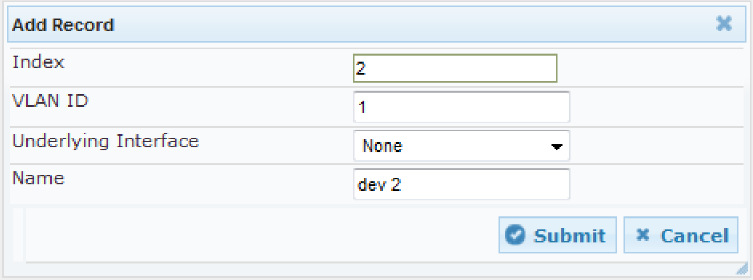
You can view configured Ethernet Devices that have been successfully applied to the device (saved to flash), in the Ethernet Device Status Table page. This page is accessed by clicking the **Ethernet Device Status Table** button, located at the bottom of the Ethernet Device Table page. The Ethernet Device Status Table page can also be accessed from the **Status & Diagnostics** tab > **VoIP Status** menu > **Ethernet Device Status Table** (see "Viewing Ethernet Device Status" on page 339).

Note: You cannot delete an Ethernet Device that is associated with an IP network interface (in the Interface table). Only after the Ethernet Device has been disassociated from the IP network interface can it be deleted.

The procedure below describes how to configure Ethernet devices in the Web interface. You can also configure this using the table ini file parameter, DeviceTable or CLI command, config-voip > interface network-dev.

➢ To configure an Ethernet Device:

1. Open the Ethernet Device Table page (**Configuration** tab > **VoIP** menu > **Network** > **Ethernet Device Table**).
2. Click **Add**; the following dialog box appears:



Add Record

Index	2
VLAN ID	1
Underlying Interface	None
Name	dev 2

3. Configure an Ethernet Device according to the parameters described in the table below.
4. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

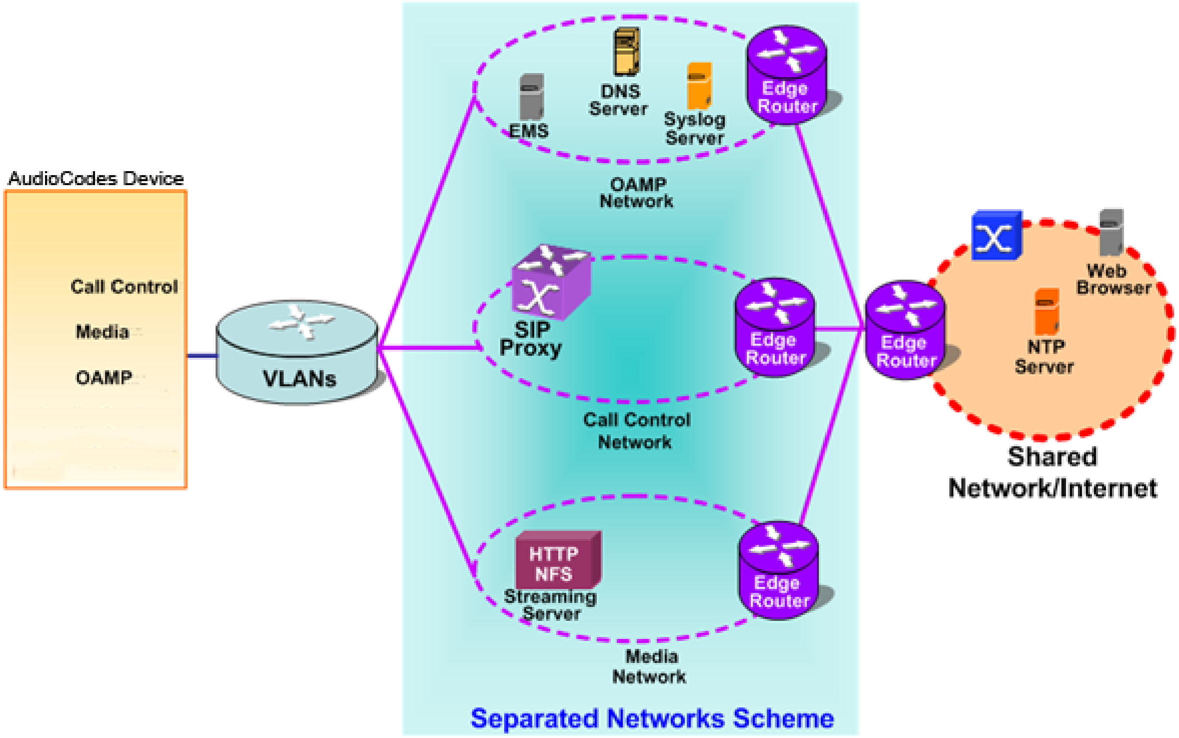
Table 20: Ethernet Device Table Parameter Descriptions

Parameter	Description
Index **[DeviceTable_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
VLAN ID CLI: vlan-id **[DeviceTable_VlanID]**	Defines a VLAN ID. The valid value is 1 to 3999. The default value is 1. **Note:** Each Ethernet Port Group must be configured with a unique VLAN ID.
Underlying Interface CLI: underlying-if [DeviceTable_UnderlyingInterface]	Assigns an Ethernet Port Group to the VLAN (mandatory field). For configuring Ethernet Port Groups, see Configuring Tx/Rx for Ethernet Port-Pair Groups on page 88.
Name CLI: name **[DeviceTable_DeviceName]**	Defines a name for the VLAN. This name is used to associate the VLAN with an IP network interface in the Interface table ('Underlying Device' field - see "Configuring IP Network Interfaces" on page 91) and/or with a static route in the Static Route table ('Device Name' field - see "Configuring Static IP Routing" on page 99). By default, the device automatically assigns a name using the following syntax: "dev <next available table row index>" (e.g., "dev 3").

13.4 Configuring IP Network Interfaces

You can configure a single VoIP network interface for all applications, including OAMP (management traffic), call control (SIP signaling messages), and media (RTP traffic), or you can configure multiple logical, IP network interfaces for these applications. You may need to logically separated network segments for these applications for administration and security. This can be achieved by employing Layer-2 VLANs and Layer-3 subnets. The figure below illustrates a typical network architecture where the device is configured with three network interfaces, each representing the OAMP, call control, and media applications. The device is connected to a VLAN-aware switch for directing traffic from and to the device to the three separated Layer-3 broadcast domains according to VLAN tags (middle pane).

Figure 36: Multiple Network Interfaces



The device is shipped with a default OAMP interface. For more information, see "Default OAMP IP Address" on page 17. The Interface table lets you change this OAMP interface and configure additional network interfaces for control and media, if necessary. You can configure up to 48 interfaces, consisting of up to 47 Control and Media interfaces including a Maintenance interface if your device is deployed in a High Availability (HA) mode, and 1 OAMP interface. Each IP interface is configured with the following:

- Application type allowed on the interface:
  - Control: call control signaling traffic (i.e., SIP)
  - Media: RTP traffic
  - Operations, Administration, Maintenance and Provisioning (OAMP): management (i.e., Web, CLI, and SNMP based management)
  - Maintenance: This interface is used in HA mode when two devices are deployed for redundancy, and represents one of the LAN interfaces or Ethernet groups on each device used for the Ethernet connectivity between the two devices. For more information on HA and the Maintenance interface, see Configuring High Availability on page **Erreur ! Signet non défini.**.
- IP address (IPv4 and IPv6) and subnet mask (prefix length)
- For configuring Quality of Service (QoS), see "Configuring the QoS Settings" on page 102.
- Default Gateway: Traffic from this interface destined to a subnet that does not meet any of the routing rules (local or static) are forwarded to this gateway
- Primary and secondary domain name server (DNS) addresses (optional)
- Underlying Ethernet Device: Layer-2 bridging device and assigned a VLAN ID. As the Ethernet Device is associated with an Ethernet Port Group, this is useful for setting trusted and un-trusted networks on different physical Ethernet ports. Multiple entries in the Interface table may be associated with the same Ethernet Device, providing multi-homing IP configuration (i.e., multiple IP addresses on the same interface/VLAN).

Complementing the Interface table is the Static Route table, which lets you configure static routing rules for non-local hosts/subnets. For more information, see "Configuring Static IP Routing" on page 99.

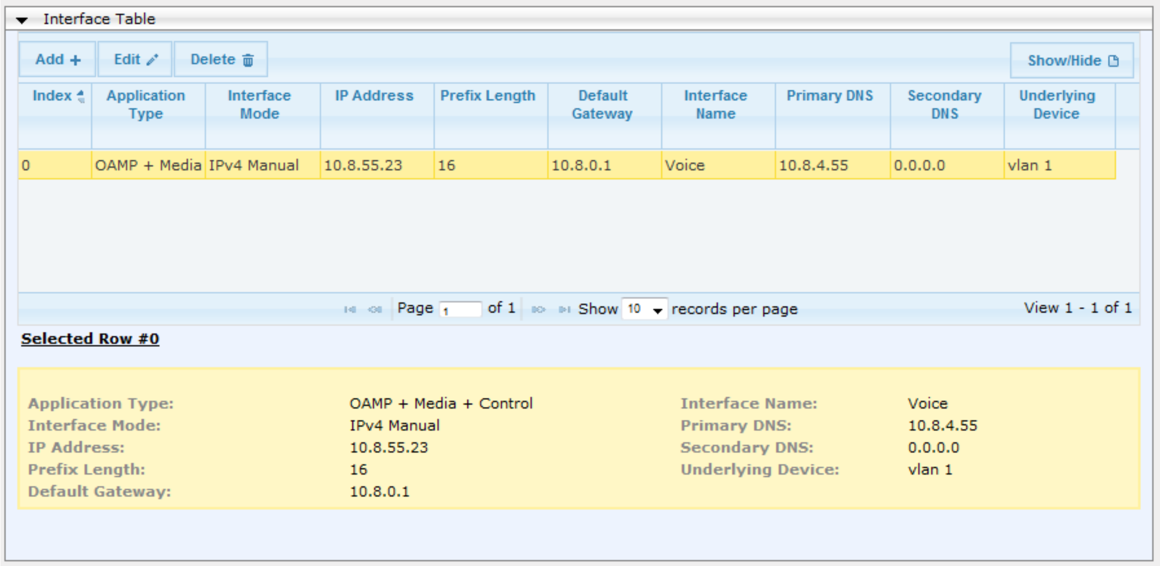
Notes:

- Before configuring IP interfaces, it is recommended that you read the IP interface configuration guidelines in "Interface Table Configuration Guidelines" on page 95.
- The IPv6 feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see Software License Key on page 309.

The procedure below describes how to configure the IP network interfaces in the Web interface. You can also configure IP network interfaces using the table ini file parameter, InterfaceTable or CLI command, configure voip/interface network-if.

➢ To configure IP network interfaces:

1. Open the Interface Table page (**Configuration** tab > **VoIP** menu > **Network** > **IP Interfaces Table**).



Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Interface Name	Primary DNS	Secondary DNS	Underlying Device
0	OAMP + Media	IPv4 Manual	10.8.55.23	16	10.8.0.1	Voice	10.8.4.55	0.0.0.0	vlan 1

Page 1 of 1 Show 10 records per page View 1 - 1 of 1

Selected Row #0

Application Type:	OAMP + Media + Control	Interface Name:	Voice
Interface Mode:	IPv4 Manual	Primary DNS:	10.8.4.55
IP Address:	10.8.55.23	Secondary DNS:	0.0.0.0
Prefix Length:	16	Underlying Device:	vlan 1
Default Gateway:	10.8.0.1		

2. Click **Add**; a dialog box appears.
3. Configure the IP network interface according to the parameters described in the table below.
4. Click **Submit**.

To view configured network interfaces that are currently active, click the **IP Interface Status Table** button. For more information, see "Viewing Active IP Interfaces" on page 339.

Table 21: Interface Table Parameters Description

Parameter	Description
Table parameters	
Index CLI: network-if **[InterfaceTable_Index]**	Table index row of the interface. The range is 0 to 47.
Web: Application Type CLI: application-type **[InterfaceTable_ApplicationTypes]**	Defines the applications allowed on the interface. ▪ **[0]** OAMP = Operations, Administration, Maintenance and Provisioning (OAMP) applications (e.g., Web, Telnet, SSH, and SNMP). ▪ **[1]** Media = Media (i.e., RTP streams of voice). ▪ **[2]** Control = Call Control applications (e.g., SIP). ▪ **[3]** OAMP + Media = OAMP and Media applications. ▪ **[4]** OAMP + Control = OAMP and Call Control applications. ▪ **[5]** Media + Control = Media and Call Control applications. ▪ **[6]** OAMP + Media + Control = All application types are allowed on the interface. ▪ [99] MAINTENANCE = Only the Maintenance application for HA is allowed on this interface.
Web: Interface Mode [InterfaceTable_InterfaceMode]	Defines the method that the interface uses to acquire its IP address. ▪ [3] IPv6 Manual Prefix = IPv6 manual prefix IP address assignment. The IPv6 prefix (higher 64 bits) is set manually while the interface ID (the lower 64 bits) is derived from the device's MAC address. ▪ [4] IPv6 Manual = IPv6 manual IP address (128 bits) assignment. ▪ [10] IPv4 Manual = IPv4 manual IP address (32 bits) assignment.
Web: IP Address CLI: ip-address **[InterfaceTable_IPAddress]**	Defines the IPv4/IPv6 address, in dotted-decimal notation.
Web: Prefix Length CLI: prefix-length **[InterfaceTable_PrefixLength]**	Defines the prefix length of the related IP address. This is a Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted-decimal format. For example, 192.168.0.0/16 is synonymous with 192.168.0.0 and subnet 255.255.0.0. This CIDR lists the number of '1' bits in the subnet mask (i.e., replaces the standard dotted-decimal representation of the subnet mask for IPv4 interfaces). For example, a subnet mask of 255.0.0.0 is represented by a prefix length of 8 (i.e., 11111111 00000000 00000000 00000000) and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (i.e., 11111111 11111111 11111111 11111100). The prefix length is a Classless Inter-Domain Routing (CIDR) style presentation of a dotted-decimal subnet notation. The CIDR-style presentation is the latest method for interpretation of IP addresses. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes. The prefix length for IPv4 must be set to a value from 0 to 30. The prefix length for IPv6 must be set to a value from 0 to 64.
Web: Default Gateway CLI: gateway **[InterfaceTable_Gateway]**	Defines the IP address of the default gateway for the interface. When traffic is sent from this interface to an unknown destination (i.e., not in the same subnet and not defined for any static routing rule), it is forwarded to this default gateway.

Parameter	Description
Web: Interface Name CLI: name **[InterfaceTable_InterfaceName]**	Defines a name for the interface. This name is used in various configuration tables to associate the network interface with other configuration entities such as Media Realms. It is also displayed in management interfaces (Web, CLI, and SNMP) for clarity where it has no functional use. The valid value is a string of up to 16 characters.
Web: Primary DNS CLI: primary-dns **[InterfaceTable_PrimaryDNSServerIPAddress]**	(Optional) Defines the primary DNS server's IP address (in dotted-decimal notation), which is used for translating domain names into IP addresses for the interface. By default, no IP address is defined.
Web: Secondary DNS CLI: secondary-dns **[InterfaceTable_SecondaryDNSServerIPAddress]**	(Optional) Defines the secondary DNS server's IP address (in dotted-decimal notation), which is used for translating domain names into IP addresses for the interface. By default, no IP address is defined.
Underlying Interface [InterfaceTable_UnderlyingInterface]	Assigns an Ethernet Device to the IP interface. To configure Ethernet Devices, see Configuring Underlying Ethernet Devices on page 90.

13.4.1 Assigning NTP Services to Application Types

You can associate the Network Time Protocol (NTP) application with the OAMP or Control application type. This is done using the EnableNTPasOAM ini file parameter.

13.4.2 Multiple Interface Table Configuration Summary and Guidelines

The Interface table configuration must adhere to the following rules:

- Multiple Control and Media interfaces can be configured with overlapping IP addresses and subnets.
- The prefix length replaces the dotted-decimal subnet mask presentation and **must** have a value of 0-30 for IPv4 addresses and a value of 0-64 for IPv6 addresses.
- **One** OAMP interface must be configured and this **must** be an IPv4 address. This OAMP interface can be combined with Media and Control.
- At least one Control interface **must** be configured.
- At least one Media interface **must** be configured.
- Multiple Media and/or Control interfaces can be configured with an IPv6 address.
- The network interface types can be combined:
  - Example 1:
    - One combined OAMP-Media-Control interface with an IPv4 address
  - Example 2:
    - One OAMP interface with an IPv4 address
    - One or more Control interfaces with IPv4 addresses
    - One or more Media interfaces with IPv4 interfaces
  - Example 3:
    - One OAMP with an IPv4 address
    - One combined Media-Control interface with IPv4 address
    - One combined Media-Control interface with IPv6 address
- Each network interface can be configured with a Default Gateway. The address of the

Default Gateway **must** be in the same subnet as the associated interface. Additional static routing rules can be configured in the Static Route table.

- The interface name **must** be configured (mandatory) and must be unique for each interface.
- For IPv4 addresses, the 'Interface Mode' column must be set to IPv4 Manual. For IPv6 addresses, this column must be set to IPv6 Manual or IPv6 Manual Prefix.

> **Note:** Upon device start up, the Interface table is parsed and passes comprehensive validation tests. If any errors occur during this validation phase, the device sends an error message to the Syslog server and falls back to a "safe mode", using a single interface without VLANs. Ensure that you view the Syslog messages that the device sends in system startup to see if any errors occurred.

13.4.3 Networking Configuration Examples

This section provides configuration examples of networking interfaces.

13.4.3.1 One VoIP Interface for All Applications

This example describes the configuration of a single VoIP interface for all applications:

1. **Interface table:** Configured with a single interface for OAMP, Media and Control:

Table 22: Example of Single VoIP Interface in Interface Table

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Interface	Interface Name
0	OAMP, Media & Control	IPv4	192.168.0.2	16	192.168.0.1	1	myInterface

2. **Static Route table:** Two routes are configured for directing traffic for subnet 201.201.0.0/16 to 192.168.11.10, and all traffic for subnet 202.202.0.0/16 to 192.168.11.1:

Table 23: Example of Static Route Table

Destination	Prefix Length	Gateway
201.201.0.0	16	192.168.11.10
202.202.0.0	16	192.168.11.1

3. The NTP applications remain with their default application types.

13.4.3.2 VoIP Interface per Application Type

This example describes the configuration of three VoIP interfaces; one for each application type:

1. **Interface table:** Configured with three interfaces, each for a different application type, i.e., one for OAMP, one for Call Control, and one for RTP Media, and each with a different VLAN ID and default gateway:

Table 24: Example of VoIP Interfaces per Application Type in Interface Table

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Interface	Interface Name

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Interface	Interface Name
0	OAMP	IPv4 Manual	192.168.0.2	16	192.168.0.1	1	ManagementIF
1	Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	200	myControlIF
2	Media	IPv4 Manual	211.211.85.14	24	211.211.85.1	211	myMediaIF

2. **Static Route table:** A routing rule is required to allow remote management from a host in 176.85.49.0 / 24:

Table 25: Example Static Route Table

Destination	Prefix Length	Gateway
176.85.49.0	24	192.168.11.1

3. All other parameters are set to their respective default values. The NTP application remains with its default application types.

13.4.3.3 *VoIP Interfaces for Combined Application Types*

This example describes the configuration of multiple interfaces for the following applications:

- One interface for the OAMP application.
- Interfaces for Call Control and Media applications, where two of them are IPv4 interfaces and one is an IPv6 interface.

1. **Interface table:**

Table 26: Example of VoIP Interfaces of Combined Application Types in Interface Table

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Interface	Interface Name
0	OAMP	IPv4 Manual	192.168.0.2	16	192.168.0.1	1	Mgmt
1	Media & Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	201	MediaCntrl1
2	Media & Control	IPv4 Manual	200.200.86.14	24	200.200.86.1	202	MediaCntrl2
3	Media & Control	IPv6 Manual	2000::1:200:200:86:14	64	::	202	V6CntrlMedia2

2. **Static Route table:** A routing rule is required to allow remote management from a host in 176.85.49.0/24:

Table 27: Example of Static Route Table

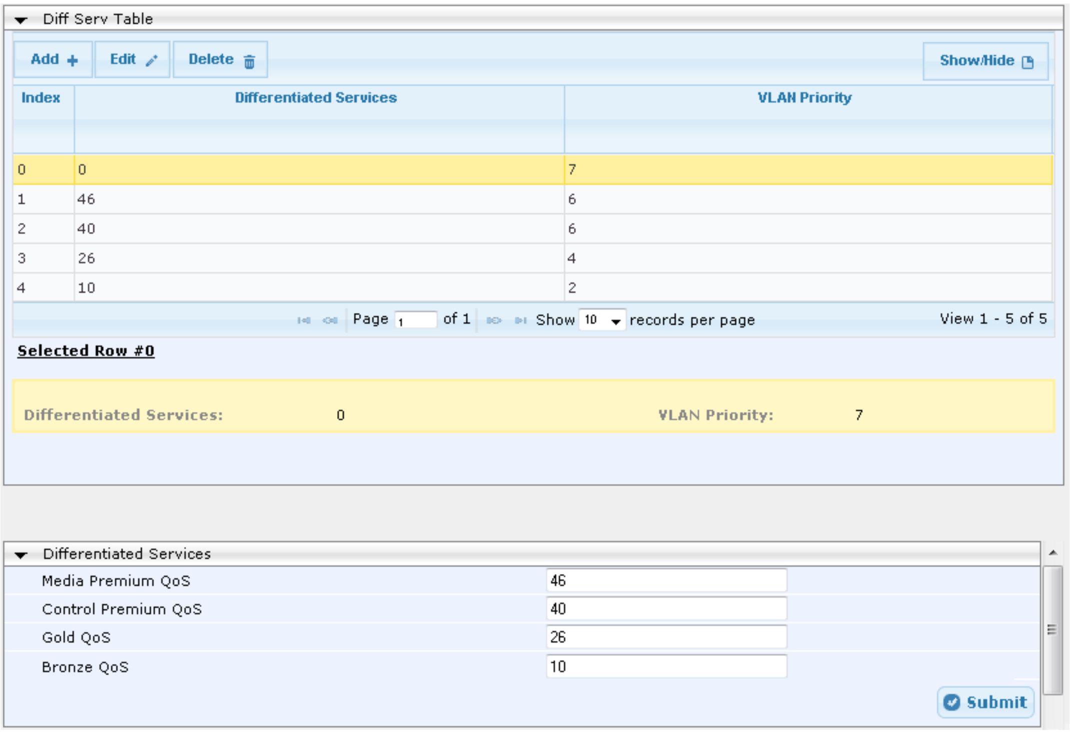
Destination	Prefix Length	Gateway
176.85.49.0	24	192.168.0.10

3. The NTP application is configured (using the ini file) to serve as OAMP applications:

```
EnableNTPasOAM = 1
```

4. DiffServ table:
   - Layer-2 QoS values are assigned:
     - ♦ For packets sent with DiffServ value of 46, set VLAN priority to 6
     - ♦ For packets sent with DiffServ value of 40, set VLAN priority to 6
     - ♦ For packets sent with DiffServ value of 26, set VLAN priority to 4
     - ♦ For packets sent with DiffServ value of 10, set VLAN priority to 2
   - Layer-3 QoS values are assigned:
     - ♦ For Media Service class, the default DiffServ value is set to 46
     - ♦ For Control Service class, the default DiffServ value is set to 40
     - ♦ For Gold Service class, the default DiffServ value is set to 26
     - ♦ For Bronze Service class, the default DiffServ value is set to 10

Figure 37: Example of Layer-2 QoS in DiffServ Table



Index	Differentiated Services	VLAN Priority
0	0	7
1	46	6
2	40	6
3	26	4
4	10	2

Selected Row #0

Differentiated Services: 0 VLAN Priority: 7

Differentiated Services

Media Premium QoS	46
Control Premium QoS	40
Gold QoS	26
Bronze QoS	10

13.4.3.4 VoIP Interfaces with Multiple Default Gateways

Below is a configuration example using default gateways per IP network interface. In this example, the default gateway for OAMP is 192.168.0.1 and for Media and Control it is 200.200.85.1.

Table 28: Configured Default Gateway Example

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Interface	Interface Name
0	OAMP	IPv4 Manual	192.168.0.2	16	192.168.0.1	100	Mgmt
1	Media & Control	IPv4 Manual	200.200.85.14	24	200.200.85.1	200	CntrlMedia

A separate Static Route table lets you configure static routing rules. Configuring the following static routing rules enables OAMP applications to access peers on subnet 17.17.0.0 through the gateway 192.168.10.1 (which is not the default gateway of the interface), and Media & Control applications to access peers on subnet 171.79.39.0 through the gateway 200.200.85.10 (which is not the default gateway of the interface).

Table 29: Separate Static Route Table Example

Destination	Prefix Length	Gateway	Interface Name
17.17.0.0	16	192.168.10.1	0
171.79.39.0	24	200.200.85.10	1

13.5 Configuring Static IP Routes

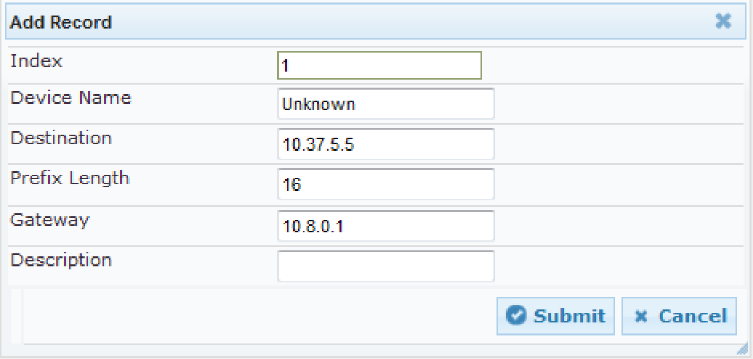
The Static Route Table page lets you configure up to 30 static IP routing rules for the device. These rules can be associated with a network interface (configured in the Interface table), through its association with an Ethernet Device. Thus, the routing decision is based on the source subnet/VLAN. If not associated with an IP interface, the static route rule is based on destination IP address. Traffic destined to the subnet specified in the routing rule is re-directed to the defined gateway, reachable through the specified interface. Before sending an IP packet, the device searches this table for an entry that matches the requested destination host/network. If such an entry is found, the device sends the packet to the indicated router. If no explicit entry is found, the packet is sent to the default gateway.

The status of the configured static routes can be viewed in the IP Routing Status Table page. This page can be accessed by clicking the **Static Route Status Table** button located at the bottom of the Static Route Table page, or it can be accessed from the Navigation tree under the **Status & Diagnostics** tab (see "Viewing Static Routes Status" on page 339).

The procedure below describes how to configure static routes in the Web interface. You can also configure this using the table ini file parameter, StaticRouteTable or the CLI command, configure voip/routing static.

➢ To configure a static IP route:

1. Open the Static Route Table page (**Configuration** tab > **VoIP** menu > **Network** > **Static Route Table**).
2. Click **Add**; the following dialog box appears:



Add Record

Index	1
Device Name	Unknown
Destination	10.37.5.5
Prefix Length	16
Gateway	10.8.0.1
Description	

3. Configure a static route according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Note: You can delete only static routing rules that are inactive.

Table 30: Static Route Table Parameter Descriptions

Parameter	Description

Parameter	Description
Index **[StaticRouteTable_Index]**	Defines an index number for the new table record. The valid value is 0 to 29. **Note:** Each table row must be configured with a unique index.
Device Name CLI: device-name [StaticRouteTable_DeviceName]	Associates the static route with an Ethernet Device (e.g., VLAN), configured in the Ethernet Device table. The static route is used for traffic received on this Ethernet Device. This value must be identical to the string value configured for the Ethernet Device in the 'Name' field of the Ethernet Device table. For configuring Ethernet Devices, see Configuring Underlying Ethernet Devices on page 90.
Destination CLI: destination **[StaticRouteTable_Destination]**	Defines the IP address of the destination host/network. The destination can be a single host or a whole subnet, depending on the prefix length configured for this routing rule.
Prefix Length CLI: prefix-length **[StaticRouteTable_PrefixLength]**	Defines the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation, of the destination host/network. The CIDR-style representation uses a suffix indicating the number of bits that are set in the dotted-decimal format. For example, 16 is synonymous with subnet 255.255.0.0.
The address of the host/network you want to reach is determined by an AND operation that is applied to the fields 'Destination' and 'Prefix Length'. For example, to reach the network 10.8.x.x, enter 10.8.0.0 in the 'Destination' field and 16 in the 'Prefix Length'. As a result of the AND operation, the value of the last two octets in the 'Destination' field is ignored. To reach a specific host, enter its IP address in the 'Destination' field and 32 in the 'Prefix Length' field.	
Gateway CLI: gateway **[StaticRouteTable_Gateway]**	Defines the IP address of the router (next hop) used for traffic destined to the subnet/host defined in the 'Destination' / 'Prefix Length' field. **Note:** The Gateway's address must be in the same subnet as the IP address of the network interface (Ethernet Device) on which you configure this static route.
Description CLI: description **[StaticRouteTable_Description]**	Defines an arbitrary name to easily identify the static route rule. The valid value is a string of up to 20 characters.

13.5.1 Configuration Example of Static IP Routes

The example below describes how to configure static routes.

Note: The Interface Address family must be coherent with the Routing Rule Address family. IPv4 interfaces cannot be selected in an IPv6 routing rule, and vice versa.

1. Configure network interfaces in the Interface table:

Table 31: Configured Network Interfaces in Interface Table

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Interface	Interface Name
0	OAMP	IPv4 Manual	192.168.0.2	16	192.168.0.1	501	Mng
1	Media &	IPv4 Manual	10.32.174.50	24	10.32.174.1	2012	MediaCntrl

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Interface	Interface Name
	Control						
2	Media	IPv4 Manual	10.33.174.50	24	10.33.174.1	2013	Media1
3	Control	IPv4 Manual	10.34.174.50	24	10.34.174.1	2014	Cntrl1

2. Configure static routes in the Static Route table:

Table 32: Configured Static Routes in Static Route Table

Destination	Prefix Length	Gateway	Device Name
10.31.174.0	24	192.168.11.1	Mng
174.96.151.15	24	10.32.174.12	MediaCntrl
10.35.174.0	24	10.34.174.240	Cntrl1

Note that the IP address configured in the 'Gateway' field (i.e., next hop) must reside on the same subnet as the IP address of the associated network interface that is specified in the 'Interface Name' field.

13.5.2 Static Route Table Configuration Guidelines

The Static Route table configurations must adhere to the following rules:

- Up to 30 different static routing rules can be configured.
- The 'Prefix Length' replaces the dotted-decimal subnet mask presentation. This column must have a value of 0-31 for IPv4 interfaces and a value of 0-64 for IPv6 interfaces.
- The 'Gateway' field must be on the same subnet as the IP address of the associated interface specified in the 'Interface Name' field.
- The 'Interface Name' selected for the routing rule must be of the same address family as the rule defined.
- For the configuration settings to take effect, you must reset the device with a "burn" to flash memory.

13.5.3 Troubleshooting the Routing Table

When adding a new static route to the Static Route table, the added rule passes a validation test. If errors are found, the static route is rejected and not added to the table. Failed static route validations may result in limited connectivity (or no connectivity) to the destinations specified in the incorrect static route. For any error found in the Static Route table or failure to configure a static route, the device sends a notification message to the Syslog server reporting the problem.

Common static routing configuration errors may include the following:

- The IP address specified in the 'Gateway' field is unreachable from the interface specified in the 'Interface Name' field.
- The same destination is configured in two different static routes.
- More than 30 static routes have been configured.

Note: If a static route is required to access OAMP applications (for remote management, for example) and the route is not configured correctly, the route is not added and the device is not accessible remotely. To restore connectivity, the device must be accessed locally from the OAMP subnet and the required routes be configured.

13.6 Configuring Quality of Service

The QoS Settings page lets you configure Layer-2 and Layer-3 Quality of Service (QoS). Differentiated Services (DiffServ) is an architecture providing different types or levels of service for IP traffic. DiffServ (according to RFC 2474), prioritizes certain traffic types based on priority, accomplishing a higher-level QoS at the expense of other traffic types. By prioritizing packets, DiffServ routers can minimize transmission delays for time-sensitive packets such as VoIP packets.

You can assign DiffServ to the following class of services (CoS) and assign VLAN priorities (IEEE 802.1p) to various values of DiffServ:

- Media Premium – RTP packets sent to the LAN
- Control Premium – control protocol (SIP) packets sent to the LAN
- Gold – HTTP streaming packets sent to the LAN
- Bronze – OAMP packets sent to the LAN

The Layer-3 QoS parameters define the values of the DiffServ field in the IP header of the frames related to a specific service class. The Layer-2 QoS parameters define the values for the 3 priority bits in the VLAN tag according to the value of the DiffServ field in the packet IP header (according to the IEEE 802.1p standard). The DiffServ table lets you configure up to 64 DiffServ-to-VLAN Priority mapping (Layer 2 class of service). For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet.

The mapping of an application to its CoS and traffic type is shown in the table below:

Table 33: Traffic/Network Types and Priority

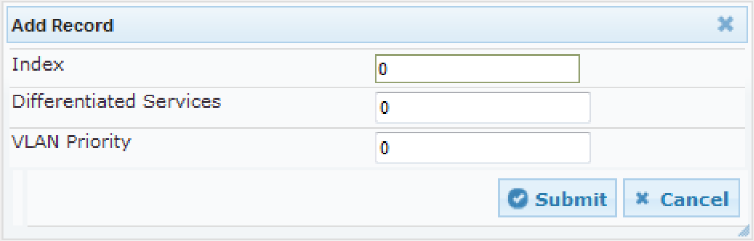
Application	Traffic / Network Types	Class-of-Service (Priority)
Debugging interface	Management	Bronze
Telnet	Management	Bronze
DHCP	Management	Network
Web server (HTTP)	Management	Bronze
SNMP GET/SET	Management	Bronze
Web server (HTTPS)	Management	Bronze
RTP traffic	Media	Premium media
RTCP traffic	Media	Premium media
T.38 traffic	Media	Premium media
SIP	Control	Premium control
SIP over TLS (SIPS)	Control	Premium control
Syslog	Management	Bronze
SNMP Traps	Management	Bronze
DNS client	Varies according to DNS settings: ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium Control ▪ Management: Bronze
NTP	Varies according to the interface type associated with NTP (see "Assigning NTP Services to Application Types" on page 95): ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium control ▪ Management: Bronze

The procedure below describes how to configure DiffServ-to-VLAN priority mapping in the Web interface. You can also configure this using the table ini file parameter, DiffServToVlanPriority or CLI command configure voip > qos vlan-mapping.

➢ To configure QoS:

1. Open the Diff Serv Table page (**Configuration** tab > **VoIP** menu > **Network** > **QoS Settings**).
2. Configure DiffServ-to-VLAN priority mapping (Layer-2 QoS):
   a. Click Add; the following dialog box appears:

Figure 38: DiffServ Table Page - Add Record



Add Record

Index	0
Differentiated Services	0
VLAN Priority	0

   b. Configure a DiffServ-to-VLAN priority mapping (Layer-2 QoS) according to the parameters described in the table below.

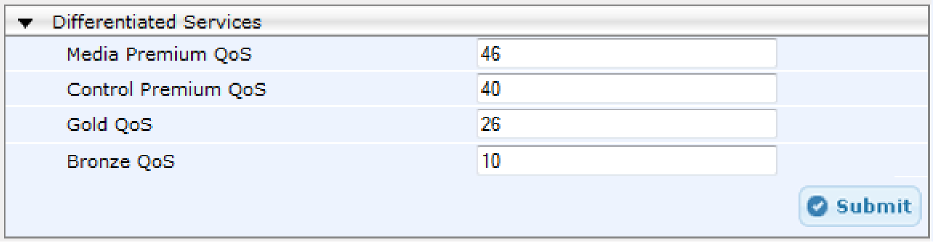
c. Click Submit, and then save ("burn") your settings to flash memory.

Table 34: DiffServ Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Differentiated Services CLI: diff-serv [DiffServToVlanPriority_DiffServ]	Defines a DiffServ value. The valid value is 0 to 63.
VLAN Priority CLI: vlan-priority [DiffServToVlanPriority_VlanPriority]	Defines the VLAN priority level. The valid value is 0 to 7.

3. Under the Differentiated Services group, configure DiffServ (Layer-3 QoS) values per CoS.

Figure 39: QoS Settings Page - Differentiated Services



Differentiated Services

Media Premium QoS: 46

Control Premium QoS: 40

Gold QoS: 26

Bronze QoS: 10

13.7 Configuring ICMP Messages

Internet Control Message Protocol (ICMP) is one of the core protocols of the Internet Protocol suite. It is used by network devices such as routers to send error messages indicating, for example, that a requested service is unavailable.

You can configure the device to handle ICMP messages as follows:

- Send and receive ICMP Redirect messages.
- Send ICMP Destination Unreachable messages. The device sends this message in response to a packet that cannot be delivered to its destination for reasons other than congestion. The device sends a Destination Unreachable message upon any of the following:
  - Address unreachable
  - Port unreachable

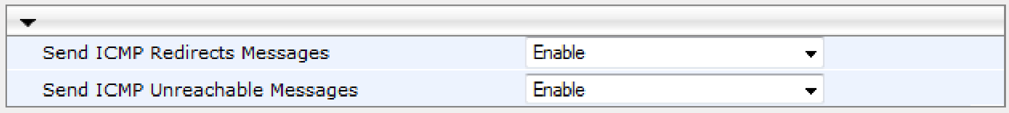
  This feature is applicable to IPv4 and IPv6 addressing schemes.

The procedure below describes how to configure ICMP messaging in the Web interface. You can also configure ICMP messaging using the ini file parameters DisableICMPUnreachable (ICMP Unreachable messages) and DisableICMPRedirects (ICMP Redirect messages).

➢ To configure handling of ICMP messages:

1. Open the Network Settings page (**Configuration** tab > **VoIP** menu > **Network** > **Network Settings**).

Figure 40: Configuring ICMP Messaging in Network Settings Page



Send ICMP Redirects Messages: Enable

Send ICMP Unreachable Messages: Enable

2. To enable or disable sending and receipt of ICMP Redirect messages, use the 'Send ICMP Redirects Messages' parameter.
3. To enable or disable the sending of ICMP Destination Unreachable messages, use the 'Send ICMP Unreachable Messages' parameter.
4. Click **Submit**.

13.8 DNS

You can use the device's embedded domain name server (DNS) or an external, third-party DNS to translate domain names into IP addresses. This is useful if domain names are used as the destination in call routing. The device supports the configuration of the following DNS types:

- Internal DNS table - see "Configuring the Internal DNS Table" on page 105
- Internal SRV table - see "Configuring the Internal SRV Table" on page 106

13.8.1 Configuring the Internal DNS Table

The Internal DNS table, similar to a DNS resolution, translates up to 20 host (domain) names into IP addresses. This functionality can be used when a domain name (FQDN) is configured as an IP destination in a routing rule. Up to four different IP addresses can be assigned to the same host name.

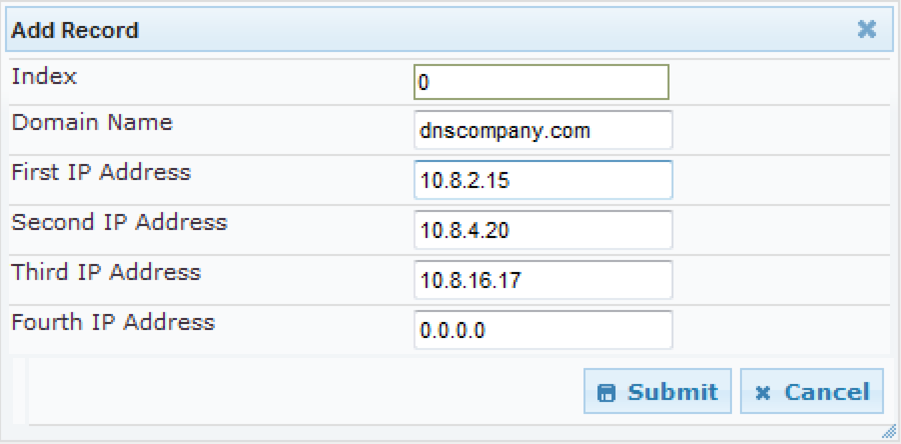
Note: The device initially attempts to resolve a domain name using the Internal DNS table. If the domain name is not configured in the table, the device performs a DNS resolution using an external DNS server for the related IP network interface (see "Configuring IP Network Interfaces" on page 91).

The procedure below describes how to configure the DNS table in the Web interface. You can also this using the table ini file parameter, DNS2IP or CLI command, configure voip > voip-network dns dns-to-ip.

➢ To configure the internal DNS table:

1. Open the Internal DNS Table page (**Configuration** tab > **VoIP** menu > **Network** > **DNS** > **Internal DNS Table**).
2. Click **Add**; the following dialog box appears:

Figure 41: Internal DNS Table - Add Record Dialog Box



Index	0
Domain Name	dnscompany.com
First IP Address	10.8.2.15
Second IP Address	10.8.4.20
Third IP Address	10.8.16.17
Fourth IP Address	0.0.0.0

3. Configure the DNS rule, as required. For a description of the parameters, see the table below.
4. Click **Submit**; the DNS rule is added to the table.

Table 35: Internal DNS Table Parameter Description

Parameter	Description
Domain Name CLI: domain-name **[Dns2Ip_DomainName]**	Defines the host name to be translated. The valid value is a string of up to 31 characters.
First IP Address CLI: first-ip-address **[Dns2Ip_FirstIpAddress]**	Defines the first IP address (in dotted-decimal format notation) to which the host name is translated. The IP address can be configured as an IPv4 and/or IPv6 address.
Second IP Address CLI: second-ip-address **[Dns2Ip_SecondIpAddress]**	Defines the second IP address (in dotted-decimal format notation) to which the host name is translated.
Third IP Address CLI: third-ip-address **[Dns2Ip_ThirdIpAddress]**	Defines the third IP address (in dotted-decimal format notation) to which the host name is translated.
Fourth IP Address CLI: fourth-ip-address **[Dns2Ip_FourthIpAddress]**	Defines the fourth IP address (in dotted-decimal format notation) to which the host name is translated.

13.8.2 Configuring the Internal SRV Table

The Internal SRV table resolves host names to DNS A-Records. Three different A-Records can be assigned to each host name, where each A-Record contains the host name, priority, weight, and port.

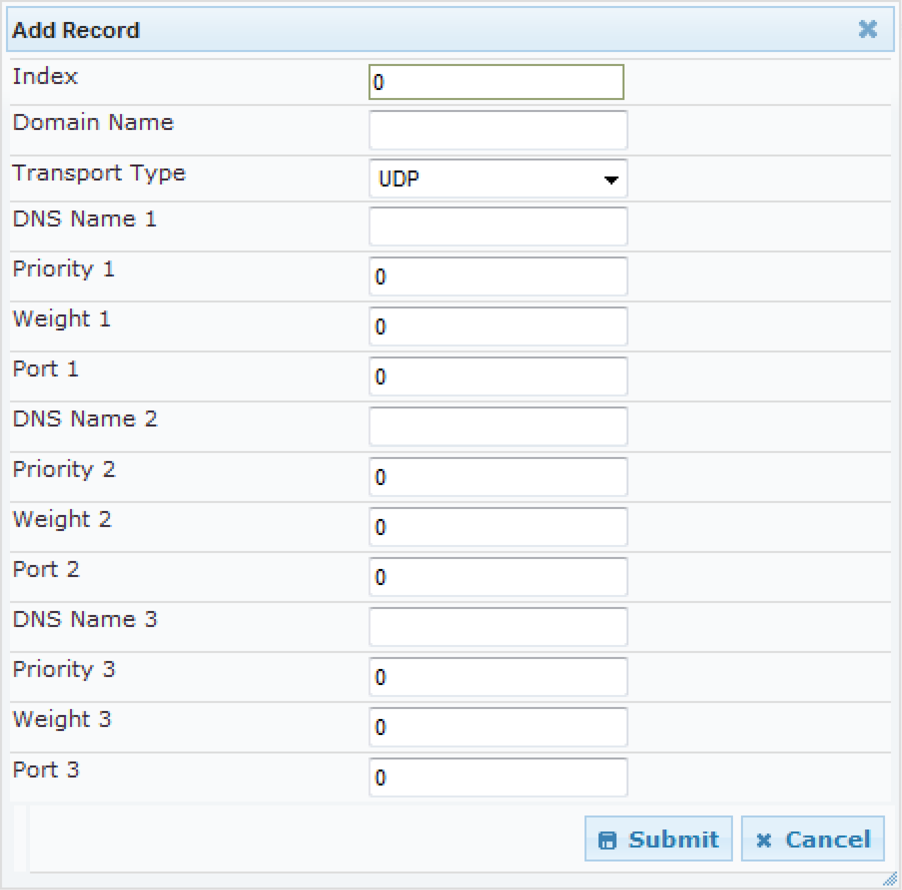
Note: If you configure the Internal SRV table, the device initially attempts to resolve a domain name using this table. If the domain is not configured in the table, the device performs a Service Record (SRV) resolution using an external DNS server, configured in the Interface table (see "Configuring IP Network Interfaces" on page 91).

The procedure below describes how to configure the Internal SRV table in the Web interface. You can also configure this using the table ini file parameter, SRV2IP or CLI command, configure voip > voip-network dns srv2ip.

➢ To configure an SRV rule:

1. Open the Internal SRV Table page (**Configuration** tab > **VoIP** menu > **Network** > **DNS** > **Internal SRV Table**).
2. Click **Add**; the following dialog box appears:

Figure 42: Internal SRV Table Page



Add Record

Index	0
Domain Name	
Transport Type	UDP
DNS Name 1	
Priority 1	0
Weight 1	0
Port 1	0
DNS Name 2	
Priority 2	0
Weight 2	0
Port 2	0
DNS Name 3	
Priority 3	0
Weight 3	0
Port 3	0

3. Configure an SRV rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 36: Internal SRV Table Parameter Descriptions

Parameter	Description
Domain Name CLI: domain-name **[Srv2Ip_InternalDomain]**	Defines the host name to be translated. The valid value is a string of up to 31 characters.
Transport Type CLI: transport-type **[Srv2Ip_TransportType]**	Defines the transport type. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS
DNS Name (1-3) CLI: dns-name-1|2|3 **[Srv2Ip_Dns1/2/3]**	Defines the first, second or third DNS A-Record to which the host name is translated.
Priority (1-3) CLI: priority-1|2|3 **[Srv2Ip_Priority1/2/3]**	Defines the priority of the target host. A lower value means that it is more preferred.

Parameter	Description
Weight (1-3) CLI: weight-1|2|3 **[Srv2Ip_Weight1/2/3]**	Defines a relative weight for records with the same priority.
Port (1-3) CLI: port-1|2|3 **[Srv2Ip_Port1/2/3]**	Defines the TCP or UDP port on which the service is to be found.

13.9 Configuring NFS Settings

Network File System (NFS) enables the device to access a remote server's shared files and directories and to handle them as if they're located locally. The device can use NFS to load *cmp*, *ini*, and auxiliary files through the Automatic Update mechanism (see Configuration using FTP or NFS).

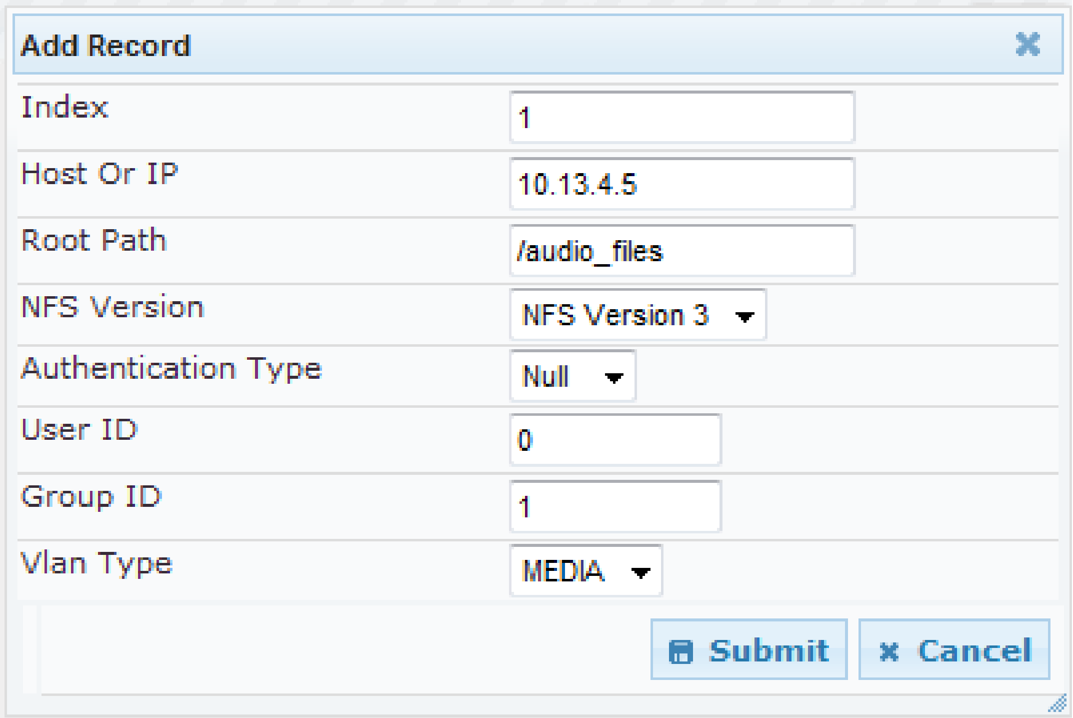
You can configure up to 16 different NFS file systems. As a file system, the NFS is independent of machine types, operating systems and network architectures. Note that an NFS file server can share multiple file systems. There must be a separate row for each remote file system shared by the NFS file server that needs to be accessed by the device.

The procedure below describes how to configure NFS in the Web interface. You can also configure this using the table ini file parameter, NFSServers or CLI command, configure system > nfs > servers.

➢ To configure an NFS file systems:

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).
2. Under the NFS Settings group, click the **NFS Table** ➡ button; the NFS Table page appears.
3. Click **Add**; the following dialog box appears:

Figure 43: NFS Table Page - Add Record



Add Record

Index	1
Host Or IP	10.13.4.5
Root Path	/audio_files
NFS Version	NFS Version 3
Authentication Type	Null
User ID	0
Group ID	1
Vlan Type	MEDIA

4. Configure the NFS parameters according to the table below.
5. Configure an NFS according to the parameters described in the table below.
6. Click **Submit**, and then save ("burn") your settings to flash memory. The remote NFS file system is immediately applied, which can be verified by the appearance of the "NFS mount was successful" message in the Syslog server.

⚠

Notes:

- To avoid terminating current calls, do not delete or edit a row while the device is currently accessing files on that remote NFS file system.
- The combination of 'Host Or IP' and 'Root Path' must be unique for each row in the table. For example, the table must include only one row with a Host/IP of 192.168.1.1 and Root Path of /audio.

Table 37: NFS Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Host Or IP CLI: host **[NFSServers_HostOrIP]**	The domain name or IP address of the NFS server. If a domain name is provided, a DNS server must be configured.
Root Path CLI: root-path **[NFSServers_RootPath]**	Path to the root of the remote file system in the format: /[path]. For example, '/audio'.
NFS Version CLI: version **[NFSServers_NfsVersion]**	NFS version used to access the remote file system. ▪ **[2]** NFS Version 2 ▪ **[3]** NFS Version 3 (default)
Authentication Type CLI: authentication-type **[NFSServers_AuthType]**	Authentication method used for accessing the remote file system. ▪ **[0]** Null ▪ **[1]** Unix (default)
User ID CLI: uid **[NFSServers_UID]**	User ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 0.
Group ID CLI: gid **[NFSServers_GID]**	Group ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 1.
VLAN Type CLI: vlan-type **[NFSServers_VlanType]**	The VLAN type for accessing the remote file system. ▪ **[0]** OAM ▪ **[1]** MEDIA (default) **Note:** This parameter applies only if VLANs are enabled or if Multiple IPs is configured (see "Configuring IP Network Interfaces" on page 91).

13.10 Network Address Translation Support

Network Address Translation (NAT) is a mechanism that maps internal IP addresses (and ports) used within a private network to global IP addresses and vice versa, providing transparent routing to end hosts. The primary advantages of NAT include (1) reduction in the number of global IP addresses required in a private network (global IP addresses are only used to connect to the Internet) and (2) better network security by hiding the internal architecture.

The design of SIP creates a problem for VoIP traffic to pass through NAT. SIP uses IP addresses and port numbers in its message body. However, the NAT server is unable to modify the SIP messages and thus, can't change local addresses to global addresses.

This section discusses the device's solutions for overcoming NAT traversal issues.

13.10.1 Device Located behind NAT

Two different streams traverse through NAT - signaling and media. A device located behind a NAT that initiates a signaling path has problems receiving incoming signaling responses as they are blocked by the NAT server. Therefore, the initiating device must inform the receiving device where to send the media. To resolve this NAT problem, the following solutions are provided by the device, listed in priority of the selected method used by the device:

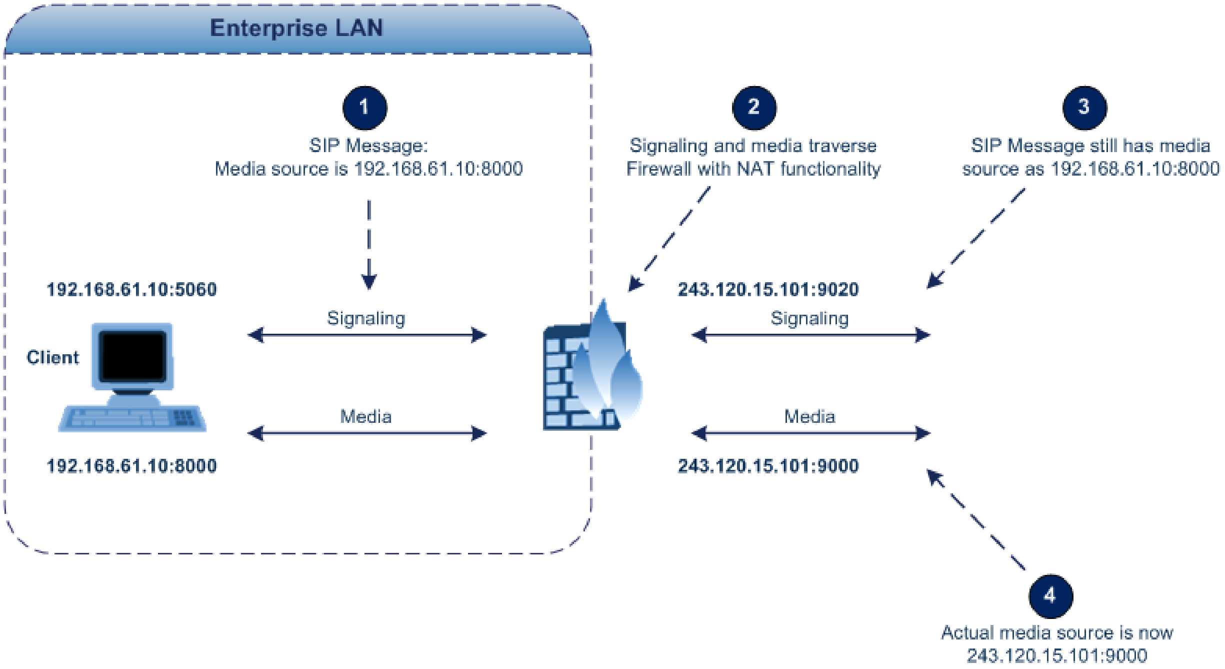
a. If configured, uses the single Static NAT IP address for all interfaces - see "Configuring a Static NAT IP Address for All Interfaces" on page 110.

b. If configured, uses the NAT Translation table which configures NAT per interface - see Configuring NAT Translation per IP Interface on page 111.

If NAT is not configured by any of the above-mentioned methods, the device sends the packet according to its IP address configured in the Interface table.

The figure below illustrates the NAT problem faced by the SIP networks where the device is located behind a NAT:

Figure 44: Device behind NAT and NAT Issues


Enterprise LAN
1
SIP Message:
Media source is 192.168.61.10:8000
2
Signaling and media traverse
Firewall with NAT functionality
3
SIP Message still has media
source as 192.168.61.10:8000
192.168.61.10:5060
Client
Signaling
Media
192.168.61.10:8000
243.120.15.101:9020
Signaling
Media
243.120.15.101:9000
4
Actual media source is now
243.120.15.101:9000

13.10.1.1 Configuring a Static NAT IP Address for All Interfaces

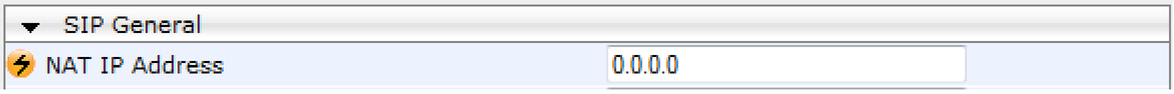
You can configure a global (public) IP address of the router to enable static NAT between the device and the Internet for all network interfaces. Thus, the device replaces the source IP address for media of all outgoing SIP messages sent on any of its network interfaces to this public IP address.

The procedure below describes how to configure a static NAT address in the Web interface. You can also configure this using the ini file parameter, StaticNATIP or CLI command, configure voip > sip-definition general-settings > nat-ip-addr.

➢ To configure a single static NAT IP address:

1. Open the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**).

Figure 45: Configuring Static NAT IP Address in SIP General Parameters Page



SIP General

NAT IP Address: 0.0.0.0

2. In the 'NAT IP Address' field, enter the NAT IP address in dotted-decimal notation.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

13.10.1.2 Configuring NAT Translation per IP Interface

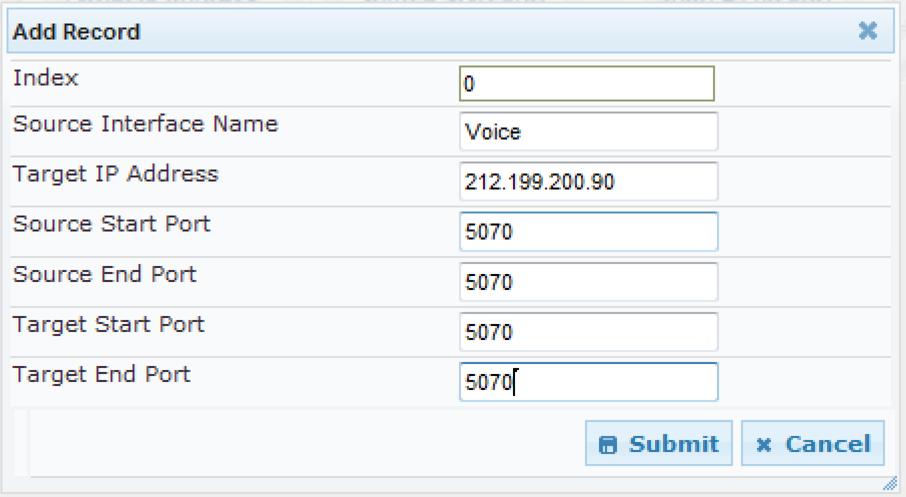
The NAT Translation table lets you configure up to 32 network address translation (NAT) rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses (*global* or *public*), when the device is located behind NAT. This allows, for example, the separation of VoIP traffic between different ITSP's, and topology hiding of internal IP addresses to the "public" network. Each IP interface (configured in the Interface table) can be associated with a NAT rule in this table, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range). The device's NAT traversal mechanism replaces the source IP address of SIP messages sent from a specified VoIP interface to a public IP address.

The procedure below describes how to configure NAT translation rules in the Web interface. You can also configure Bandwidth Profiles using the table ini file parameter, NATTranslation or CLI command, voip-network NATTranslation.

➢ To configure NAT translation rules:

1. Open the NAT Translation Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **NAT Translation Table**).
2. Click **Add**; the following dialog box appears:

Figure 46: NAT Translation Table Page



Add Record

Index	0
Source Interface Name	Voice
Target IP Address	212.199.200.90
Source Start Port	5070
Source End Port	5070
Target Start Port	5070
Target End Port	5070

3. Configure a NAT translation rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 38: NAT Translation Table Parameter Descriptions

Parameter	Description
Index CLI: index **[NATTranslation_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Source Interface Name CLI: SourceIPInterfaceName **[NATTranslation_SourceIPInterfaceName]**	Defines the name of the IP interface, as configured in the Interface table.

Parameter	Description
Target IP Address CLI: TargetIPAddress **[NATTranslation_TargetIPAddress]**	Defines the global IP address. This address is set in the SIP Via and Contact headers as well as in the o= and c= SDP fields.
Source Start Port CLI: SourceStartPort **[NATTranslation_SourceStartPort]**	Defines the optional starting port range (1-65536) of the IP interface, used as matching criteria for this NAT rule. If not configured, the match is done on the entire port range. Only IP addresses and ports of matched source ports will be replaced.
Source End Port CLI: SourceEndPort **[NATTranslation_SourceEndPort]**	Defines the optional ending port range (1-65536) of the IP interface, used as matching criteria for this NAT rule. If not configured, the match is done on the entire port range. Only IP addresses and ports of matched source ports will be replaced.
Target Start Port CLI: TargetStartPort **[NATTranslation_TargetStartPort]**	Defines the optional, starting port range (1-65536) of the global address. If not configured, the ports are not replaced. Matching source ports are replaced with the target ports. This address is set in the SIP Via and Contact headers, as well as in the o= and c= SDP fields.
Target End Port CLI: TargetEndPort **[NATTranslation_TargetEndPort]**	Defines the optional, ending port range (1-65536) of the global address. If not configured, the ports are not replaced. Matching source ports are replaced with the target ports. This address is set in the SIP Via and Contact headers, as well as in the o= and c= SDP fields.

13.10.2 Remote UA behind NAT

13.10.2.1 SIP Signaling Messages

By default, the device resolves NAT issues for SIP signaling, using its NAT Detection mechanism. The NAT Detection mechanism checks whether the endpoint is located behind NAT, by comparing the incoming packet's source IP address with the SIP Contact header's IP address. If the packet's source IP address is a public address and the Contact header's IP address a local address, the device considers the endpoint as located behind NAT. In this case, the device sends the SIP messages to the endpoint, using the packet's source IP address. Otherwise (or if you have disabled the NAT Detection mechanism), the device sends the SIP messages according to the SIP standard RFC 3261, where requests within the SIP dialog are sent using the IP address in the Contact header, and responses to INVITEs are sent using the IP address in the Via header. To enable or disable the device's NAT Detection mechanism, use the 'SIP NAT Detection' parameter.

If necessary, you can also configure the device to always consider incoming SIP INVITE messages as sent from endpoints that are located behind NAT. When this is enabled, the device sends responses to the INVITE (to the endpoint), using the the source IP address of the packet (INVITE) initially received from the endpoint. This is especially useful in scenarios where the endpoint is located behind a NAT firewall and the device (for whatever reason) is unable to identify NAT using its regular NAT Detection mechanism. This feature is enabled per specific calls using IP Groups. To configure this feature, use the 'Always Use Source Address' parameter in the IP Group table (see "Configuring IP Groups" on page 174). If this feature is disabled, the device's NAT detection is according to the settings of the global parameter, 'SIP NAT Detection' parameter.

13.10.2.2 Media (RTP/RTCP/T.38)

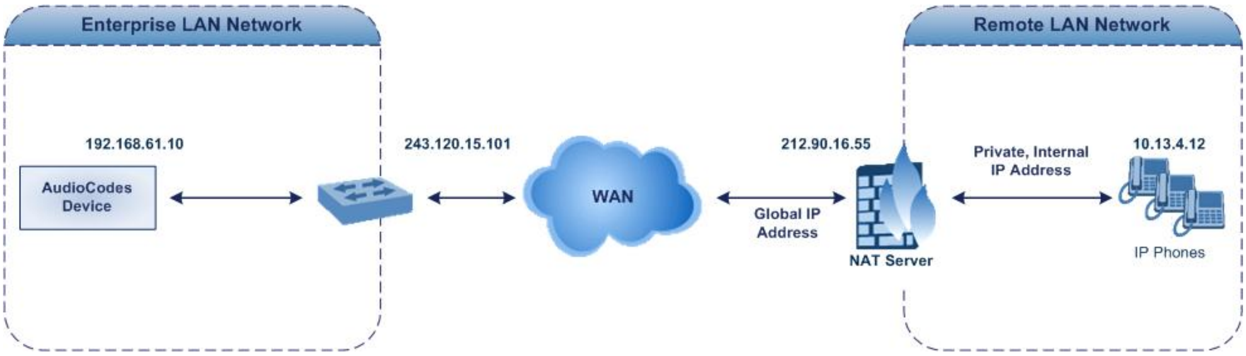
When a remote UA initiates a call and is not located behind a NAT server, the device sends the RTP (or RTCP, T.38) packets to the remote UA using the IP address / UDP port in the SIP message (Contact header). However, if the UA is located behind NAT, the device sends the RTP with the IP address of the UA (i.e., private IP address) as the destination, instead of that of the NAT server. Thus, the RTP will not reach the UA.

To resolve this NAT traversal problem, the device offers the following features:

- First Incoming Packet Mechanism - see "First Incoming Packet Mechanism" on page 113
- RTP No-Op packets according to the avt-rtp-noop draft - see "No-Op Packets" on page 113

The figure below illustrates a typical network architecture where the remote UA is located behind NAT:

Figure 47: Remote UA behind NAT



13.10.2.2.1 First Incoming Packet Mechanism

If the remote UA resides behind a NAT device, it's possible that the device can activate the RTP/RTCP/T.38 streams to an invalid IP address / UDP port. To avoid such cases, the device automatically compares the source address of the first received incoming RTP/RTCP/T.38 stream from the remote UA with the IP address and UDP port of the remote UA when the session was initially opened (i.e., originally received INVITE). If they are different, the device sets the destination IP address of the outgoing RTP packets to the source IP address of the first incoming packet. The RTP, RTCP and T.38 can thus have independent destination IP addresses and UDP ports.

➢ To enable NAT resolution using the First Incoming Packet mechanism:

1. Open the General Media Settings page (**Configuration** tab > **VoIP** menu > **Media** > **General Media Settings**).
2. Set the 'NAT Traversal' parameter to **Enable**.
3. Click **Submit**.

13.10.2.2.2 No-Op Packets

The device's No-Op packet support can be used to verify Real-Time Transport Protocol (RTP) and T.38 connectivity, and to keep NAT bindings and Firewall pinholes open. The No-Op packets are available for sending in RTP and T.38 formats.

You can control the activation of No-Op packets by using the *ini* file parameter NoOpEnable. If No-Op packet transmission is activated, you can control the time interval in which No-Op packets are sent in the case of silence (i.e., no RTP or T.38 traffic). This is done using the *ini* file parameter NoOpInterval. For a description of the RTP No-Op *ini* file parameters, see "Networking Parameters" on page 391.

- **RTP No-Op:** The RTP No-Op support complies with IETF Internet-Draft draft-wing-avt-rtp-noop-03 ("A No-Op Payload Format for RTP"). This IETF document defines a No-Op payload format for RTP. The draft defines the RTP payload type as dynamic. You can control the payload type with which the No-Op packets are sent. This is performed using the RTPNoOpPayloadType *ini* parameter (see "Networking Parameters" on page 391). The default payload type is 120.
- **T.38 No-Op:** T.38 No-Op packets are sent only while a T.38 session is activated. Sent packets are a duplication of the previously sent frame (including duplication of the sequence number).

Note: Receipt of No-Op packets is always supported.

13.11 Robust Receipt of Media Streams by Media Latching

The Robust Media mechanism (or media latching) is an AudioCodes proprietary mechanism to filter out unwanted media (RTP, RTCP, SRTP, SRTCP, and T.38) streams that are sent to the same port number of the device. Media ports may receive additional multiple unwanted media streams (from multiple sources of traffic) as result of traces of previous calls, call control errors, or deliberate malicious attacks (e.g., Denial of Service). When the device receives more than one media stream on the same port, the Robust Media mechanism detects the valid media stream and ignores the rest. Thus, this can prevent an established call been stolen by a malicious attacker on the media stream.

For the involved voice channel, the device latches onto the first stream of the first received packet. All packets (of any media type) received from the same IP address and SSRC are accepted (for T.38 packets, the device considers only the IP address). If the channel receives subsequent packets from a non-latched source, the device can either ignore this new stream and remain latched to the first original stream (IP address:port), or it can latch onto this new stream. The media latch mode is configured using the InboundMediaLatchMode parameter. If this mode is configured to latch onto new streams, you also need to configure the following:

- Minimum number of continuous media packets that need to be received from a different source(s) before the channel can latch onto this new incoming stream.
- Period (msec) during which if no packets are received from the current stream, the channel latches onto the next packet received from any other stream.

Depending on media latch mode, if the device has latched onto a new stream and a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this original stream.

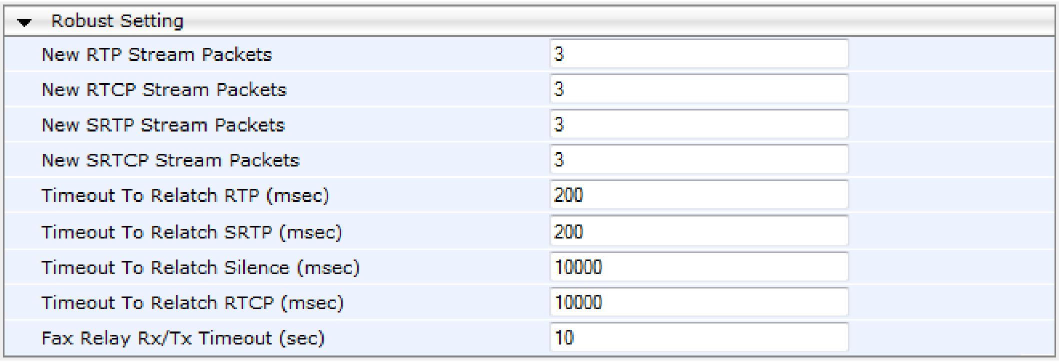
Latching onto a new T.38 stream is reported in CDR using the CDR fields, LatchedT38Ip (new IP address) and LatchedT38Port (new port). In addition, the SIP PUBLISH message updates the latched RTP SSRC, for example:

```
RemoteAddr: IP=10.33.2.55 Port=4000 SSRC=0x66d510ec
```

➢ To configure media latching:

1. Define the Robust Media method, using the InboundMediaLatchMode ini file parameter.
2. Open the General Settings page (Configuration tab > VoIP menu > Media > General Media Settings).

Figure 48: General Settings Page - Robust Setting



▾ Robust Setting

New RTP Stream Packets	3
New RTCP Stream Packets	3
New SRTP Stream Packets	3
New SRTCP Stream Packets	3
Timeout To Relatch RTP (msec)	200
Timeout To Relatch SRTP (msec)	200
Timeout To Relatch Silence (msec)	10000
Timeout To Relatch RTCP (msec)	10000
Fax Relay Rx/Tx Timeout (sec)	10

3. If you have set the InboundMediaLatchMode parameter to 1 or 2, scroll down to the Robust Settings group and do the following:
   - Define the minimum number of continuous media (RTP, RTCP, SRTP, and SRTCP) packets that need to be received by the channel before it can latch onto this new incoming stream:
     - 'New RTP Stream Packets'

- ♦ 'New RTCP Stream Packets'
- ♦ 'New SRTP Stream Packets'
- ♦ 'New SRTCP Stream Packets'

- Define a period (msec) during which if no packets are received from the current media session, the channel can re-latch onto another stream:
  - ♦ 'Timeout To Relatch RTP'
  - ♦ 'Timeout To Relatch SRTP'
  - ♦ 'Timeout To Relatch Silence'
  - ♦ 'Timeout To Relatch RTCP'
  - ♦ 'Fax Relay Rx/Tx Timeout'

4. Click Submit, and then save ("burn") your settings to flash memory.

For a detailed description of the robust media parameters, see "General Security Parameters" on page 406.

13.12 Multiple Routers Support

Multiple routers support is designed to assist the device when it operates in a multiple routers network. The device learns the network topology by responding to Internet Control Message Protocol (ICMP) redirections and caches them as routing rules (with expiration time).

When a set of routers operating within the same subnet serve as devices to that network and intercommunicate using a dynamic routing protocol, the routers can determine the shortest path to a certain destination and signal the remote host the existence of the better route. Using multiple router support, the device can utilize these router messages to change its next hop and establish the best path.

Note: Multiple Routers support is an integral feature that doesn't require configuration.

14 Security

This section describes the VoIP security-related configuration.

14.1 Configuring Firewall Settings

The Firewall Settings table lets you configure the device's Firewall, which defines network traffic filtering rules (*access list*). You can add up to 50 firewall rules. The access list offers the following firewall possibilities:

- Block traffic from known malicious sources
- Allow traffic only from known "friendly" sources, and block all other traffic
- Mix allowed and blocked network sources
- Limit traffic to a user-defined rate (blocking the excess)
- Limit traffic to specific protocols, and specific port ranges on the device

For each packet received on the network interface, the table is scanned from top to bottom until the first matching rule is found. This rule can either permit (*allow*) or deny (*block*) the packet. Once a rule in the table is located, subsequent rules further down the table are ignored. If the end of the table is reached without a match, the packet is accepted.

Notes:

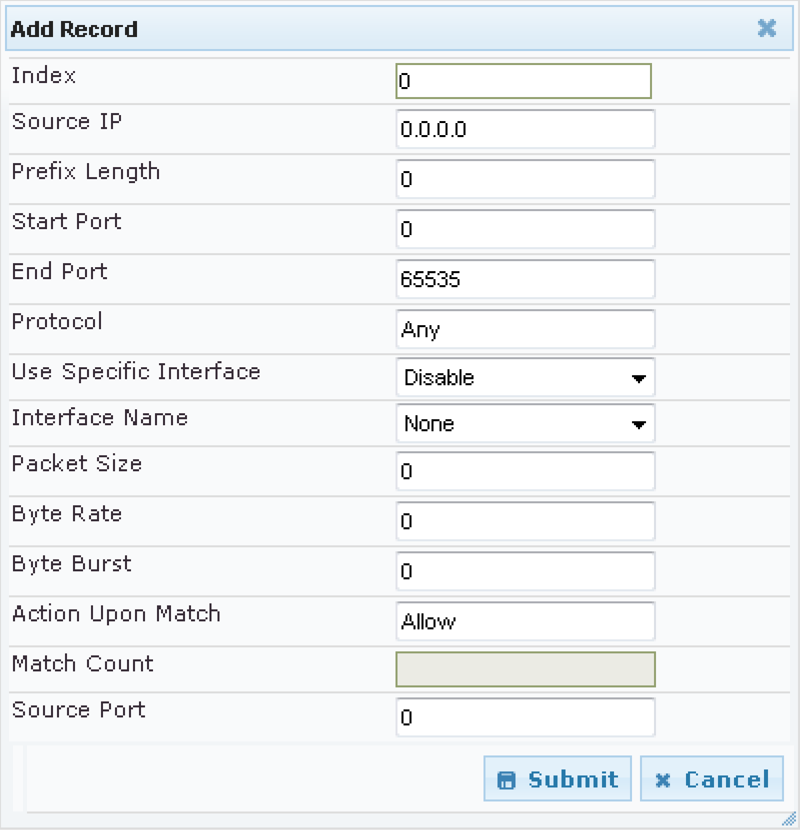
- This firewall applies to a very low-level network layer and overrides all your other security-related configuration. Thus, if you have configured higher-level security features (e.g., on the Application level), you must also configure firewall rules to permit this necessary traffic. For example, if you have configured IP addresses to access the Web and Telnet interfaces in the Web Access List (see "Configuring Web and Telnet Access List" on page 51), you must configure a firewall rule that permits traffic from these IP addresses.
- Only Security Administrator users or Master users can configure firewall rules.
- Setting the 'Prefix Length' field to **0** means that the rule applies to **all** packets, regardless of the defined IP address in the 'Source IP' field. Thus, it is highly recommended to set this parameter to a value other than 0.
- It is recommended to add a rule at the end of your table that blocks all traffic and to add firewall rules above it that allow required traffic (with bandwidth limitations). To block all traffic, use the following firewall rule:
  - √ Source IP: 0.0.0.0
  - √ Prefix Length: 0 (i.e., rule matches all IP addresses)
  - √ Start Port - End Port: 0-65535
  - √ Protocol: **Any**
  - √ Action Upon Match: **Block**
- If you are using the High Availability feature and you have configured "block" rules, ensure that you also add "allow" rules for HA traffic. For more information, see Configuring Firewall Allowed Rules on page 284.

The procedure below describes how to configure Firewall rules in the Web interface. You can also configure this using the table ini file parameter, AccessList or the CLI command, configure voip/access-list.

➢ To configure a Firewall rule:

1. Open the Firewall Settings page (**Configuration** tab > **VoIP** menu > **Security** > **Firewall Settings**).
2. Click **Add**; the following dialog box appears:

Figure 49: Firewall Settings Page - Add Record



Add Record

Index	0
Source IP	0.0.0.0
Prefix Length	0
Start Port	0
End Port	65535
Protocol	Any
Use Specific Interface	Disable
Interface Name	None
Packet Size	0
Byte Rate	0
Byte Burst	0
Action Upon Match	Allow
Match Count	
Source Port	0

3. Configure a Firewall rule according to the parameters described in the table below.
4. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

Table 39: Firewall Settings Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Source IP CLI: source-ip **[AccessList_Source_IP]**	Defines the IP address (or DNS name) or a specific host name of the source network (i.e., from where the incoming packet is received).
Source Port CLI: src-port **[AccessList_Source_Port]**	Defines the source UDP/TCP ports (of the remote host) from where packets are sent to the device. The valid range is 0 to 65535. **Note:** When set to 0, this field is ignored and any source port matches the rule.

Parameter	Description
Prefix Length CLI: prefixLen **[AccessList_PrefixLen]**	(**Mandatory**) Defines the IP network mask - 32 for a single host or the appropriate value for the source IP addresses. ▪ A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0). ▪ A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0). ▪ A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0). The IP address of the sender of the incoming packet is trimmed in accordance with the prefix length (in bits) and then compared to the parameter 'Source IP'. The default is 0 (i.e., applies to all packets). You **must** change this value to any of the above options. **Note:** A value of 0 applies to **all** packets, regardless of the defined IP address. Therefore, you must set this parameter to a value other than 0.
Start Port CLI: start-port **[AccessList_Start_Port]**	Defines the destination UDP/TCP start port (on this device) to where packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.
End Port CLI: end-port **[AccessList_End_Port]**	Defines the destination UDP/TCP end port (on this device) to where packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.
Protocol CLI: protocol **[AccessList_Protocol]**	Defines the protocol type (e.g., UDP, TCP, ICMP, ESP or 'Any') or the IANA protocol number in the range of 0 (Any) to 255. **Note:** This field also accepts the abbreviated strings 'SIP' and 'HTTP'. Specifying these strings implies selection of the TCP or UDP protocols, and the appropriate port numbers as defined on the device.
Use Specific Interface CLI: use-specific-interface **[AccessList_Use_Specific_Interface]**	Determines whether you want to apply the rule to a specific network interface defined in the Interface table (i.e., packets received from that defined in the Source IP field and received on this network interface): ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If enabled, then in the 'Interface Name' field (described below), select the interface to which the rule is applied. ▪ If disabled, then the rule applies to all interfaces.
Interface Name CLI: network-interface-name **[AccessList_Interface_x]**	Defines the network interface to which you want to apply the rule. This is applicable if you enabled the 'Use Specific Interface' field. The list displays interface names as defined in the Interface table in "Configuring IP Network Interfaces" on page 91.
Packet Size CLI: packet-size **[AccessList_Packet_Size]**	Defines the maximum allowed packet size. The valid range is 0 to 65535. **Note:** When filtering fragmented IP packets, this field relates to the overall (re-assembled) packet size, and not to the size of each fragment.
Byte Rate CLI: byte-rate **[AccessList_Byte_Rate]**	Defines the expected traffic rate (bytes per second), i.e., the allowed bandwidth for the specified protocol. In addition to this field, the 'Burst Bytes' field provides additional allowance such that momentary bursts of data may

Parameter	Description
	utilize more than the defined byte rate, without being interrupted. For example, if 'Byte Rate' is set to 40000 and 'Burst Bytes' to 50000, then this implies the following: the allowed bandwidth is 40000 bytes/sec with extra allowance of 50000 bytes; if, for example, the actual traffic rate is 45000 bytes/sec, then this allowance would be consumed within 10 seconds, after which all traffic exceeding the allocated 40000 bytes/sec is dropped. If the actual traffic rate then slowed to 30000 bytes/sec, then the allowance would be replenished within 5 seconds.
Burst Bytes CLI: byte-burst **[AccessList_Byte_Burst]**	Defines the tolerance of traffic rate limit (number of bytes). The default is 0.
Action Upon Match CLI: allow-type **[AccessList_Allow_Type]**	Defines the firewall action to be performed upon rule match. ▪ "Allow" = (Default) Permits these packets ▪ "Block" = Rejects these packets
Match Count **[AccessList_MatchCount]**	(Read-only) Displays the number of packets accepted or rejected by the rule.

The table below provides an example of configured firewall rules:

Table 40: Configuration Example of Firewall Rules

Parameter	Firewall Rule				
	1	2	3	4	5
Source IP	12.194.231.76	12.194.230.7	0.0.0.0	192.0.0.0	0.0.0.0
Prefix Length	16	16	0	8	0
Start Port and End Port	0-65535	0-65535	0-65535	0-65535	0-65535
Protocol	Any	Any	icmp	Any	Any
Use Specific Interface	Enable	Enable	Disable	Enable	Disable
Interface Name	WAN	WAN	None	Voice-Lan	None
Byte Rate	0	0	40000	40000	0
Burst Bytes	0	0	50000	50000	0
Action Upon Match	Allow	Allow	Allow	Allow	Block

The firewall rules in the above configuration example do the following:

- **Rules 1 and 2:** Typical firewall rules that allow packets ONLY from specified IP addresses (e.g., proxy servers). Note that the prefix length is configured.
- **Rule 3:** A more "advanced" firewall rule - bandwidth rule for ICMP, which allows a maximum bandwidth of 40,000 bytes/sec with an additional allowance of 50,000 bytes. If, for example, the actual traffic rate is 45,000 bytes/sec, then this allowance would be consumed within 10 seconds, after which all traffic exceeding the allocated 40,000 bytes/sec is dropped. If the actual traffic rate then slowed to 30,000 bytes/sec, the allowance would be replenished within 5 seconds.
- **Rule 4:** Allows traffic from the LAN voice interface and limits bandwidth.
- **Rule 5:** Blocks all other traffic.

14.2 Configuring General Security Settings

The Transport Layer Security (TLS) parameters are configured in the General Security Settings page. TLS, also known as Secure Socket Layer (SSL), is used to secure the device's SIP signaling connections, Web interface, and Telnet server. The SSL protocol provides confidentiality, integrity, and authenticity between two communicating applications over TCP/IP.

Specifications for the SSL/TLS implementation include the following:

- Transport: SSL 2.0, SSL 3.0, TLS 1.0
- Cipher: DES, RC4 compatible, Advanced Encryption Standard (AES)
- Authentication: X.509 certificates (CRLs are currently not supported)

The device uses TLS over TCP to encrypt SIP transport and (optionally) to authenticate it. To enable TLS on the device, set the transport type to TLS (SIPTransportType = 2). In this mode, the device initiates a TLS connection only for the next network hop. To enable TLS all the way to the destination (over multiple hops), set EnableSIPS to 1. When a TLS connection with the device is initiated, the device also responds using TLS, regardless of the configured SIP transport type (in this case, the parameter EnableSIPS is also ignored). To change the port number used for SIPS transport (by default, 5061), use the parameter TLSLocalSIPPort. TLS and SIPS use the Certificate Exchange process, as described in "Configuring Certificates" on page 76.

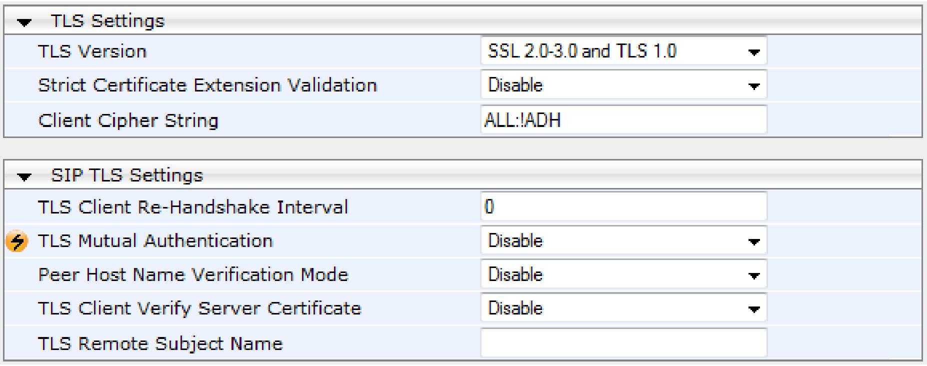
When SIPS is implemented, it is sometimes required to use two-way authentication. When acting as the TLS server (in a specific connection), it is possible to demand the authentication of the client's certificate. To enable two-way authentication on the device, set the ini file parameter SIPSRequireClientCertificate to 1. For information on installing a client certificate, refer to Client Certificates described in the User's Manual.

For a description of the TLS parameters, see "TLS Parameters" on page 410.

➢ To configure TLS parameters:

1. Open the General Security Settings page (**Configuration** tab > **VoIP** menu > **Security** > **General Security Settings**).

Figure 50: TLS Parameters on General Security Settings Page



TLS Settings

TLS Version	SSL 2.0-3.0 and TLS 1.0
Strict Certificate Extension Validation	Disable
Client Cipher String	ALL:!ADH

SIP TLS Settings

TLS Client Re-Handshake Interval	0
TLS Mutual Authentication	Disable
Peer Host Name Verification Mode	Disable
TLS Client Verify Server Certificate	Disable
TLS Remote Subject Name	

2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, refer to "Saving Configuration" on page 295.

14.3 Intrusion Detection System

The device's Intrusion Detection System (IDS) feature detects malicious attacks on the device and reacts accordingly. A remote host is considered malicious if it has reached or exceeded a user-defined threshold (counter) of specified malicious attacks.

If malicious activity is detected, the device can do the following:

- Block (blacklist) remote hosts (IP addresses / ports) considered by the device as malicious. The device automatically blacklists the malicious source for a user-defined period after which it is removed from the blacklist. When an attacker is added to or removed from the blacklist, the device sends the following new SNMP trap

- Send SNMP traps to notify of malicious activity and/or whether an attacker has been added to or removed from the blacklist. For more information, see "Viewing IDS Alarms" on page 126.

The Intrusion Detection System (IDS) is an important feature for Enterprises to ensure legitimate calls are not being adversely affected by attacks and to prevent Theft of Service and unauthorized access.

There are many types of malicious attacks, the most common being:

- **Denial of service:** This can be Denial of Service (DoS) where an attacker wishing to prevent a server from functioning correctly directs a large amount of requests – sometimes meaningless and sometimes legitimate, or it can be Distributed Denial of Service (DDoS) where the attacker controls a large group of systems to coordinate a large scale DoS attack against a system:
  - Message payload tampering: Attacker may inject harmful content into a message, e.g., by entering meaningless or wrong information, with the goal of exploiting a buffer overflow at the target. Such messages can be used to probe for vulnerabilities at the target.
  - Message flow tampering: This is a special case of DoS attacks. These attacks disturb the ongoing communication between users. An attacker can then target the connection by injecting fake signaling messages into the communication channel (such as CANCEL messages).
  - Message Flooding: The most common DoS attack is where an attacker sends a huge amount of messages (e.g., INVITEs) to a target. The goal is to overwhelm the target's processing capabilities, thereby rendering the target inoperable.
- **SPAM over Internet Telephony (SPIT):** VoIP spam is unwanted, automatically dialed, pre-recorded phone calls using VoIP. It is similar to e-mail spam.
- **Theft of Service (ToS):** Service theft can be exemplified by phreaking, which is a type of hacking that steals service (i.e., free calls) from a service provider, or uses a service while passing the cost to another person.

The IDS configuration is based on IDS Policies, where each policy can be configured with a set of IDS rules. Each rule defines a type of malicious attack to detect and the number of attacks during an interval (threshold) before an SNMP trap is sent. Each policy is then applied to a target under attack (SIP interface) and/or source of attack (Proxy Set and/or subnet address).

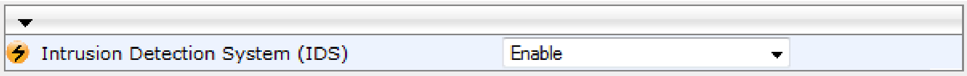
14.3.1 Enabling IDS

The procedure below describes how to enable IDS.

➢ To enable IDS:

1. Open the IDS Global Parameters page (**Configuration** tab > **VoIP** menu > **Security** > **Intrusion Detection and Prevention** > **Global Parameters**).

Figure 51: Enabling IDS on IDS Global Parameters Page



Intrusion Detection System (IDS): Enable

2. From the 'Intrusion Detection System' drop-down list, select **Enable**.
3. Click **Submit**, and then reset the device with a burn-to-flash for the setting to take effect.

14.3.2 Configuring IDS Policies

Configuring IDS Policies is a two-stage process that includes the following tables:

1. **IDS Policy (parent table):** Defines a name and description for the IDS Policy. You can configure up to 20 IDS Policies.

2. **IDS Rules table (child table):** Defines the actual rules for the IDS Policy. Each IDS Policy can be configured with up to 20 rules.

> **Note:** A maximum of 100 IDS rules can be configured (regardless of how many rules are assigned to each policy).

The device provides the following pre-configured IDS Policies that can be used in your deployment (if they meet your requirements):

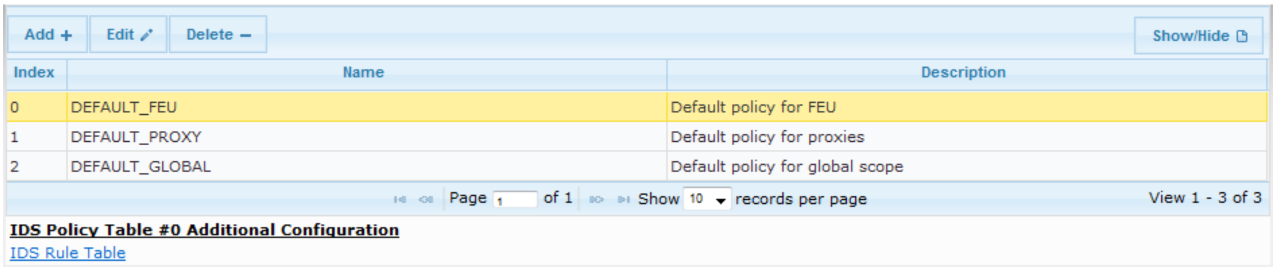
- "DEFAULT_FEU": IDS Policy for far-end users in the WAN
- "DEFAULT_PROXY": IDS Policy for proxy server
- "DEFAULT_GLOBAL": IDS Policy with global thresholds

These default IDS Policies are read-only and cannot be modified.

➢ To configure an IDS Policy:

1. Open the IDS Policy Table page (**Configuration** tab > **VoIP** menu > **Security** > **Intrusion Detection and Prevention** > **Policy Table**); the table shows the pre-configured IDS policies:

Figure 52: IDS Policy Table with Default Rules



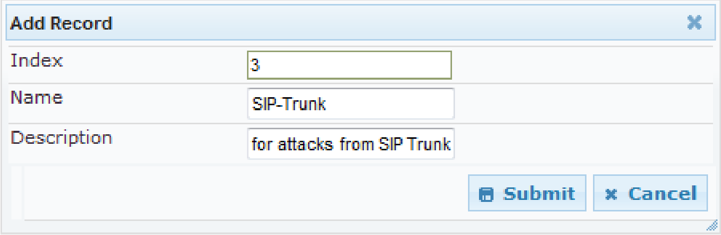
Index	Name	Description
0	DEFAULT_FEU	Default policy for FEU
1	DEFAULT_PROXY	Default policy for proxies
2	DEFAULT_GLOBAL	Default policy for global scope

Page 1 of 1 Show 10 records per page View 1 - 3 of 3

IDS Policy Table #0 Additional Configuration
IDS Rule Table

2. Click **Add**; the following dialog box appears:

Figure 53: IDS Policy Table - Add Record


Add Record

Index	3
Name	SIP-Trunk
Description	for attacks from SIP Trunk

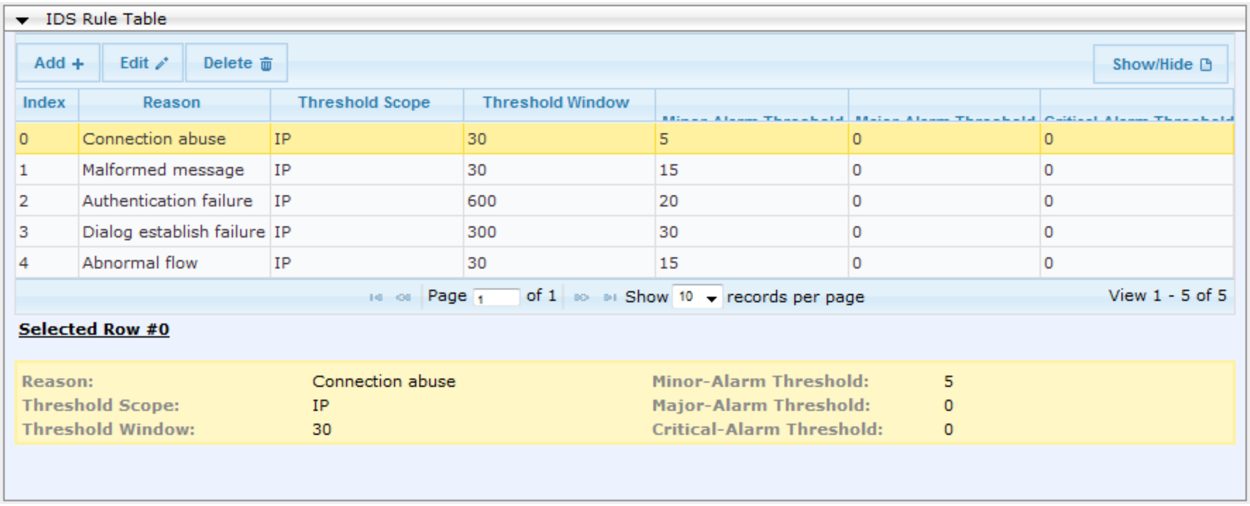
3. Configure an IDS Policy name according to the parameters described in the table below.
4. Click **Submit**.

Table 41: IDS Policy Table Parameter Descriptions

Parameter	Description
Index CLI: policy **[IDSPolicy_Index]**	Defines an index number for the new table record.
Name CLI: rule **[IDSPolicy_Description]**	Defines an arbitrary name to easily identify the IDS Policy. The valid value is a string of up to 20 characters.
Description **[IDSPolicy_Name]**	Defines a brief description for the IDS Policy. The valid value is a string of up to 100 characters.

5. In the IDS Policy table, select the required IDS Policy row, and then click the **IDS Rule Table** link located below the table; the IDS Rule table opens:

Figure 54: IDS Rule Table of Selected IDS Policy



Index	Reason	Threshold Scope	Threshold Window	Minor-Alarm Threshold	Major-Alarm Threshold	Critical-Alarm Threshold
0	Connection abuse	IP	30	5	0	0
1	Malformed message	IP	30	15	0	0
2	Authentication failure	IP	600	20	0	0
3	Dialog establish failure	IP	300	30	0	0
4	Abnormal flow	IP	30	15	0	0

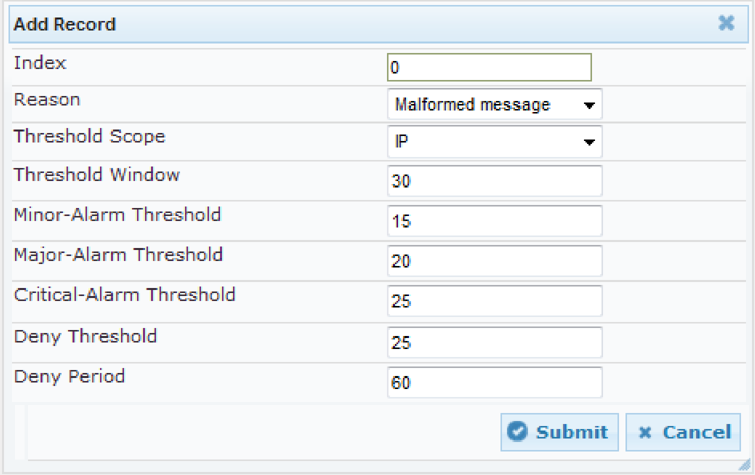
Page 1 of 1 Show 10 records per page View 1 - 5 of 5

Selected Row #0

Reason:	Connection abuse	Minor-Alarm Threshold:	5
Threshold Scope:	IP	Major-Alarm Threshold:	0
Threshold Window:	30	Critical-Alarm Threshold:	0

6. Click **Add**; the following dialog box appears:

Figure 55: IDS Rule Table - Add Record



Add Record

Index	0
Reason	Malformed message
Threshold Scope	IP
Threshold Window	30
Minor-Alarm Threshold	15
Major-Alarm Threshold	20
Critical-Alarm Threshold	25
Deny Threshold	25
Deny Period	60

The figure above shows a configuration example. If 15 malformed SIP messages are received within a period of 30 seconds, a minor alarm is sent. Every 30 seconds, the rule's counters are cleared. In addition, if more than 25 malformed SIP messages are received within this period, the device blacklists the remote IP host from where the messages were received for 60 seconds.

7. Configure an IDS Rule according to the parameters described in the table below.

8. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 42: IDS Rule Table Parameter Descriptions

Parameter	Description
Index CLI: rule-id **[IDSRule_RuleID]**	Defines an index number for the new table record.
Reason CLI: reason **[IDSRule_Reason]**	Defines the type of intrusion attack (malicious event). ▪ **[0]** Any = All events listed below are considered as attacks and are counted together. ▪ **[1]** Connection abuse (default) = TLS authentication failure.

Parameter	Description
	▪ **[2]** Malformed message = ✓ Message exceeds a user-defined maximum message length (50K) ✓ Any SIP parser error ✓ Message Policy match (see "Configuring SIP Message Policy Rules") ✓ Basic headers not present ✓ Content length header not present (for TCP) ✓ Header overflow ▪ **[3]** Authentication failure = ✓ Local authentication ("Bad digest" errors) ✓ Remote authentication (SIP 401/407 is sent if original message includes authentication) ▪ **[4]** Dialog establish failure = ✓ Classification failure (see "Configuring Classification Rules" on page 249) ✓ Routing failure ✓ Other local rejects (prior to SIP 180 response) ✓ Remote rejects (prior to SIP 180 response) ▪ **[5]** Abnormal flow = ✓ Requests and responses without a matching transaction user (except ACK requests) ✓ Requests and responses without a matching transaction (except ACK requests)
Threshold Scope CLI: threshold-scope **[IDSRule_ThresholdScope]**	Defines the source of the attacker to consider in the device's detection count. ▪ **[0]** Global = All attacks regardless of source are counted together during the threshold window. ▪ **[2]** IP = Attacks from each specific IP address are counted separately during the threshold window. ▪ **[3]** IP+Port = Attacks from each specific IP address:port are counted separately during the threshold window. This option is useful for NAT servers, where numerous remote machines use the same IP address but different ports. However, it is not recommended to use this option as it may degrade detection capabilities.
Threshold Window CLI: threshold-window **[IDSRule_ThresholdWindow]**	Defines the threshold interval (in seconds) during which the device counts the attacks to check if a threshold is crossed. The counter is automatically reset at the end of the interval. The valid range is 1 to 1,000,000. The default is 1.
Minor-Alarm Threshold CLI: minor-alrm-thr **[IDSRule_MinorAlarmThreshold]**	Defines the threshold that if crossed a minor severity alarm is sent. The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.
Major-Alarm Threshold CLI: major-alrm-thr **[IDSRule_MajorAlarmThreshold]**	Defines the threshold that if crossed a major severity alarm is sent. The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.
Critical-Alarm Threshold CLI: critical-alrm-thr **[IDSRule_CriticalAlarmThreshold]**	Defines the threshold that if crossed a critical severity alarm is sent. The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.
Deny Threshold **[IDSRule_DenyThreshold]**	Defines the threshold that if crossed, the device blocks (blacklists) the remote host (attacker). The default is -1 (i.e., not configured). **Note:** This parameter is applicable only if the 'Threshold Scope' parameter is set to **IP** or **IP+Port**.
Deny Period	Defines the duration (in sec) to keep the attacker on the blacklist.

Parameter	Description
[IDSRule_DenyPeriod]	The valid range is 0 to 1,000,000. The default is -1 (i.e., not configured).

14.3.3 Assigning IDS Policies

The IDS Match table lets you implement your configured IDS Policies. You do this by assigning specific IDS Policies to any, or a combination of, the following configuration entities:

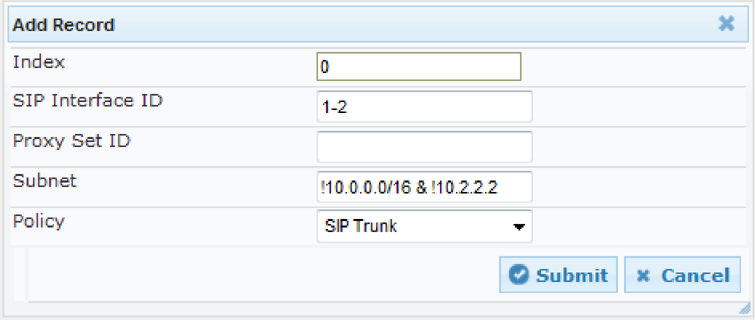
- **SIP Interface:** For detection of malicious attacks on specific SIP Interface(s). For configuring SIP Interfaces, see "Configuring SIP Interfaces" on page 171.
- **Proxy Sets:** For detection of malicious attacks from specified Proxy Set(s). For configuring Proxy Sets, see "Configuring Proxy Sets" on page 182.
- **Subnet addresses:** For detection of malicious attacks from specified subnet addresses.

You can configure up to 20 IDS Policy-Matching rules.

➢ To configure an IDS Policy-Matching rule:

1. Open the IDS Match Table page (**Configuration** tab > **VoIP** menu > **Security** > **Intrusion Detection and Prevention** > **Match Table**).
2. Click **Add**; the following dialog box appears:

Figure 56: IDS Match Table - Add Record



Index	0
SIP Interface ID	1-2
Proxy Set ID	
Subnet	!10.0.0.0/16 & !10.2.2.2
Policy	SIP Trunk

The figure above shows a configuration example where the IDS Policy "SIP Trunk" is applied to SIP Interfaces 1 and 2, and all source IP addresses outside of subnet 10.1.0.0/16 and IP address 10.2.2.2.

3. Configure a rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 43: IDS Match Table Parameter Descriptions

Parameter	Description
Index **[IDSMatch_Index]**	Defines an index number for the new table record.
SIP Interface ID CLI: sip-interface **[IDSMatch_SIPInterface]**	Defines the SIP Interface(s) to which you want to assign the IDS Policy. This indicates the SIP Interfaces that are being attacked. The valid value is the ID of the SIP Interface. The following syntax is supported: ▪ A comma-separated list of SIP Interface IDs (e.g., 1,3,4) ▪ A hyphen "-" indicates a range of SIP Interfaces (e.g., 3,4-7 means IDs 3, and 4 through 7) ▪ A prefix of an exclamation mark "!" means negation of the set (e.g., !3,4-7 means all indexes excluding 3, and excluding 4 through 7)
Proxy Set ID	Defines the Proxy Set(s) to which the IDS Policy is assigned. This indicates the

Parameter	Description
CLI: proxy-set **[IDSMatch_ProxySet]**	Proxy Sets from where the attacks are coming from. The following syntax is supported: ▪ A comma-separated list of Proxy Set IDs (e.g., 1,3,4) ▪ A hyphen "-" indicates a range of Proxy Sets (e.g., 3,4-7 means IDs 3, and 4 through 7) ▪ A prefix of an exclamation mark "!" means negation of the set (e.g., !3,4-7 means all indexes excluding 3, and excluding 4 through 7) **Notes:** ▪ Only the IP address of the Proxy Set is considered (not port). ▪ If a Proxy Set has multiple IP addresses, the device considers the Proxy Set as one entity and includes all its IP addresses in the same IDS count.
Subnet CLI: subnet **[IDSMatch_Subnet]**	Defines the subnet to which the IDS Policy is assigned. This indicates the subnets from where the attacks are coming from. The following syntax can be used: ▪ Basic syntax is a subnet in CIDR notation (e.g., 10.1.0.0/16 means all sources with IP address in the range 10.1.0.0–10.1.255.255) ▪ An IP address can be specified without the prefix length to refer to the specific IP address. ▪ Each subnet can be negated by prefixing it with "!", which means all IP addresses outside that subnet. ▪ Multiple subnets can be specified by separating them with "&" (and) or "|" (or) operations. For example: ✓ 10.1.0.0/16 | 10.2.2.2: includes subnet 10.1.0.0/16 and IP address 10.2.2.2. ✓ !10.1.0.0/16 & !10.2.2.2: includes all addresses except those of subnet 10.1.0.0/16 and IP address 10.2.2.2. Note that the exclamation mark "!" appears before each subnet. ✓ 10.1.0.0/16 & !10.1.1.1: includes subnet 10.1.0.0/16, except IP address 10.1.1.1.
Policy CLI: policy **[IDSMatch_Policy]**	Assigns an IDS Policy (configured in ''Configuring IDS Policies'' on page 121).

14.3.4 Viewing IDS Alarms

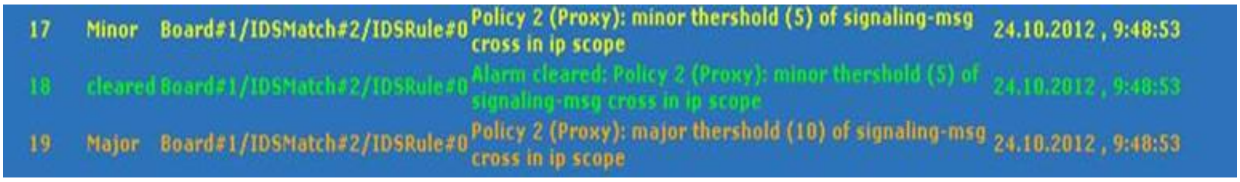
For the IDS feature, the device sends the following SNMP traps:

- Traps that notify the detection of malicious attacks:
  - **acIDSPolicyAlarm:** The device sends this alarm whenever a threshold of a specific IDS Policy rule is crossed. The trap displays the crossed severity threshold (Minor or Major), IDS Policy and IDS Rule, and the IDS Policy-Match index.
  - **acIDSThresholdCrossNotification:** The device sends this event for each scope (IP address) that crosses the threshold. In addition to the crossed severity threshold (Minor or Major) of the IDS Policy-Match index, this event shows the IP address (or IP address:port) of the malicious attacker.

    If the severity level is raised, the alarm of the former severity is cleared and the device sends a new alarm with the new severity. The alarm is cleared after a user-defined period (configured by the ini file parameter, IDSAlarmClearPeriod) during which no thresholds have been crossed. However, this "quiet" period must be at least twice the 'Threshold Window' value (configured in ''Configuring IDS Policies'' on page 121). For example, if you set IDSAlarmClearPeriod to 20 sec and 'Threshold Window' to 15 sec, the IDSAlarmClearPeriod parameter is ignored and the alarm is cleared only after 30 seconds (2 x 15 sec).

The figure below displays an example of IDS alarms in the Active Alarms table ("Viewing Active Alarms" on page 333). In this example, a Minor threshold alarm is cleared and replaced by a Major threshold alarm:

Figure 57: IDS Alarms in Active Alarms Table



17	Minor	Board#1/IDSMatch#2/IDSRule#0	Policy 2 (Proxy): minor thershold (5) of signaling-msg cross in ip scope	24.10.2012 , 9:48:53
18	cleared	Board#1/IDSMatch#2/IDSRule#0	Alarm cleared: Policy 2 (Proxy): minor thershold (5) of signaling-msg cross in ip scope	24.10.2012 , 9:48:53
19	Major	Board#1/IDSMatch#2/IDSRule#0	Policy 2 (Proxy): major thershold (10) of signaling-msg cross in ip scope	24.10.2012 , 9:48:53

- acIDSBlacklistNotification event: The device sends this event whenever an attacker (remote host at IP address and/or port) is added to or removed from the blacklist.

You can also view IDS alarms in the CLI, using the following commands:

- To view all active IDS alarms:

```
show voip security ids active-alarm all
```

- To view all IP addresses that crossed the threshold for an active IDS alarm:

```
show voip security ids active-alarm match * rule *
```

- To view the blacklist:

```
# show voip security ids blacklist active
```

For example:

```
Active blacklist entries:
  10.33.5.110(NI:0) remaining 00h:00m:10s in blacklist
```

Where SI is the SIP Interface and NI is the network interface.

The device also sends IDS notifications in Syslog messages to a Syslog server. This only occurs if you have configured Syslog (see "Configuring Syslog" on page 358). The table below shows the Syslog text message per malicious event:

Table 44: Types of Malicious Events and Syslog Text String

Type	Description	Syslog String
Connection Abuse	TLS authentication failure	abuse-tls-auth-fail
Malformed Messages	▪ Message exceeds a user-defined maximum message length (50K) ▪ Any SIP parser error ▪ Message policy match ▪ Basic headers not present ▪ Content length header not present (for TCP) ▪ Header overflow	▪ malformed-invalid-msg-len ▪ malformed-parse-error ▪ malformed-message-policy ▪ malformed-miss-header ▪ malformed-miss-content-len ▪ malformed-header-overflow
Authentication Failure	▪ Local authentication ("Bad digest" errors) ▪ Remote authentication (SIP 401/407 is sent if original message includes authentication)	▪ auth-establish-fail ▪ auth-reject-response
Dialog Establishment Failure	▪ Classification failure ▪ Routing failure ▪ Other local rejects (prior to SIP 180 response) ▪ Remote rejects (prior to SIP 180 response)	▪ establish-classify-fail ▪ establish-route-fail ▪ establish-local-reject ▪ establish-remote-reject
Abnormal Flow	▪ Requests and responses without a matching transaction user (except ACK requests) ▪ Requests and responses without a matching transaction (except ACK requests)	▪ flow-no-match-tu ▪ flow-no-match-transaction

15 Media

This section describes the media-related configuration.

15.1 Configuring RTP/RTCP Settings

This section describes configuration relating to Real-Time Transport Protocol (RTP) and RTP Control Protocol (RTCP).

15.1.1 Configuring RTP Base UDP Port

You can configure the range of UDP ports for RTP, RTCP, and T.38. The UDP port range can be configured using media realms in the Media Realm table, allowing you to assign different port ranges (media realms) to different interfaces. However, if you do not use media realms, you can configure the lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2), using the 'RTP Base UDP Port' (BaseUDPport) parameter. For example, if the BaseUDPPort is set to 6000, then one channel may use the ports RTP 6000, RTCP 6001, and T.38 6002, while another channel may use RTP 6010, RTCP 6011, and T.38 6012.

The range of possible UDP ports is 6,000 to 64,000 (default base UDP port is 6000). The port range is calculated using the BaseUDPport parameter as follows: **BaseUDPPort to (BaseUDPPort + <channels - 1> * 10)**

The default local UDP ports for audio and fax media streams is calculated using the following formula: **BaseUDPPort + (Channel ID * 10) + Port Offset**

Where the port offsets are as follows:

- **Audio RTP:** 0
- **Audio RTCP:** 1
- **Fax T.38:** 2

For example, the local T.38 UDP port for channel 30 is calculated as follows: **6000 + (30*10) + 2 = 6302**

The maximum (when all channels are required) UDP port range is calculated as follows:

- BaseUDPport to (BaseUDPport + 4000*10)

Notes:

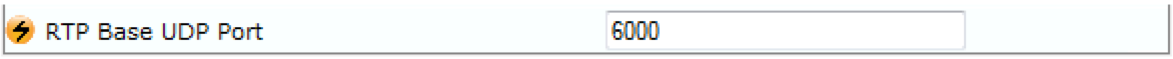
- The device allocates the UDP ports randomly to the channels.
- To configure the device to use the same port for both RTP and T.38 packets, set the T38UseRTPPort parameter to 1.
- If you are using Media Realms (see "Configuring Media Realms" on page 164), the port range configured for the Media Realm must be within this range defined by the BaseUDPPort parameter.

The procedure below describes how to configure the RTP base UDP port using the Web interface.

➢ To configure the RTP base UDP port:

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** > **RTP/RTCP Settings**). The relevant parameter is listed under the 'General Settings' group, as shown below:

Figure 58: RTP Based UDP Port in RTP/RTCP Settings Page



RTP Base UDP Port: 6000

2. Set the 'RTP Base UDP Port' parameter to the required value.
3. Click **Submit**.
4. Reset the device for the settings to take effect.

15.2 Configuring Media (SRTP) Security

The device supports Secured RTP (SRTP) according to RFC 3711. SRTP is used to encrypt RTP and RTCP transport for protecting VoIP traffic. SRTP requires a key exchange mechanism that is performed according to RFC 4568 – "Session Description Protocol (SDP) Security Descriptions for Media Streams". The key exchange is done by adding a 'crypto' attribute to the SDP. This attribute is used (by both sides) to declare the various supported cipher suites and to attach the encryption key. If negotiation of the encryption data is successful, the call is established.

SRTP supports the following cipher suites (all other suites are ignored):

- AES_CM_128_HMAC_SHA1_32
- AES_CM_128_HMAC_SHA1_80

When the device is the offering side, it generates an MKI of a size configured by the 'Master Key Identifier (MKI) Size' parameter. The length of the MKI is limited to four bytes. If the remote side sends a longer MKI, the key is ignored. The key lifetime field is not supported. However, if it is included in the key it is ignored and the call does not fail.

The device supports the following session parameters (as defined in RFC 4568, SDP Security Descriptions for Media Streams):

- UNENCRYPTED_SRTP
- UNENCRYPTED_SRTCP
- UNAUTHENTICATED_SRTP

Session parameters should be the same for the local and remote sides. When the device is the offering side, the session parameters are configured by the following parameter - 'Authentication On Transmitted RTP Packets', 'Encryption On Transmitted RTP Packets, and 'Encryption On Transmitted RTCP Packets'. When the device is the answering side, the device adjusts these parameters according to the remote offering. Unsupported session parameters are ignored, and do not cause a call failure.

Below is an example of crypto attributes usage:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80 inline:PsKoMpHlCg+b5X0YLuSvNrImEh/dAe
a=crypto:2 AES_CM_128_HMAC_SHA1_32
inline:IsPtLoGkBf9a+c6XVzRuMqHlDnEiAd
```

The device also supports symmetric MKI negotiation, whereby it can be configured to forward the MKI size received in the SDP offer crypto line in the SDP answer crypto line.

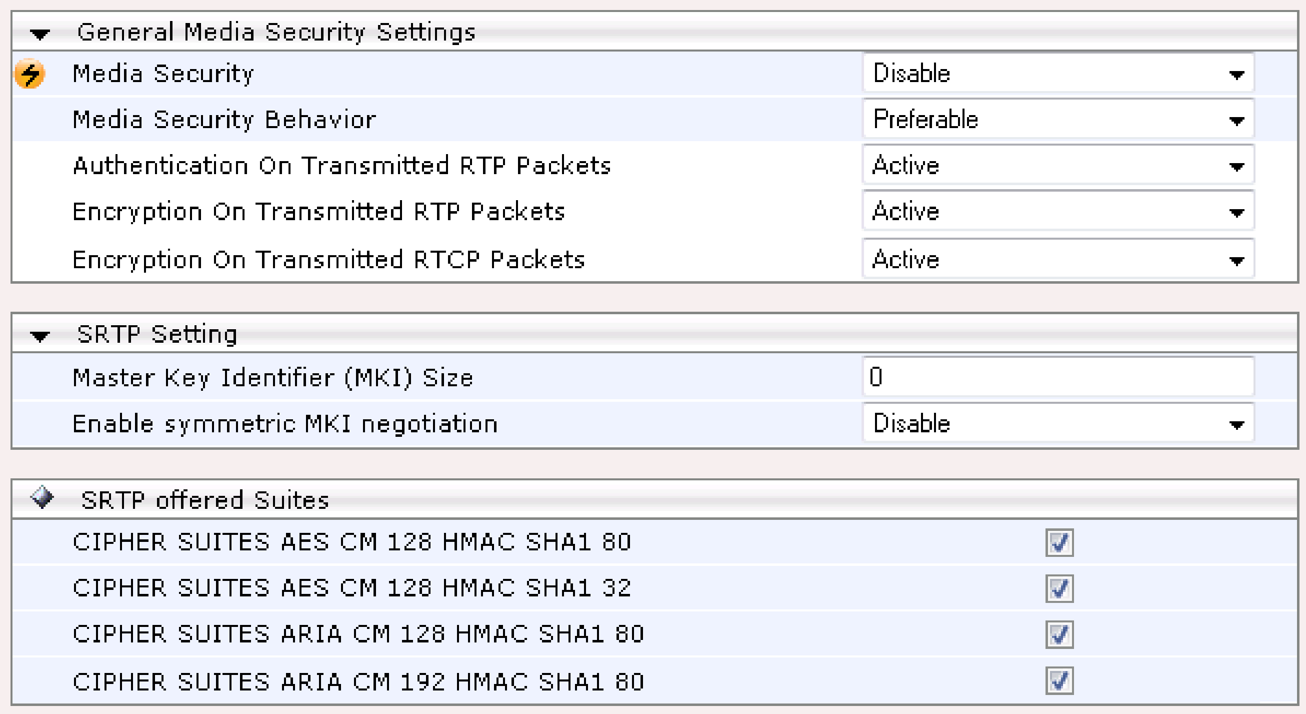
To configure the device's mode of operation if negotiation of the cipher suite fails, use the 'Media Security Behavior' parameter. This parameter can be set to enforce SRTP, whereby incoming calls that don't include encryption information are rejected.

Notes:

- For a detailed description of the SRTP parameters, see "SRTP Parameters" on page 408.
- When SRTP is used, the channel capacity may be reduced.

➢ To enable and configure SRTP:

1. Open the Media Security page (**Configuration** tab > **VoIP** menu > **Media** > **Media Security**).



General Media Security Settings

Media Security	Disable
Media Security Behavior	Preferable
Authentication On Transmitted RTP Packets	Active
Encryption On Transmitted RTP Packets	Active
Encryption On Transmitted RTCP Packets	Active

SRTP Setting

Master Key Identifier (MKI) Size	0
Enable symmetric MKI negotiation	Disable

SRTP offered Suites

CIPHER SUITES AES CM 128 HMAC SHA1 80	☑
CIPHER SUITES AES CM 128 HMAC SHA1 32	☑
CIPHER SUITES ARIA CM 128 HMAC SHA1 80	☑
CIPHER SUITES ARIA CM 192 HMAC SHA1 80	☑

2. Set the 'Media Security' parameter to **Enable** to enable SRTP.
3. Configure the other SRTP parameters as required.
4. Click **Submit**.
5. To save the changes to flash memory, see "Saving Configuration" on page 295.

Reader's Notes

16 Services

This section describes configuration for various supported services.

16.1 SIP-based Media Recording

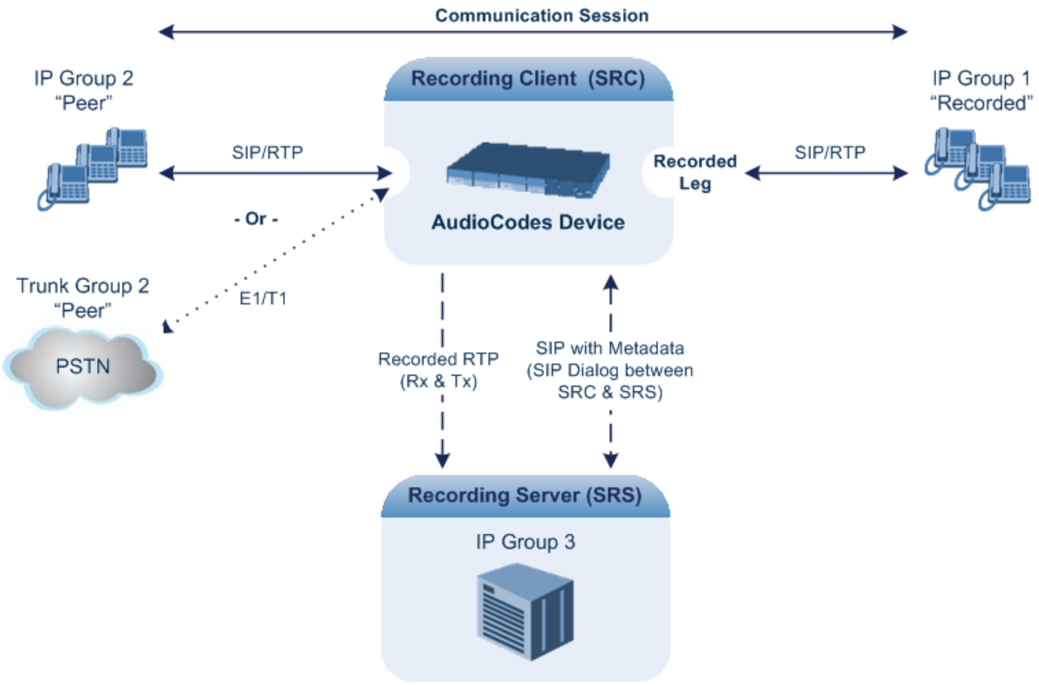
The device can record the SIP-based media (call sessions) traversing it. This support is in accordance with the Session Recording Protocol (siprec), which describes architectures for deploying session recording solutions and specifies requirements for extensions to SIP that will manage delivery of RTP media to a recording device. The siprec protocol is based on RFC 6341 (Use Cases and Requirements for SIP-Based Media Recording), Session Recording Protocol (draft-ietf-siprec-protocol-02), and Architecture (draft-ietf-siprec-architecture-03).

Notes:

- The SIP-based Media Recording feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 309. The Software License Key also specifies the maximum number of supported SIP recording sessions.
- For the maximum number of concurrent sessions that the device can record, contact your AudioCodes sales representative.

Session recording is a critical requirement in many business communications environments such as call centers and financial trading floors. In some of these environments, all calls must be recorded for regulatory and compliance reasons. In others, calls may be recorded for quality control or business analytics. Recording is typically performed by sending a copy of the session media to the recording devices.

The siprec protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) from the Session Recording Client (SRC), which is on the path of the Communication Session (CS), to a Session Recording Server (SRS) at the recording equipment. The device functions as the SRC, sending recording sessions to a third-party SRS, as shown in the figure below.



The device can record calls between two IP Groups. The type of calls to record can be specified by source and/or destination prefix number or SIP Request-URI, as well as by call initiator. The side ("leg") on which the recording is done must be specified. This recording leg must be one that is interfacing with one of the IP Groups. Specifying the leg is important as it determines the various call media attributes of the recorded RTP, for example, coder type.

The device initiates a recording session by sending an INVITE message to the SRS when the recorded call is connected. The SIP From header contains the identity of the SRC and the To header contains the identity of the SRS. The SDP in the INVITE contains:

- Two 'm=' lines that represent the two RTP streams (Rx and Tx).
- Two 'a=label:' lines that identify the streams.
- XML body (also referred to as metadata) that provides information on the participants of the call session:
  - <group id>: Logging Session ID (displayed as [SID:nnnnn] in Syslog), converted from decimal to hex. This number remains the same even if the call is forwarded or transferred. This is important for recorded calls.
  - <session id>: Originally recorded Call-ID, converted from decimal to hex.
  - <group-ref>: same as <group id>.
  - <participant id>: SIP From / To user.
  - <nameID aor>: From/To user@host.
  - <send> and <recv>: ID's for the RTP streams in hex - bits 0-31 are the same as group, bits 32-47 are the RTP port.
  - <stream id>: Same as <send> for each participant.
  - <label>: 1 and 2 (same as in the SDP's 'a=label:' line).

The SRS can respond with 'a=recvonly' for immediate recording or 'a=inactive' if recording is not yet needed, and send re-INVITE at any later time with the desired RTP mode change. If a re-INVITE is received in the original call (e.g. when a call is on hold), the device sends another re-INVITE with two 'm=' lines (but without metadata) to the SRS with the updated RTP data.

Below is an example of an INVITE sent by the device to an SRS:

```
INVITE sip:VSRP@1.9.64.253 SIP/2.0
Via: SIP/2.0/UDP 192.168.241.44:5060;branch=z9hG4bKac505782914
Max-Forwards: 10
From: <sip:192.168.241.44>;tag=1c505764207
To: <sip:VSRP@1.9.64.253>
Call-ID: 505763097241201011157@192.168.241.44
CSeq: 1 INVITE
Contact: <sip:192.168.241.44:5060>;src
Supported: replaces,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
Require: siprec
User-Agent: Mediant /v.6.80A.014
Content-Type: multipart/mixed;boundary=boundary ac1fffff85b
Content-Length: 1832
```

```
--boundary ac1fffff85b
Content-Type: application/sdp
v=0
o=AudiocodesGW 921244928 921244893 IN IP4 10.33.8.70
s=SBC-Call
c=IN IP4 10.33.8.70
t=0 0
m=audio 6020 RTP/AVP 8 96
c=IN IP4 10.33.8.70
a=ptime:20
a=sendonly
a=label:1
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
m=audio 6030 RTP/AVP 8 96
c=IN IP4 10.33.8.70
a=ptime:20
a=sendonly
a=label:2
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
--boundary ac1fffff85b
Content-Type: application/rs-metadata
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
  <datamode>complete</datamode>
  <group id="00000000-0000-0000-0000-00003a36c4e3">
    <associate-time>2010-01-24T01:11:57Z</associate-time>
  </group>
  <session id="0000-0000-0000-0000-00000000d0d71a52">
    <group-ref>00000000-0000-0000-0000-00003a36c4e3</group-ref>
    <start-time>2010-01-24T01:11:57Z</start-time>
     <ac:AvayaUCID
xmlns="urn:ietf:params:xml:ns:Avaya">FA080030C4E34B5B9E59</ac:AvayaUCID>
  </session>
  <participant id="1056" session="0000-0000-0000-0000-00000000d0d71a52">
    <nameID aor="1056@192.168.241.20"></nameID>
    <associate-time>2010-01-24T01:11:57Z</associate-time>
    <send>00000000-0000-0000-0000-1CF23A36C4E3</send>
    <recv>00000000-0000-0000-0000-BF583A36C4E3</recv>
  </participant>
     <participant id="182052092" session="0000-0000-0000-0000-00000000d0d71a52">
    <nameID aor="182052092@voicelab.local"></nameID>
    <associate-time>2010-01-24T01:11:57Z</associate-time>
    <recv>00000000-0000-0000-0000-1CF23A36C4E3</recv>
    <send>00000000-0000-0000-0000-BF583A36C4E3</send>
  </participant>
  <stream id="00000000-0000-0000-0000-1CF23A36C4E3" session="0000-0000-0000-0000-
00000000d0d71a52">
    <label>1</label>
  </stream>
  <stream id="00000000-0000-0000-0000-BF583A36C4E3" session="0000-0000-0000-0000-
00000000d0d71a52">
    <label>2</label>
  </stream>
</recording>
--boundary_ac1fffff85b–
```

16.1.1 Enabling SIP-based Media Recording

The procedure below describes how to enable the SIP-based media Recording feature. Once you have enabled this feature, your SIP Recording Routing rules (configured in "Configuring SIP Recording Routing Rules" on page 135) become active.

➢ To enable SIP-based media recording:

1. Open the SIP Recording page (**Configuration** tab > **VoIP** menu > **Services** > **SIP Recording**).
2. From the 'SIP Recording Application' drop-down list, select **Enable**.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

16.1.2 Configuring SIP Recording Routing Rules

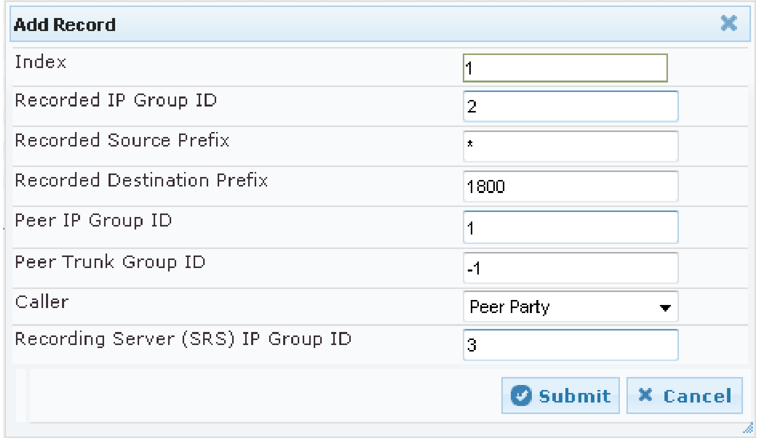
The SIP Recording Routing table lets you configure up to 30 SIP-based media recording rules. A SIP Recording Routing rule defines calls that you want to record. For an overview of this feature, see "SIP-based Media Recording" on page 132.

The procedure below describes how to configure SIP Recording Routing rules in the Web interface. You can also configure SIP Recording Routing rules using the table ini file parameter, SIPRecRouting or CLI command, configure voip/services sip-recording sip-rec-routing.

➢ To configure a SIP Recording Routing rule:

1. Open the SIP Recording page (**Configuration** tab > **VoIP** menu > **Services** > **SIP Recording**).
2. Click **Add**; the following dialog box appears:

Figure 59: SIP Recording Routing Table - Add Record



Add Record

Index	1
Recorded IP Group ID	2
Recorded Source Prefix	*
Recorded Destination Prefix	1800
Peer IP Group ID	1
Peer Trunk Group ID	-1
Caller	Peer Party
Recording Server (SRS) IP Group ID	3

The figure above shows a configuration example where the device records calls made by IP Group 1 to IP Group 2 that have the destination number prefix "1800". The device records the calls from the leg interfacing with IP Group 2, sending the recorded media to IP Group 3 (i.e., the SRS).

3. Configure a SIP recording route according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 45: SIP Recording Routing Parameter Descriptions

Parameter	Description
Index **[SIPRecRouting_Index]**	Defines an index number for the new table record.
Recorded IP Group ID CLI: recorded-ip-group-id **[SIPRecRouting_RecordedIPGroupID]**	Defines the IP Group participating in the call and the recording is done on the leg interfacing with this IP Group.
Recorded Source Prefix CLI: recorded-src-prefix **[SIPRecRouting_RecordedSourcePrefix]**	Defines calls to record based on source number or URI.
Recorded Destination Prefix CLI: recorded-dst-prefix **[SIPRecRouting_RecordedDestinationPrefix]**	Defines calls to record based on destination number or URI.
Peer IP Group ID CLI: peer-ip-group-id **[SIPRecRouting_PeerIPGroupID]**	Defines the peer IP Group that is participating in the call.
Caller CLI: caller **[SIPRecRouting_Caller]**	Defines which calls to record according to which party is the caller. ▪ **[0]** Both (default) = Caller can be peer or recorded side ▪ **[1]** Recorded Party ▪ **[2]** Peer Party
Recording Server (SRS) IP Group ID **[SIPRecRouting_SRSIPGroupID]**	Defines the IP Group of the recording server (SRS). **Note:** The SIP Interface used for communicating with the SRS is according to the SRD assigned to the SRS IP Group (in the IP Group table). If two SIP Interfaces are associated with the SRD - one for "SBC" and one for "GW &IP2IP" – the device uses the "SBC" SIP Interface. If no SBC SIP Interface type is configured, the device uses the "GW & IP2IP" interface.

16.1.3 Configuring SIP User Part for SRS

You can configure the SIP user part of the Request-URI for the recording server (SRS). The device inserts this user part in the SIP To header of the INVITE message sent to the SRS.

➢ To configure the SIP user part for SRS:

1. Open the SIP Recording page (**Configuration** tab > **VoIP** menu > **Services** > **SIP Recording**).
2. In the 'Recording Server (SRS) Destination Username' field, enter a user part value (string of up to 50 characters).
3. Click **Submit**, and then save ("burn") your settings to flash memory.

16.1.4 Interworking SIP-based Media Recording with Third-Party Vendors

The device can interwork the SIP-based Media Recording feature with third-party vendors, as described in the following subsections.

16.1.4.1 Genesys

The device's SIP-based media recording can interwork with Genesys' equipment. Genesys sends its proprietary X-Genesys-CallUUID header (which identifies the session) in the first SIP message, typically in the INVITE and the first 18x response. If the device receives a SIP message with Genesys SIP header, it adds the header's information to AudioCodes' proprietary tag in the XML metadata of the SIP INVITE that it sends to the recording server, as shown below:

```
<ac:GenesysUUID
xmlns="urn:ietf:params:xml:ns:Genesys">4BOKLLA3VH66JF112M1CC9VHKS14F0KP</ac:GenesysU
UID>
```

16.1.4.2 Avaya UCID

The device's SIP-based media recording can interwork with Avaya equipment. The Universal Call Identifier (UCID) is Avaya's proprietary call identifier used to correlate call records between different systems and identifies sessions. Avaya generates this in outgoing calls. If the device receives a SIP INVITE from Avaya, it adds the UCID value, received in the User-to-User SIP header to AudioCodes' proprietary tag in the XML metadata of the SIP INVITE that it sends to the recording server. For example, if the received SIP header is:

```
User-to-User: 00FA080019001038F725B3;encoding=hex
```

the device includes the following in the XML metadata:

```
xml metadata:
<ac:AvayaUCID xmlns="urn:ietf:params:xml:ns:Avaya">
FA080019001038F725B3</ac:AvayaUCID>
```

Note: For calls sent from the device to Avaya, the device can generate the Avaya UCID, if required. To configure this support, use the following parameters:

- 'UUI Format' in the IP Group table - enables Avaya support.
- 'Network Node ID' - defines the Network Node Identifier of the <device >for Avaya UCID.

16.2 Routing Based on LDAP Active Directory Queries

The device supports the Lightweight Directory Access Protocol (LDAP) application protocol and operating with third-party, LDAP-compliant servers such as Microsoft Active Directory. LDAP can be used for routing decisions based on retrieved LDAP queried information, for example, calling name and destination address.

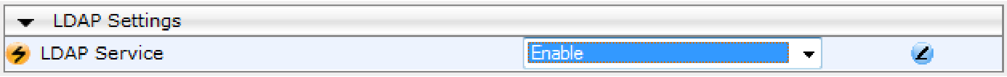
16.2.1 Enabling the LDAP Service

Before you can configure LDAP, you need to enable the LDAP service.

➢ To enable LDAP:

1. Open the LDAP Settings page (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Settings**).

Figure 60: Enabling LDAP on the LDAP Settings Page



LDAP Settings

LDAP Service: Enable

2. Under LDAP Settings, from the 'LDAP Service' drop-down list, select **Enable**.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

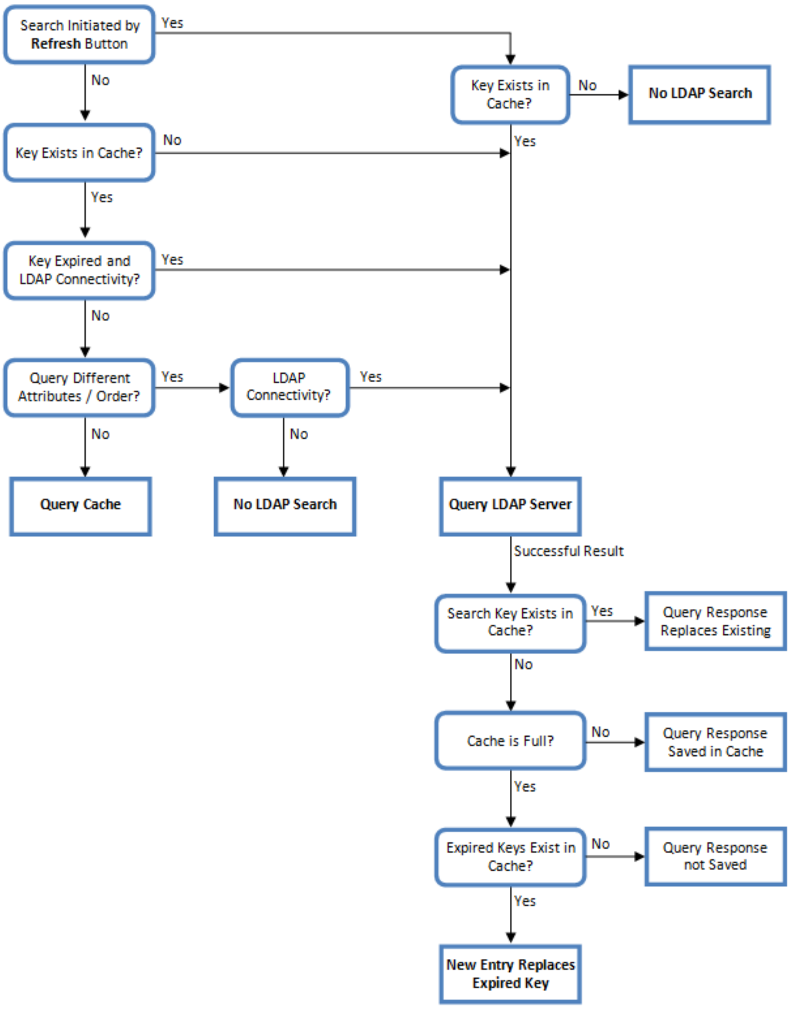
16.2.2 Configuring the Device's LDAP Cache

The device provides an option for storing recent LDAP queries and responses with an LDAP server in its local cache. The cache is used for subsequent queries and/or in case of LDAP server failure. The advantage of enabling this feature includes the following:

- Improves routing decision performance by using local cache for subsequent LDAP queries
- Reduces number of queries performed on an LDAP server and corresponding bandwidth consumption
- Provides partial survivability in case of intermittent LDAP server failure (or network isolation)

The handling of LDAP queries with the LDAP cache is shown in the flowchart below:

Figure 61: LDAP Query Process with Local LDAP Cache


Search Initiated by Refresh Button
Yes
No
Key Exists in Cache?
No
No LDAP Search
Key Exists in Cache?
No
Yes
Yes
Key Expired and LDAP Connectivity?
Yes
No
Query Different Attributes / Order?
Yes
LDAP Connectivity?
Yes
No
No
Query Cache
No LDAP Search
Query LDAP Server
Successful Result
Search Key Exists in Cache?
Yes
Query Response Replaces Existing
No
Cache is Full?
No
Query Response Saved in Cache
Yes
Expired Keys Exist in Cache?
No
Query Response not Saved
Yes
New Entry Replaces Expired Key

Note: If on the first LDAP query, the result fails for at least one attribute and is successful for at least one, the partial result is cached. However, for subsequent queries, the device does not use the partially cached result, but does a new query with the LDAP server again.

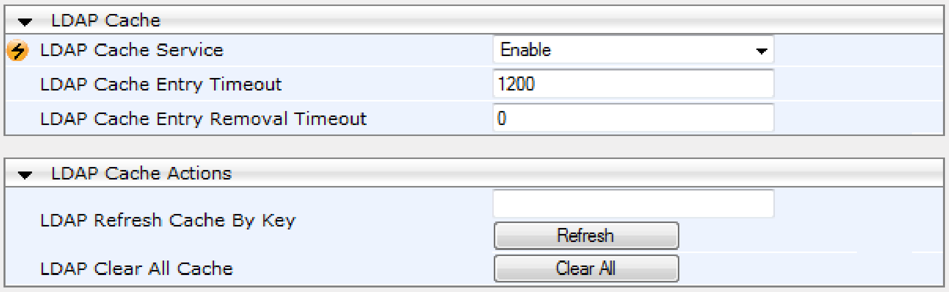
The procedure below describes how to configure the device's LDAP cache in the Web interface. For a full description of the cache parameters, see "LDAP Parameters" on page 454.

➢ To configure the LDAP cache:

1. Open the LDAP Settings page (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP**

Settings).

Figure 62: LDAP Settings Page - Cache Parameters



LDAP Cache

LDAP Cache Service	Enable
LDAP Cache Entry Timeout	1200
LDAP Cache Entry Removal Timeout	0

LDAP Cache Actions

LDAP Refresh Cache By Key	Refresh
LDAP Clear All Cache	Clear All

2. From the 'LDAP Cache Service' drop-down list, select **Enable** to enable LDAP cache.
3. In the 'LDAP Cache Entry Timeout' field, enter the duration (in minutes) that an entry in the LDAP cache is valid.
4. In the 'LDAP Cache Entry Removal Timeout' field, enter the duration (in hours) after which the device removes the LDAP entry from the cache.
5. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

The LDAP Settings page also provides you with the following buttons:

- **LDAP Refresh Cache By Key:** Refreshes a saved LDAP entry response in the cache of a specified LDAP search key. If a request with the specified key exists in the cache, the request is resent to the LDAP server.
- **LDAP Clear All Cache:** Removes all LDAP entries in the cache.

16.2.3 Configuring LDAP Servers

The device can operate with up to two LDAP servers. The basic LDAP mechanism is described below:

- **LDAP Connection:** The address and connectivity settings of the LDAP server are configured in the LDAP Configuration table. The device connects and binds to the remote LDAP server (IP address or DNS/FQDN) either during the service's initialization (at device start-up) or whenever the LDAP server's IP address and port is changed. Service makes 10 attempts to connect and bind to the remote LDAP server with a timeout of 20 seconds between attempts. Binding to the LDAP server can be anonymous or not. For anonymous binding, do not configure the the 'LDAP Bind DN' and 'LDAP Password' parameters. If connection fails, the service remains in disconnected state until either the LDAP server's IP address or port is changed. If connection to the LDAP server later fails, the service attempts to reconnect, as described previously. The SNMP alarm, acLDAPLostConnection is sent when connection is broken. Upon successful reconnection, the alarm is cleared.

Note: If you configure an FQDN, make sure that the 'LDAP Server IP' parameter is left empty.

- **Search Queries:** For the device to run a search using the LDAP service, the path to the directory's subtree (distinguished name / DN) where the search is to be done must be configured. This is done in the Ldap Servers Search Dns table. The device can query up to three DN objects per LDAP server. The search key, or *filter* in LDAP references, which defines the exact DN to be found and one or more attributes whose values should be returned, must also be defined. If connection to the LDAP server is disrupted during the search, all search requests are dropped and an alarm indicating a failed status is sent to client applications.

When two LDAP servers are implemented, the device runs an LDAP query to search for DN object records on both LDAP servers. The method of how the device queries the DN object record between the two LDAP servers can be configured to one of the following:

- **Parallel Search:** The device queries the LDAP servers at the same time.
- **Sequential Search:** The device first queries one of the LDAP servers, and if the DN object is not found, it queries the second LDAP server.

The method for searching DN objects within each LDAP server can be configured to one of the following:

- **Parallel Search:** The device queries in all DN objects at the same time. For example, a search for the DN object record, "JohnD" is done at the same time in the "Marketing", "Sales" and "Administration" DN objects.
- **Sequential Search:** The device queries in each DN object, one by one, until a result is found. For example, a search for the DN object record, "JohnD" is first run in DN object "Marketing" and if a result is not found, it searches in "Sales", and if not found, it then searches in "Administration", and so on.

To configure the above search methods, use the following parameters:

- LDAPSearchServerMethod: Configures the method for querying between two LDAP severs.
- LdapSearchDnsInParallel: Configures the method for querying the DN within each LDAP server.

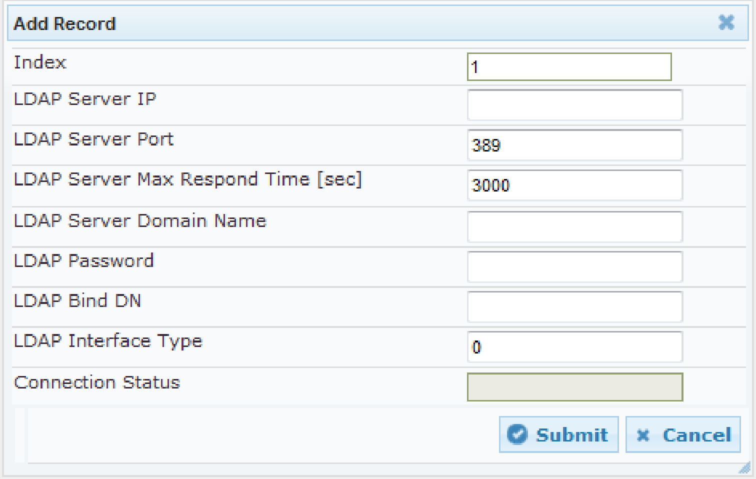
The procedure below describes how to configure an LDAP server with base path DNs in the Web interface. You can also configure this using other management tools:

- **LDAP server (LDAP Configuration table):** table ini file parameter, LdapConfiguration or CLI command, configure voip/ldap/ldap-configuration
- **DN base paths per LDAP server (Ldap Servers Search Dns Table):** table ini file parameter, LdapServersSearchDNs or CLI command, configure voip/ldap/ldap-servers-search-dns

➢ To configure an LDAP server:

1. Open the LDAP Configuration Table page (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Configuration Table**).
2. Click **Add**; the following dialog box appears:

Figure 63: LDAP Settings Page



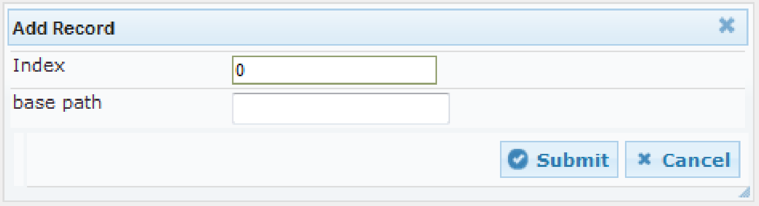
Add Record

Index	1
LDAP Server IP	
LDAP Server Port	389
LDAP Server Max Respond Time [sec]	3000
LDAP Server Domain Name	
LDAP Password	
LDAP Bind DN	
LDAP Interface Type	0
Connection Status	

3. Configure an LDAP server according to the parameters described in the table below.
4. Click **Submit**.
5. Configure LDAP base paths per LDAP server:
   a. In the LDAP Configuration table, select the row of the LDAP server for which you want to configure DN base paths.
   b. Click the **Ldap Servers Search Dns** link, located at the bottom of the page; the Ldap Servers Search Dns Table page opens.

c. Click **Add**; the following dialog box appears:

Figure 64: Ldap Servers Search Dns Table - Add Record



Add Record

Index: 0

base path:

d. Configure an LDAP DN base path according to the parameters described at the end of the table below.

e. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 46: LDAP Configuration Table Parameter Descriptions

Parameter	Description
Index **[LdapConfiguration_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
LDAP Server IP CLI: server-ip **[LdapConfiguration_LdapConfServerIp]**	Defines the IP address of the LDAP server (in dotted-decimal notation, e.g., 192.10.1.255). By default, no IP address is defined.
LDAP Server Port CLI: server-port **[LdapConfiguration_LdapConfServerPort]**	Defines the port number of the LDAP server. The valid value range is 0 to 65535. The default port number is 389.
LDAP Server Max Respond Time CLI: max-respond-time **[LdapConfiguration_LdapConfServerMaxRespondTime]**	Defines the time (in seconds) that the device waits for LDAP server responses. The valid value range is 0 to 86400. The default is 3000.
LDAP Server Domain Name CLI: domain-name **[LdapConfiguration_LdapConfServerDomainName]**	Defines the domain name (FQDN) of the LDAP server. The device tries to connect to the LDAP server according to the IP address listed in the received DNS query. If there is no connection to the LDAP server or the connection to the LDAP server fails, the device tries to connect to the LDAP server with the next IP address in the DNS query list. **Note:** The 'LDAP Server IP' parameter takes precedence over this parameter. Thus, if you want to use an FQDN, keep the 'LDAP Server IP' parameter empty.
LDAP Password CLI: password **[LdapConfiguration_LdapConfPassword]**	Defines the user password for the LDAP server.
LDAP Bind DN CLI: bind-dn **[LdapConfiguration_LdapConfBindDn]**	Defines the LDAP server's bind Distinguished Name (DN). This is used as the username during connection and binding to the server. For example: "CN=Search user,OU=Labs,DC=OCSR2,DC=local" **Note:** The DN is used to uniquely name an AD object.
LDAP Interface Type CLI: interface-type **[LdapConfiguration_LdapConfInterfaceType]**	Assigns a network interface to the LDAP.

Parameter	Description
Connection Status CLI: connection-status **[LdapConfiguration_ConnectionStatus]**	(Read-only) Displays the connection status with the LDAP server. ▪ "Not Applicable" ▪ "LDAP Connection Broken" ▪ "Connecting" ▪ "Connected"
Ldap Servers Search Dns Table ("child" table of the LDAP Configuration table, defining DN base paths per LDAP server)	
Index **[LdapServersSearchDNs_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
base path **[LdapServersSearchDNs_Base_Path]**	Defines the full path (DN) to the objects in the AD where the query is done. You can configure up to three DNs per LDAP server. For example: OU=NY,DC=OCSR2,DC=local In this example, the DN path is defined by the LDAP names, OU (organizational unit) and DC (domain component).

16.3 Least Cost Routing

This section provides a description of the device's least cost routing (LCR) feature and how to configure it.

16.3.1 Overview

The LCR feature enables the device to choose the outbound IP destination routing rule based on lowest call cost. This is useful in that it enables service providers to optimize routing costs for customers. For example, you may wish to define different call costs for local and international calls, or different call costs for weekends and weekdays (specifying even the time of call). The device sends the calculated cost of the call to a Syslog server (as Information messages), thereby enabling billing by third-party vendors.

LCR is implemented by defining Cost Groups and assigning them to routing rules in the IP-to-IP Routing table. The device searches this routing table for matching routing rules, and then selects the rule with the lowest call cost. If two routing rules have identical costs, then the rule appearing higher up in the table is used (i.e., first-matched rule). If a selected route is unavailable, the device selects the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched rules with Cost Groups. This is determined according to the settings of the Default Cost parameter in the Routing Rule Groups table.

The Cost Group defines a fixed connection cost (*connection cost*) and a charge per minute (*minute cost*). Cost Groups can also be configured with time segments (*time bands*), which define connection cost and minute cost based on specific days of the week and time of day (e.g., from Saturday through Sunday, between 6:00 and 18:00). If multiple time bands are configured per Cost Group and a call spans multiple time bands, the call cost is calculated using only the time band in which the call was initially established.

In addition to Cost Groups, the device can calculate the call cost using an optional, user-defined average call duration value. The logic in using this option is that a Cost Group may be cheap if the call duration is short, but due to its high minute cost, may prove very expensive if the duration is lengthy. Thus, together with Cost Groups, the device can use this option to determine least cost routing. The device calculates the Cost Group call cost as follows: Total Call Cost = Connection Cost + (Minute Cost * Average Call Duration).

The below table shows an example of call cost when taking into consideration call duration. This example shows four defined Cost Groups and the total call cost if the average call duration is 10 minutes:

Table 47: Call Cost Comparison between Cost Groups for different Call Durations

Cost Group	Connection Cost	Minute Cost	Total Call Cost per Duration

			1 Minute	10 Minutes
A	1	6	7	61
B	0	10	10	100
C	0.3	8	8.3	80.3
D	6	1	7	**16**

If four matching routing rules are located in the routing table and each one is assigned a different Cost Group as listed in the table above, then the rule assigned Cost Group "D" is selected. Note that for one minute, Cost Groups "A" and "D" are identical, but due to the average call duration, Cost Group "D" is cheaper. Therefore, average call duration is an important factor in determining the cheapest routing role.

Below are a few examples of how you can implement LCR:

- **Example 1:** This example uses two different Cost Groups for routing local calls and international calls:

  Two Cost Groups are configured as shown below:

Cost Group	Connection Cost	Minute Cost
1. "Local Calls"	2	1
2. "International Calls"	6	3

The Cost Groups are assigned to routing rules for local and international calls:

Routing Index	Dest Phone Prefix	Destination IP	Cost Group ID
1	2000	x.x.x.x	1 "Local Calls"
2	00	x.x.x.x	2 "International Calls"

- **Example 2:** This example shows how the device determines the cheapest routing rule in the Outbound IP Routing table:

  The Default Cost parameter (global) in the Routing Rule Groups table is set to **Min**, meaning that if the device locates other matching LCR routing rules (with Cost Groups assigned), the routing rule without a Cost Group is considered the lowest cost route.

  - The following Cost Groups are configured:

Cost Group	Connection Cost	Minute Cost
1. "A"	2	1
2. "B"	6	3

- The Cost Groups are assigned to routing rules:

Routing Index	Dest Phone Prefix	Destination IP	Cost Group
1	201	x.x.x.x	"A'
2	201	x.x.x.x	"B"
3	201	x.x.x.x	0
4	201	x.x.x.x	"B"

The device calculates the optimal route in the following index order: 3, 1, 2, and then 4, due to the following logic:

- Index 1 - Cost Group "A" has the lowest connection cost and minute cost

- Index 2 - Cost Group "B" takes precedence over Index 4 entry based on the first-matched method rule
- Index 3 - no Cost Group is assigned, but as the Default Cost parameter is set to **Min**, it is selected as the cheapest route
- Index 4 - Cost Group "B" is only second-matched rule (Index 1 is the first)

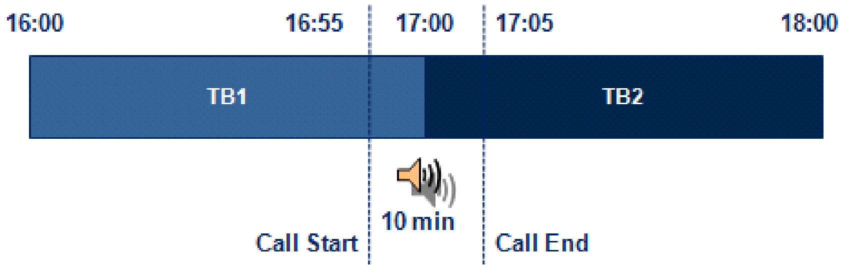
- **Example 3:** This example shows how the cost of a call is calculated if the call spans over multiple time bands:

Assume a Cost Group, "CG Local" is configured with two time bands, as shown below:

Cost Group	Time Band	Start Time	End Time	Connection Cost	Minute Cost
CG Local	TB1	16:00	17:00	2	1
	TB2	17:00	18:00	7	2

Assume that the call duration is 10 minutes, occurring between 16:55 and 17:05. In other words, the first 5 minutes occurs in time band "TB1" and the next 5 minutes occurs in "TB2", as shown below:

Figure 65: LCR using Multiple Time Bands (Example)



The device calculates the call using the time band in which the call was initially established, regardless of whether the call spans over additional time bands:

Total call cost = "TB1" Connection Cost + ("TB1" Minute Cost x call duration) = 2 + 1 x 10 min = 12

16.3.2 Configuring LCR

The following main steps need to be done to configure LCR:

1. Enable the LCR feature and configure the average call duration and default call connection cost - see "Enabling LCR and Configuring Default LCR" on page 144.
2. Configure Cost Groups - see "Configuring Cost Groups" on page 146.
3. Configure Time Bands for a Cost Group - see "Configuring Time Bands for Cost Groups" on page 146.
4. Assign Cost Groups to outbound IP routing rules - see "Assigning Cost Groups to Routing Rules" on page 148.

16.3.2.1 Enabling the LCR Feature

The Routing Rule Groups table lets you enable the LCR feature. This also includes configuring the average call duration and default call cost for routing rules that are not assigned Cost Groups in the Outbound IP Routing table.

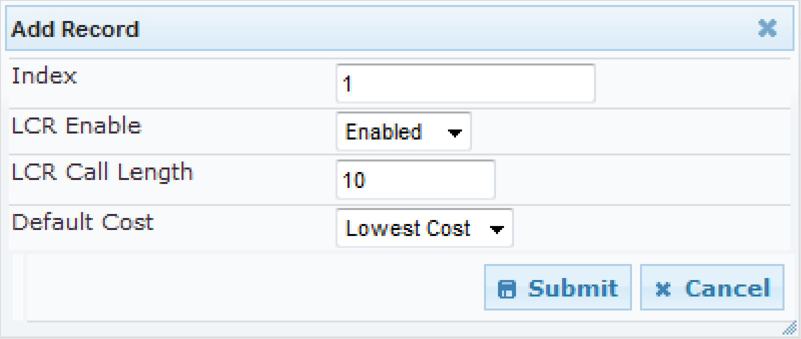
The procedure below describes how to enable LCR in the Web interface. You can also do this using the table ini file parameter, RoutingRuleGroups or CLI command, configure voip > services least-cost-routing routing-rule-groups.

➢ To enable LCR:

1. Open the Routing Rule Groups Table page (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Routing Rule Groups Table**).

2. Click **Add**; the following dialog box appears:

Figure 66: Routing Rule Groups Table - Add Record



Add Record

Index	1
LCR Enable	Enabled
LCR Call Length	10
Default Cost	Lowest Cost

3. Enable LCR according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 48: Routing Rule Groups Table Parameter Descriptions

Parameter	Description
Index **[RoutingRuleGroups_Index]**	Defines an index number for the new table record. **Note:** Only one index entry can be configured.
LCR Enable CLI: lcr-enable **[RoutingRuleGroups_LCREnable]**	Enables the LCR feature: ▪ **[0]** Disabled (default) ▪ **[1]** Enabled
LCR Call Length CLI: lcr-call-length **[RoutingRuleGroups_LCRAverageCallLength]**	Defines the average call duration (in minutes) and is used to calculate the variable portion of the call cost. This is useful, for example, when the average call duration spans over multiple time bands. The LCR is calculated as follows: cost = call connect cost + (minute cost * average call duration) The valid value range is 0-65533. The default is 1. For example, assume the following Cost Groups: ▪ "Weekend A": call connection cost is 1 and charge per minute is 6. Therefore, a call of 1 minute cost 7 units. ▪ "Weekend_ B": call connection cost is 6 and charge per minute is 1. Therefore, a call of 1 minute cost 7 units. Therefore, for calls under one minute, "Weekend A" carries the lower cost. However, if the average call duration is more than one minute, then "Weekend B" carries the lower cost.
Default Cost CLI: lcr-default-cost **[RoutingRuleGroups_LCRDefaultCost]**	Determines whether routing rules in the Outbound IP Routing table without an assigned Cost Group are considered a higher cost or lower cost route compared to other matched routing rules that are assigned Cost Groups. ▪ **[0]** Lowest Cost = If the device locates other matching LCR routing rules, this routing rule is considered the lowest cost route and therefore, it is selected as the route to use (default.) ▪ **[1]** Highest Cost = If the device locates other matching LCR routing rules, this routing rule is considered as the highest cost route and therefore, is not used or used only if the other cheaper routes are unavailable. **Note:** If more than one valid routing rule without a defined Cost Group exists, the device selects the first-matched rule.

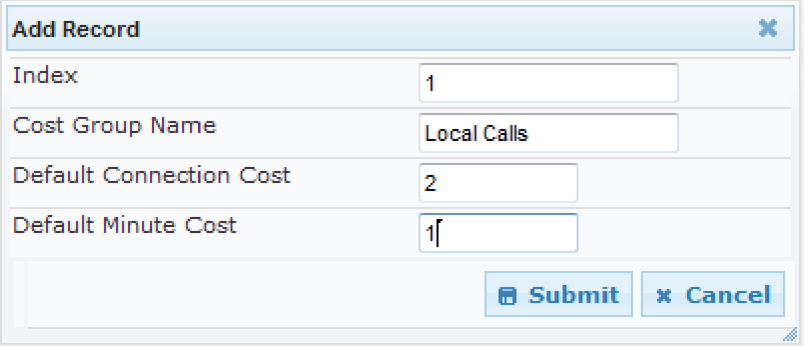
16.3.2.2 Configuring Cost Groups

The Cost Group table lets you configure Cost Groups. A Cost Group defines a fixed call connection cost and a call rate (charge per minute). Once configured, you can configure Time Bands per Cost Group. Up to 10 Cost Groups can be configured.

The procedure below describes how to configure Cost Groups in the Web interface. You can also configure this using the table ini file parameter, CostGroupTable or CLI command, configure voip > services least-cost-routing cost-group.

➢ To configure a Cost Group:

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Cost Group Table**).
2. Click **Add**; the following dialog box appears:



Add Record

Index: 1

Cost Group Name: Local Calls

Default Connection Cost: 2

Default Minute Cost: 1

3. Configure a Cost Group according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 49: Cost Group Table Parameter Descriptions

Parameter	Description
Index **[CostGroupTable_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Cost Group Name CLI: cost-group-name **[CostGroupTable_CostGroupName]**	Defines an arbitrary name for the Cost Group. The valid value is a string of up to 30 characters. **Note:** Each Cost Group must have a unique name.
Default Connection Cost CLI:default-connection-cost **[CostGroupTable_DefaultConnectionCost]**	Defines the call connection cost (added as a fixed charge to the call) for a call outside the time bands. The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default connection cost is used.
Default Minute Cost CLI: default-minute-cost **[CostGroupTable_DefaultMinuteCost]**	Defines the call charge per minute for a call outside the time bands. The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default charge per minute is used.

16.3.2.3 Configuring Time Bands for Cost Groups

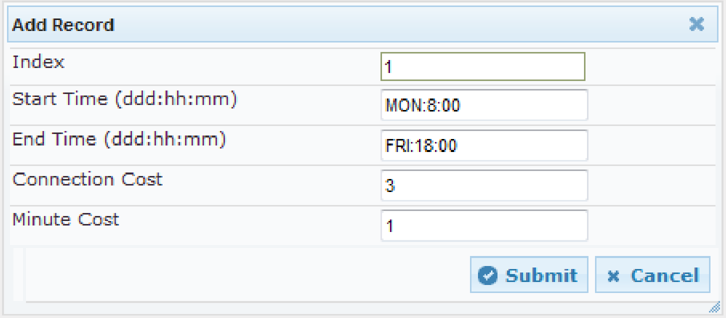
The Time Band table lets you configure Time Bands per Cost Group. A Time Band defines a day and time range (e.g., from Saturday 05:00 to Sunday 24:00), as well as the fixed call connection charge and call rate per minute for this interval. You can configure up to 70 Time Bands, where up to 21 Time Bands can be assigned to each Cost Group.

Note: You cannot configure overlapping Time Bands.

The procedure below describes how to configure Time Bands per Cost Group in the Web interface. You can also configure this using the table ini file parameter, CostGroupTimebands or CLI command, configure voip >services least-cost-routing cost-group-time-bands.

➢ To configure a Time Band per Cost Group:

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Cost Group Table**).
2. Select a Cost Group for which you want to assign Time Bands, and then click the **Time Band** link located below the table; the Time Band table for the selected Cost Group appears.
3. Click **Add**; the following dialog box appears:



Add Record

Index	1
Start Time (ddd:hh:mm)	MON:8:00
End Time (ddd:hh:mm)	FRI:18:00
Connection Cost	3
Minute Cost	1

4. Configure a Time Band according to the parameters described in the table below.
5. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 50: Time Band Table Description

Parameter	Description
Index CLI: timeband-index **[CostGroupTimebands_TimebandIndex]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Start Time CLI: start-time **[CostGroupTimebands_StartTime]**	Defines the day and time of day from when this time band is applicable. The format is DDD:hh:mm, where: ▪ *DDD* is the day of the week, represented by the first three letters of the day in upper case (i.e., SUN, MON, TUE, WED, THU, FRI, or SAT). ▪ *hh* and *mm* denote the time of day, where *hh* is the hour (00-23) and *mm* the minutes (00-59) For example, SAT:22:00 denotes Saturday at 10 pm.
End Time CLI: end-time **[CostGroupTimebands_EndTime]**	Defines the day and time of day until when this time band is applicable. For a description of the valid values, see the parameter above.
Connection Cost CLI: connection-cost **[CostGroupTimebands_ConnectionCost]**	Defines the call connection cost during this time band. This is added as a fixed charge to the call. The valid value range is 0-65533. The default is 0. **Note:** The entered value must be a whole number (i.e., not a decimal).

Parameter	Description
Minute Cost CLI: minute-cost **[CostGroupTimebands_MinuteCost]**	Defines the call cost per minute charge during this timeband. The valid value range is 0-65533. The default is 0. **Note:** The entered value must be a whole number (i.e., not a decimal).

16.3.2.4 Assigning Cost Groups to Routing Rules

To use your configured Cost Groups, you need to assign them to routing rules:

- IP-to-IP Routing table - see Configuring SBC IP-to-IP Routing Rules on page 255

16.4 Configuring Call Setup Rules

The Call Setup Rules table lets you configure up to 40 Call Setup rules. Call Setup rules define various sequences that are run upon the receipt of an incoming call (dialog) at call setup, before the device routes the call to its destination. Call Setup rules provides you with full flexibility in implementing simple or complex script-like rules that can be used for Lightweight Directory Access Protocol (LDAP) based routing as well as other advanced routing logic requirements such as manipulation. These Call Setup rules are assigned to routing rules.

Below is a summary of functions for which you can employ Call Setup rules:

- LDAP query rules: LDAP is used by the device to query Microsoft's Active Directory (AD) server for specific user details for routing, for example, office extension number, mobile number, private number, OCS (Lync) address, and display name. Call Setup rules provides full flexibility in AD-lookup configuration to suite just about any customer deployment requirement:
    - Routing based on query results.
    - Queries based on any AD attribute.
    - Queries based on any attribute value (alphanumeric), including the use of the asterisk (*) wildcard as well as the source number, destination number, redirect number, and SBC SIP messages. For example, the following Call Setup rule queries the attribute "proxyAddresses" for the record value "WOW:" followed by source number: "proxyAddresses=WOW:12345*"
    - Conditional LDAP queries, for example, where the query is based on two attributes (&(telephoneNumber=4064)(company=ABC).
    - Conditions for checking LDAP query results.
    - Manipulation of call parameters such as source number, destination number, and redirect number and SBC SIP messages, while using LDAP query results.
    - Multiple LDAP queries.
- Manipulation (similar to the Message Manipulations table) of call parameters (such as source number, destination number, and redirect number) and SBC SIP messages.
- Conditions for routing, for example, if the source number equals a specific value, then use the call routing rule.

You configure Call Setup rules with a Set ID, similar to the Message Manipulations table, where multiple rules can be associated with the same Set ID. This lets you perform multiple Call Setup rules on the same call setup dialog.

To use your Call Setup rule(s), you need to assign the Call Setup Rules Set ID to the relevant routing rule. This is done using the 'Call Setup Rules Set ID' field in the routing table:

- SBC IP-to-IP routing - see Configuring SBC IP-to-IP Routing Rules on page 255

If an incoming call matches the characteristics of a routing rule, the device **first** runs the assigned Call Setup Rules Set ID. The device uses the routing rule to route the call, depending on the result of the Call Setup Rules Set ID:

- **Rule's condition is met:** The device performs the rule's action and then runs the next rule in the Set ID until the last rule or until a rule with an **Exit** Action Type. If the **Exit** rule is configured with a "True" Action Value, the device uses the current routing rule. If the **Exit** rule is configured with a "False" Action Value, the device moves to the next routing rule. If an **Exit** Action Type is not configured and the device has run all the rules in the Set ID, the default Action Value of the Set ID is "True" (i.e., use the current routing rule).

- **Rule's condition is not met:** The device runs the next rule in the Set ID. When the device reaches the end of the Set ID and no **Exit** was performed, the Set ID ends with a "True" result.

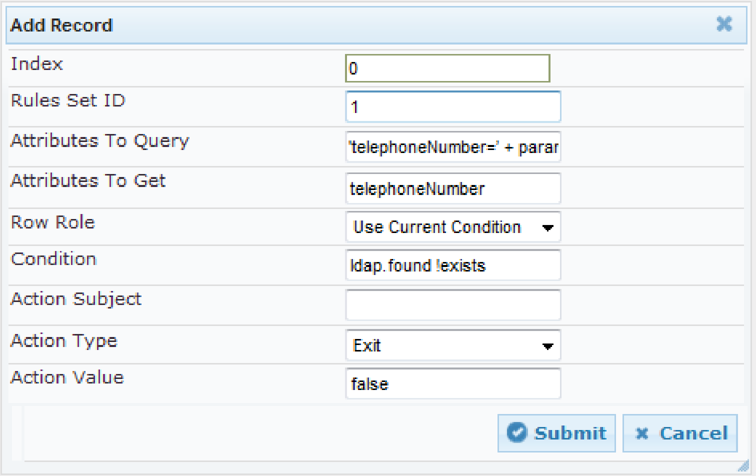
Note: If the source and/or destination numbers are manipulated by the Call Setup rules, they revert to their original values if the device moves to the next routing rule.

The procedure below describes how to configure Call Setup Rules in the Web interface. You can also configure Call Setup Rules using the table ini file parameter, CallSetupRules or CLI command, configure voip/services call-setup-rules.

➢ To configure a Call Setup rule:

1. Open the Call Setup Rules table (**Configuration** tab > **VoIP** menu > **Services** > **Call Setup Rules**).
2. Click **Add**; the following dialog box appears:

Figure 67: Call Setup Rules Table - Add Record



Add Record

Index	0
Rules Set ID	1
Attributes To Query	'telephoneNumber=' + parar
Attributes To Get	telephoneNumber
Row Role	Use Current Condition
Condition	ldap.found !exists
Action Subject	
Action Type	Exit
Action Value	false

3. Configure a Call Setup rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 51: Call Setup Rules Parameter Descriptions

Parameter	Description
Index **[CallSetupRules_Index]**	Defines an index number for the new table record. **Note:** Each rule must be configured with a unique index.
Rules Set ID CLI: rules-set-id **[CallSetupRules_RulesSetID]**	Defines a Set ID for the rule. You can define the same Set ID for multiple rules to create a group of rules. You can configure up to 10 Set IDs, where each Set ID can include up to 10 rules.The Set ID is used to assign the Call Setup rules to a routing rule in the routing table. The valid value is 0 to 9. The default is 0.
Attributes To Query CLI: attr-to-query **[CallSetupRules_AttributesToQuery]**	Defines the query string that the device sends to the LDAP server. The valid value is a string of up to 100 characters. Combined strings and values can be configured like in the Message Manipulations table, using

Parameter	Description
	the '+' operator. Single quotes (') can be used for specifying a constant string (e.g., '12345'). For example: ▪ 'mobile=' + param.call.dst.user (searches for the AD attribute, "mobile" that has the value of the destination user part of the incoming call) ▪ 'telephoneNumber=' + param.call.redirect + '*' (searches for the AD attribute, "telephoneNumber" that has a redirect number)
Attributes To Get CLI: attr-to-get **[CallSetupRules_AttributesToGet]**	Defines the attributes of the queried LDAP record that the device must handle (e.g., retrieve value). The valid value is a string of up to 100 characters. Up to five attributes can be defined, each separated by a comma (e.g., msRTCSIP-PrivateLine,msRTCSIP-Line,mobile). **Note:** The device saves the retrieved attributes' values for future use in other rules, until the next LDAP query or until the call is connected. Thus, the device does not need to re-query the same attributes.
Row Role CLI: row-role **[CallSetupRules_RowRole]**	Determines which condition must be met in order for this rule to be performed. ▪ **[0]** Use Current Condition = The Condition configured for this rule must be matched in order to perform the configured action (default). ▪ **[1]** Use Previous Condition = The Condition configured for the rule located directly above this rule in the Call Setup table must be matched in order to perform the configured action. This option lets you configure multiple actions for the same Condition.
Condition CLI: condition **[CallSetupRules_Condition]**	Defines the condition that must exist for the device to perform the action. The valid value is a string of up to 200 characters (case-insensitive). Regular Expression (regex) can also be used, for example: ▪ ldap.attr.mobile exists (attribute "mobile" exists in AD) ▪ param.call.dst.user == ldap.attr.msRTCSIP-PrivateLine (called number is the same as the number in the attribute "msRTCSIP-PrivateLine") ▪ ldap.found !exists (LDAP record not found) ▪ ldap.err exists (LDAP error exists)
Action Subject CLI: action-subject **[CallSetupRules_ActionSubject]**	Defines the element (header, parameter, or body) upon which you want to perform the action. The valid value is a string of up to 100 characters (case-insensitive). Examples: ▪ header.from contains '1234' ▪ param.call.dst.user (called number) ▪ param.call.src.user (calling number) ▪ param.call.src.name (calling name) ▪ param.call.redirect (redirect number) ▪ param.call.src.host (source host) ▪ param.call.dst.host (destination host)
Action Type CLI: action-type **[CallSetupRules_ActionType]**	Defines the type of action to perform. ▪ **[0]** Add (default) = Adds new message header, parameter or body elements. ▪ **[1]** Remove = Removes message header, parameter, or body elements. ▪ **[2]** Modify = Sets element to the new value (all element types). ▪ **[3]** Add Prefix = Adds value at the beginning of the string (string element only). ▪ **[4]** Add Suffix = Adds value at the end of the string (string element only).

Parameter	Description
	▪ **[5]** Remove Suffix = Removes value from the end of the string (string element only). ▪ **[6]** Remove Prefix = Removes value from the beginning of the string (string element only). ▪ **[20]** Exit = Stops the Rule Set ID and returns a result ("True" or "False"). ▪ **[21** Run Rules Set = Performs a different Rule Set ID, specified in the 'Action Value' parameter (below).
Action Value CLI: action-value **[CallSetupRules_ActionValue]**	Defines a value that you want to use in the action. The valid value is a string of up to 300 characters (case-insensitive). Examples: ▪ '+9723976'+ldap.attr.alternateNumber ▪ '9764000' ▪ ldap.attr.displayName ▪ true (if the 'Action Type' is set to **Exit**) ▪ false (if the 'Action Type' is set to **Exit**)

16.4.1 Call Setup Rule Examples

Below are configuration examples for using Call Setup Rules.

- **Example 1:** This example configures the device to replace (manipulate) the incoming call's source number with a number retrieved from the AD by an LDAP query. The device queries the AD server for the attribute record, "telephoneNumber" whose value is the same as the received source number (e.g., "telephoneNumber =4064"). If such an attribute is found, the device retrieves the number of the attribute record, "alternateNumber" and uses this number as the source number.
    - **Call Setup Rules table configuration:**
        - 'Rules Set ID': **1**
        - 'Attributes to Query': **'telephoneNumber=' + param.call.src.user**
        - 'Attributes to Get': **alternateNumber**
        - 'Row Role': **Use Current Condition**
        - 'Condition': **ldap.attr. alternateNumber exists**
        - 'Action Subject': **param.call.src.user**
        - 'Action Type': **Modify**
        - 'Action Value': **ldap.attr. alternateNumber**
    - **Routing table configuration:** A single routing rule is assigned the Call Setup Rule Set ID.
        - Index 1:
            - ✓ 'Call Setup Rules Set Id': **1**
- **Example 2:** This example configures the device to replace (manipulate) the incoming call's calling name (caller ID) with a name retrieved from the AD by an LDAP query. The device queries the AD server for the attribute record, "telephoneNumber" whose value is the same as the received source number (e.g., "telephoneNumber =5098"). If such an attribute is found, the device retrieves the name from the attribute record, "displayName" and uses this as the calling name in the incoming call.
    - **Call Setup Rules table configuration:**
        - 'Rules Set ID': **2**
        - 'Attributes to Query': **'telephoneNumber=' + param.call.src.user**
        - 'Attributes to Get': **displayName**

    - 'Row Role': **Use Current Condition**
    - 'Condition': **ldap.attr. displayName exists**
    - 'Action Subject': **param.call.src.name**
    - 'Action Type': **Modify**
    - 'Action Value': **ldap.attr. displayName**
- **Routing table configuration:** A single routing rule is assigned the Call Setup Rule Set ID.
    - Index 1:
        - ✓ 'Call Setup Rules Set Id': **2**

- **Example 3:** This example configures the device to route the incoming call according to whether or not the source number of the incoming call also exists in the AD server. The device queries the AD server for the attribute record, "telephoneNumber" whose value is the same as the received source number (e.g., telephoneNumber=4064"). If such an attribute is found, the device sends the call to the Lync server; if the query fails, the device sends the call to the PBX.
    - **Call Setup Rules table configuration:**
        - 'Rules Set ID': **3**
        - 'Attributes to Query': **'telephoneNumber=' + param.call.src.user**
        - 'Attributes to Get': **telephoneNumber**
        - 'Row Role': **Use Current Condition**
        - 'Condition': **ldap.found !exists**
        - 'Action Subject': -
        - 'Action Type': **Exit**
        - 'Action Value': **false**

      If the attribute record is found (i.e., condition is not met), the rule ends with a default exit result of true and uses the first routing rule (Lync). If the attribute record does not exist (i.e., condition is met), the rule exits with a false result and uses the second routing rule (PBX).
    - **Routing table configuration:** Two routing rules are assigned with the same matching characteristics. Only the main routing rule is assigned a Call Setup Rules Set ID.
        - Index 1:
            - ✓ 'Call Setup Rules Set Id': **3**
            - ✓ 'Destination IP Group ID': **3** (IP Group for Lync)
        - Index 2:
            - ✓ 'Destination IP Group ID': **4** (IP Group of PBX)

Reader's Notes

17 Quality of Experience

This chapter describes how to configure the Quality of Experience feature.

Note: The Quality of Experience feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 309.

17.1 Configuring the SEM Server

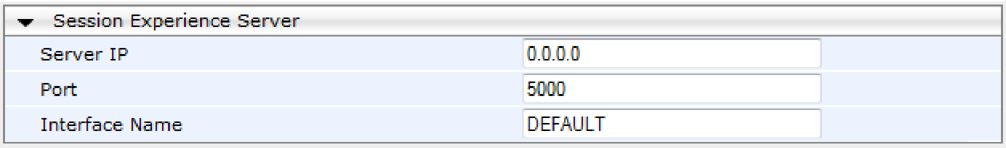
The device can be configured to report voice (media) Quality of Experience (QoE) to AudioCodes' Session Experience Manager (SEM) server, a plug-in for AudioCodes EMS. The reports include real-time metrics of the quality of the actual call experience, which are then processed by the SEM. The device connects as a client to the server using TLS over TCP.

Note: For information on the SEM server, refer to the *EMS User's Manual*.

➢ To configure the SEM server:

1. Open the Session Experience Server page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Session Experience Server**).

Figure 68: Session Experience Server Page



▾ Session Experience Server

Server IP	0.0.0.0
Port	5000
Interface Name	DEFAULT

2. In the 'Server IP' field, enter the SEM server's IP address
3. In the 'Port' field, enter the port number for the SEM server.
4. In the 'Interface Name' field, enter the device's IP network interface on which the device sends the reports to the SEM server.
5. Click **Submit**, and then save ("burn") your settings to flash memory.

17.2 Configuring Quality of Experience Profiles

The Quality of Experience feature lets you monitor the quality of voice calls traversing the device in your network. Voice monitoring profiles can be done on specified network links, including IP Groups (see "Configuring IP Groups" on page 174), Media Realms (see "Configuring Media Realms" on page 164), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 166). These metrics include the following:

- **Mean Opinion Score (MOS):** MOS is the average grade on a quality scale, expressed as a single number in the range of 1 to 5, where 1 is the lowest audio quality and 5 the highest audio quality.
- **Delay (or latency):** Time it takes for information to travel from source to destination (round-trip time).
- **Packet Loss:** Lost packets are RTP packets that are not received by the voice endpoint. Packet loss can result in choppy voice transmission.
- **Jitter:** Jitter can result from uneven delays between received voice packets. To space

evenly, the device's jitter buffer adds delay. The higher the measurement, the greater the impact of the jitter buffer's delay on audio quality.

- **Residual Echo Return Loss (RERL):** An echo is a reflection of sound arriving at the listener at some time after the sound was initiated (often by the listener). Echo is typically caused by delay.

At any given time during a call, a voice metric can be in one of the following color-coded quality states:

- **Green:** Indicates good call quality
- **Yellow:** Indicates medium call quality
- **Red:** Indicates poor call quality

Quality of Experience Profiles lets you define quality thresholds per monitored voice metric. These are based on the following color-coded quality thresholds:

- **Green-Yellow threshold:** Lower threshold that indicates changes from Green to Yellow or vice versa when the threshold is crossed.
- **Yellow-Red threshold:** Higher threshold that indicates changes from Yellow to Red or vice versa when the threshold is crossed.

Hysteresis is also used to configure the threshold. This defines the amount of fluctuation from a threshold in order for the threshold to be considered as crossed (i.e., change in color state). Hysteresis is used to avoid false reports being sent by the device.

Each time a configured voice metric threshold is crossed (i.e., color changes), the device can do the following, depending on configuration:

- Report the change in the measured metrics to AudioCodes' Session Experience Manager (SEM) server. The SEM displays this call quality status for the associated SEM link (IP Group, Media Realm, or Remote Media Subnet). For configuring the SEM server's address, see "Configuring SEM Server" on page 155.
- Determine access control and media enhancements based on measured metrics. Depending on the crossed threshold type, you can configure the device to accept or reject calls, or use an alternative IP Profile for the IP Group to which the call belongs. For more information, see "Configuring Media Enhancement Profiles" on page 161.
- Alternative routing based on measured metrics. If a call is rejected because of a crossed threshold, the device generates a SIP 806 response. You can configure this SIP response code as a reason for alternative routing (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 262).

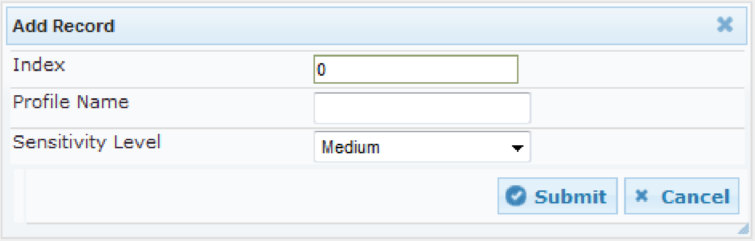
The procedure below describes how to configure Quality of Experience Profiles in the Web interface. You can also configure Quality of Experience Profiles using other management platforms:

- **Quality of Experience Profile table:** Table *ini* file parameter, QoEProfile or CLI command, configure voip/qoe qoe-profile
- **Quality of Experience Color Rules table:** Table *ini* file parameter, QOEColorRules or CLI command, configure voip/qoe qoe-profile qoe-color-rules

➢ To configure a QoE Profile:

1. Open the Quality of Experience Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Quality of Experience Profile**).
2. Click **Add**; the following dialog box appears:

Figure 69: Quality of Experience Profile - Add Record



Index: 0
Profile Name:
Sensitivity Level: Medium

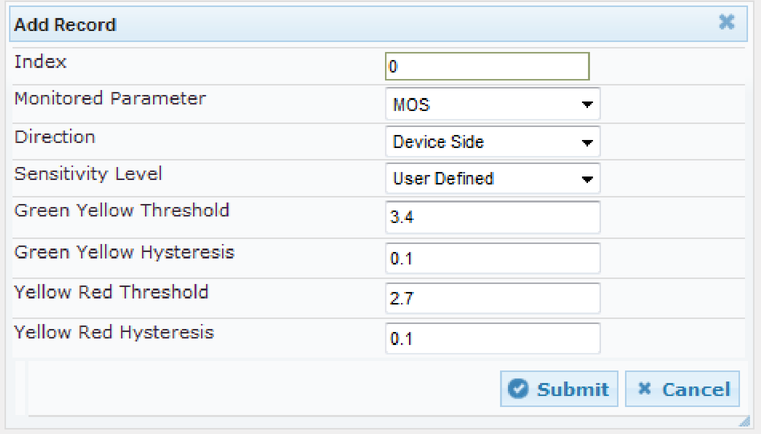
3. Configure a QoE Profile according to the parameters described in the table below.
4. Click **Submit**.

Table 52: Quality of Experience Profile Table Parameter Descriptions

Parameter	Description
Index **[QOEProfile_Index]**	Defines an index number for the new table record.
Profile Name CLI: name **[QOEProfile_Name]**	Defines an arbitrary name to easily identify the QoE Profile. The valid value is a string of up to 20 characters.
Sensitivity Level CLI: sensitivity-level **[QOEProfile_SensitivityLevel]**	Defines the pre-configured threshold profile to use. ▪ **[0]** User Defined = Need to define thresholds per monitored parameter in the Quality of Experience Color Rules table. ▪ **[1]** Low = Pre-configured low sensitivity thresholds. ▪ **[2]** Medium = Pre-configured medium sensitivity thresholds. ▪ **[3]** High = Pre-configured high sensitivity thresholds. Reporting is done for small fluctuations in parameter values.

5. In the Quality of Experience Profile page, select the QoE Profile index row for which you want to configure QoE thresholds, and then click the **Quality of Experience Color Rules** link located below the table; the Quality of Experience Color Rules page appears.
6. Click **Add**; the following dialog box appears:

Figure 70: Quality of Experience Page - Add Record Dialog Box



Index: 0
Monitored Parameter: MOS
Direction: Device Side
Sensitivity Level: User Defined
Green Yellow Threshold: 3.4
Green Yellow Hysteresis: 0.1
Yellow Red Threshold: 2.7
Yellow Red Hysteresis: 0.1

The figure above shows a configuration example where if the MOS value changes by 0.1 (hysteresis) to 3.3 or 3.5, the Green-Yellow threshold is crossed. The device considers a change to 3.3 as a Yellow state (i.e., medium quality) and a change to 3.5 as a Green state.

7. Configure a QoE Color rule according to the parameters described in the table below.
8. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 53: Quality of Experience Color Rules Table Parameter Descriptions

Parameter	Description
Index CLI: index **[QOEColorRules_ColorRuleIndex]**	Defines an index number for the new table record.
Monitored Parameter CLI: monitored-parameter **[QOEColorRules_monitoredParam]**	Defines the parameter to monitor and report. ▪ **[0]** MOS (default) ▪ **[1]** Delay ▪ **[2]** Packet Loss ▪ **[3]** Jitter ▪ **[4]** RERL [Echo]
Direction CLI: direction **[QOEColorRules_direction]**	Defines the monitoring direction. ▪ **[0]** Device Side (default) ▪ **[1]** Remote Side
Sensitivity Level CLI: sensitivity-level **[QOEColorRules_profile]**	Defines the sensitivity level of the thresholds. ▪ **[0]** User Defined = Need to define the thresholds in the parameters described below. ▪ **[1]** Low Sensitivity = Pre-configured low sensitivity threshold values. Thus, reporting is done only if changes in parameters' values is significant. ▪ **[2]** Average Sensitivity = (Default) Pre-configured medium sensitivity threshold values. ▪ **[3]** High Sensitivity = Pre-configured high sensitivity threshold values. Thus, reporting is done for small fluctuations in parameter values.
Green Yellow Threshold CLI: green-yellow-threshold **[QOEColorRules_GreenYellowThreshold]**	Defines the parameter threshold values between Green (good quality) and Yellow (medium quality) states. The valid threshold values are as follows: ▪ MOS values are in multiples of 10. For example, to denote a MOS of 3.2, the value 32 (i.e., 3.2*10) must be entered. ▪ Delay values are in msec. ▪ Packet Loss values are in percentage (%). ▪ Jitter is in msec. ▪ Echo measures the Residual Echo Return Loss (RERL) in dB.
Green Yellow Hysteresis CLI: green-yellow-hysteresis **[QOEColorRules_GreenYellowHysteresis]**	Defines the fluctuation (change) from the value configured for the Green-Yellow threshold. When the threshold is exceeded by this hysteresis, the device sends a report to the SEM indicating this change. **Note:** If the monitored parameter crosses two thresholds at once (e.g., from Green to Red), the device ignores the hysteresis value and reports the call state change to the SEM.
Yellow Red Threshold CLI: yellow-red-threshold **[QOEColorRules_YellowRedThreshold]**	Defines the parameter threshold values between Yellow (medium quality) and Red (poor quality) states. The valid threshold values are as follows: ▪ MOS values are in multiples of 10. For example, to denote a MOS of 3.2, the value 32 (i.e., 3.2*10) must be entered. ▪ Delay values are in msec. ▪ Packet Loss values are in percentage (%). ▪ Jitter is in msec. ▪ Echo measures the Residual Echo Return Loss (RERL) in dB.

Parameter	Description
Yellow Red Hysteresis CLI: yellow-red-hysteresis **[QOEColorRules_YellowRedHysteresis]**	Defines the fluctuation (change) from the value configured for the Yellow-Red threshold. When the threshold is exceeded by this hysteresis value, the device sends a report to the SEM indicating this change. **Note:** If the monitored parameter crosses two thresholds at once (e.g., from Green to Red), the device ignores the hysteresis value and reports the call state change to the SEM.

17.3 Configuring Bandwidth Profiles

Bandwidth Profiles enhance the device's monitoring of bandwidth utilization. A Bandwidth Profile defines bandwidth utilization thresholds for audio and/or video traffic (incoming and outgoing). Bandwidth Profiles can be assigned to IP Groups (see "Configuring IP Groups" on page 174), Media Realms (see "Configuring Media Realms" on page 164), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 166).

Each time a configured bandwidth threshold is crossed, the device can do the following, depending on configuration:

- Determine access control and media enhancements based on bandwidth utilization. Depending on the crossed threshold type, you can configure the device to accept or reject calls, or use an alternative IP Profile for the IP Group to which the call belongs. For more information, see "Configuring Media Enhancement Profiles" on page 161.
- Alternative routing based on bandwidth utilization. If a call is rejected because of a crossed threshold, the device generates a SIP 806 response. You can configure this SIP response code as a reason for alternative routing (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 262).
- Send an SNMP alarm (acMediaRealmBWThresholdAlarm). The device clears the alarm when bandwidth utilization returns to normal (within the thresholds).

The thresholds of Bandwidth Profiles use the same color-coding as the Quality of Experience Profile:

- **Green-Yellow threshold:** Lower threshold that indicates that the bandwidth exceeded a user-defined percentage of the configured threshold. This is referred to as a "Warning" alarm (i.e., warning you that bandwidth is nearing the threshold). When bandwidth goes over the threshold, the device considers it as a Yellow state; when it goes below the threshold, it considers it as a Green state.
- **Yellow-Red threshold:** Indicates that bandwidth has exceeded the configured threshold. When bandwidth goes over the threshold, the device considers it as a Red state; when it goes below the threshold, it considers it as a Yellow state.

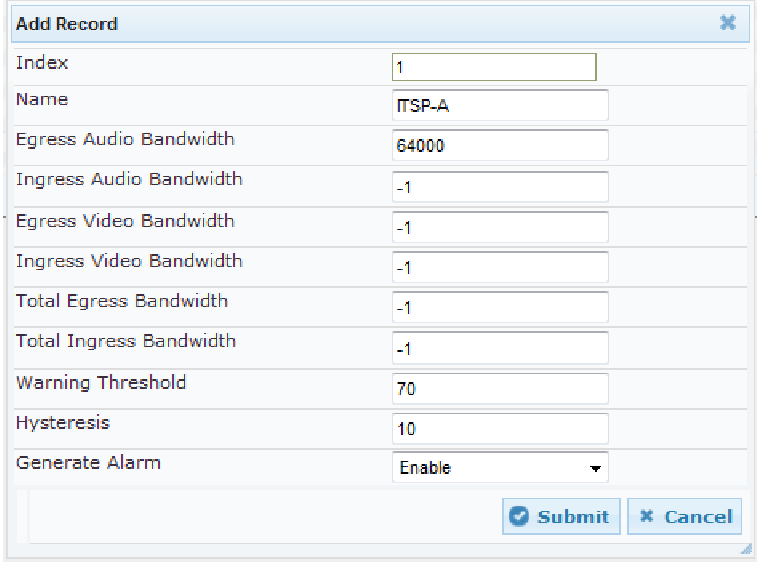
Hysteresis is also used to configure the threshold. This defines the amount of fluctuation from a threshold in order for the threshold to be considered as crossed (i.e., change in color state). Hysteresis is used to avoid false reports.

The procedure below describes how to configure Bandwidth Profiles in the Web interface. You can also configure Bandwidth Profiles using the table *ini* file parameter, BWProfile or CLI command, configure voip/qoe bw-profile.

➢ To configure Bandwidth Profiles:

1. Open the Bandwidth Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Bandwidth Profile**).
2. Click **Add**; the following dialog box appears:

Figure 71: Bandwidth Profile Page - Add Record



Add Record

Index	1
Name	ITSP-A
Egress Audio Bandwidth	64000
Ingress Audio Bandwidth	-1
Egress Video Bandwidth	-1
Ingress Video Bandwidth	-1
Total Egress Bandwidth	-1
Total Ingress Bandwidth	-1
Warning Threshold	70
Hysteresis	10
Generate Alarm	Enable

The figure above shows a configuration example where if the outgoing voice traffic threshold of 64,000 increases by 80% (70% warning threshold plus 10% hysteresis) to 115,200 (64,000 plus 51,200), a Yellow state occurs and an alarm is sent. If the threshold increases by 10%, a Red state occurs and an alarm is sent.

3. Configure a Bandwidth Profile according to the parameters described in the table below.
4. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

Table 54: Bandwidth Profile Table Parameter Descriptions

Parameter	Description
Index **[BWProfile_Index]**	Defines the index of the table row entry.
Name CLI: name **[BWProfile_Name]**	Defines an arbitrary name to easily identify the Bandwidth Profile. The valid value is a string of up to 20 characters.
Egress Audio Bandwidth CLI: egress-audio-bandwidth **[BWProfile_EgressAudioBandwidth]**	Defines the outgoing audio traffic threshold (in Kbps).
Ingress Audio Bandwidth CLI: ingress-audio-bandwidth **[BWProfile_IngressAudioBandwidth]**	Defines the incoming audio traffic threshold (in Kbps).
Egress Video Bandwidth CLI: egress-video-bandwidth **[BWProfile_EgressVideoBandwidth]**	Defines the outgoing video traffic threshold (in Kbps).
Ingress Video Bandwidth CLI: ingress-video-bandwidth **[BWProfile_IngressVideoBandwidth]**	Defines the incoming video traffic threshold (in Kbps).
Total Egress Bandwidth CLI: total-egress-bandwidth **[BWProfile_TotalEgressBandwidth]**	Defines the total (video and audio) outgoing bandwidth threshold (in Kbps).
Total Ingress Bandwidth CLI: total-ingress-bandwidth **[BWProfile_TotalIngressBandwidth]**	Defines the total (video and audio) incoming bandwidth threshold (in Kbps).

Parameter	Description
Warning Threshold CLI: warning-threshold **[BWProfile_WarningThreshold]**	Defines the threshold (in percentage) of the bandwidth thresholds that if exceeded is considered a Warning alarm (Green-Yellow threshold). This applies to any of the configured bandwidth thresholds. The Hysteresis is also added to this Warning threshold. For example, if set to 70% and the Hysteresis to 10%, when the current outgoing voice traffic exceeds 80% of the configured threshold, the Yellow state occurs and a Warning threshold alarm is sent if 'Generate Alarm' is set to **Enable**.
Hysteresis CLI: hysteresis **[BWProfile_hysteresis]**	Defines the bandwidth fluctuation (change) from the bandwidth threshold value (in percentage). The threshold is considered crossed if bandwidth exceeds the configured threshold plus this hysteresis, and a Red state occurs. For example, assume this parameter is set to 10% and the configured bandwidth threshold is set to 64000 Kbps. If current bandwidth reaches 70,400 Kbps (additional 10%), the threshold is considered crossed.
Generate Alarm CLI: generate-alarms **[BWProfile_GenerateAlarms]**	Enables the generation of an SNMP alarm if the threshold (with the hysteresis) is crossed. ▪ **[0]** Disable (default) ▪ **[1]** Enable If enabled, an alarm is sent if one of the following scenarios occurs: ▪ Warning threshold is exceeded (Warning severity - Yellow threshold). ▪ Any configured bandwidth threshold is exceeded (Major severity - Red threshold).

17.4 Configuring Media Enhancement Profiles

Media Enhancement Profiles provides support for access control and media quality enhancements based on call quality measurements (configured in ''Configuring Quality of Experience Profiles'' on page 155) and bandwidth utilization (configured in ''Configuring Bandwidth Profiles'' on page 159). These profiles contain color-coded thresholds that are used to trigger access control and/or media enhancements.

The Media Enhancement Profile table lets you configure any one of the following actions when a specific color-coded threshold (Green-Yellow or Yellow-Red) is crossed for a specific monitored voice metrics (e.g., MOS) or bandwidth (e.g., Egress Audio Bandwidth):

- Reject new calls until the voice metrics or bandwidth returns to below the threshold. This can be used, for example, to reject new calls when bandwidth threshold is exceeded.
- Use a different IP Profile. For example, if packet loss is detected, the IP Group (to which the Media Enhancement Rule is later assigned) can switch to an IP Profile configured with a higher RTP redundancy level. The ability to use a different IP Profile when call quality or bandwidth thresholds are crossed provides a wide range of options for media enhancement and traffic shaping. For example, it may be used to:
    - switch to a low bit-rate coder,
    - negotiate different p-time (and perform transrating if required),
    - increase RTP redundancy level,
    - or block video calls.
- Accept calls

A Media Enhancement Profile can later be assigned to an IP Group (in the IP Group table). However, when the device analyzes the call and determines whether Media Enhancement Profile should be applied or not, it searches for the "most relevant" Quality of Experience Profile or Bandwidth Profile in the following order: 1) Remote Media Subnet, 2) Media Realm, and then 3) IP Group. Thus, a Media Enhancement Profile associated with a specific IP Group may actually "respond" to Quality of Experience or bandwidth thresholds crossed at the Media Realm or Remote Media Subnet level.

Notes:

- The color-coded threshold is first calculated for the IP Group and only then for the Media Realm. The device uses the "worst" color-coded threshold crossing. For example, if a Media Realm crossed a Green-Yellow threshold and an IP Group a Yellow-Red threshold, the action defined for the Red color state is used.
- The device applies Media Enhancements Profiles on new calls **only**, based on the information gathered from previous and/or currently established calls.

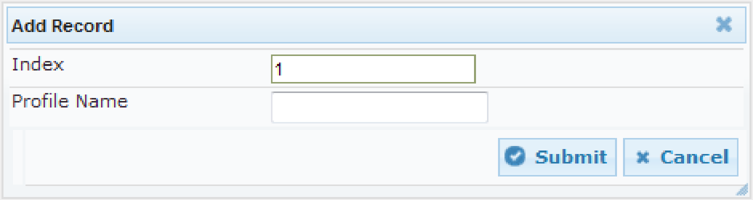
The procedure below describes how to configure Media Enhancement Profiles in the Web interface. You can also configure Media Enhancement Profiles using other management platforms:

- **Media Enhancement Profile table:** Table *ini* file parameter, MediaEnhancementProfile or CLI command, configure voip/qoe media-enhancement
- **Media Enhancement Rules table:** Table *ini* file parameter, MediaEnhancementRules or CLI command, configure voip/qoe media-enhancement-rules

➢ To configure a Media Enhancement Profile:

1. Open the Media Enhancement Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Media Enhancement Profile**).
2. Click **Add**; the following dialog box appears:

Figure 72: Media Enhancement Profile Table - Add Record



Add Record	
Index	1
Profile Name	

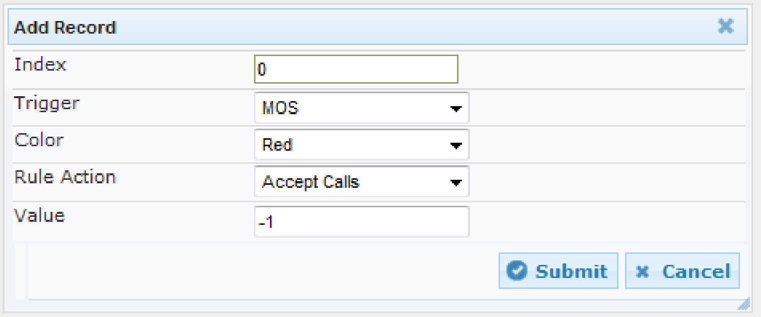
3. Configure a Media Enhancement Profile according to the parameters described in the table below.
4. Click **Submit**.

Table 55: Media Enhancement Profile Table Parameter Descriptions

Parameter	Description
Index **[MediaEnhancementProfile_Index]**	Defines the index of the table row entry.
Name CLI: profile-name **[MediaEnhancementProfile_ProfileName]**	Defines an arbitrary name to easily identify the Media Enhancement Profile. The valid value is a string of up to 20 characters.

5. In the Media Enhancement Profile table, select the required Media Enhancement Profile index row, and then click the **Media Enhancement Rules** link located below the table; the Media Enhancement Rules page appears.
6. Click **Add**; the following dialog box appears:

Figure 73: Media Enhancement Rules - Add Record



Add Record

Index	0
Trigger	MOS
Color	Red
Rule Action	Accept Calls
Value	-1

7. Configure a Media Enhancement Rule according to the parameters described in the table below.
8. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

Table 56: Media Enhancement Rules Table Parameter Descriptions

Parameter	Description
Index CLI: rule-index **[MediaEnhancementRules_RuleIndex]**	Defines the index of the table row entry.
Trigger CLI: trigger **[MediaEnhancementRules_Trigger]**	Defines the monitored metrics parameter or bandwidth associated with this rule. ▪ **[0]** MOS (default) ▪ **[1]** Delay ▪ **[2]** Packet Loss ▪ **[3]** Jitter ▪ **[4]** Bandwidth
Color CLI: color **[MediaEnhancementRules_Color]**	Defines the color-coded threshold change of the monitored metrics or bandwidth (configured in the 'Trigger' parameter) for which this rule is done. ▪ **[0]** Red (default) = Yellow-to-Red threshold is crossed. ▪ **[1]** Yellow = Green-to-Yellow threshold is crossed.
Rule Action CLI: action-rule **[MediaEnhancementRules_ActionRule]**	Defines the action that the device performs when the color-coded threshold is crossed: ▪ **[0]** Accept Calls (default) ▪ **[1]** Reject Calls ▪ **[2]** Alternative IP Profile = An alternative IP Profile ID is used, as configured in the 'Value' field (below). **Notes:** ▪ If this parameter is set to a restrictive action (i.e., **Reject Calls** or **Alternative IP Profile**) for Yellow and no action is set for Red, the device also applies the Yellow action to Red, if this color-coded threshold occurs. ▪ If this parameter is set to a permissive action (i.e., **Accept Calls**) for Red and no action is set for Yellow, the device applies the same action to Yellow, if this color-coded threshold occurs.
Value CLI: value **[MediaEnhancementRules_ActionValue]**	Defines an alternative IP Profile ID for the IP Group that is associated with this rule, if this rule is applied. This parameter is applicable only if the 'Rule Action' parameter is set to **Alternative IP Profile**.

18 Control Network

This section describes configuration of the network at the SIP control level.

18.1 Configuring Media Realms

The Media Realm table lets you configure a pool of up to 64 SIP media interfaces, termed *Media Realms*. Media Realms allow you to divide a Media-type interface (configured in the Interface table) into several realms, where each realm is specified by a UDP port range. Media Realms also define the maximum number of permitted media sessions. Media Realms can later be assigned to IP Groups (see "Configuring IP Groups" on page 174) and SRDs (see "Configuring SRDs" on page 169).

You can also apply the device's Quality of Experience feature to Media Realms:

- **Quality of Experience Profile:** Call quality monitoring based on thresholds for voice metrics (e.g., MOS) can be applied per Media Realm. For example, if MOS is considered poor, calls on this Media Realm can be rejected. For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 155.
- **Bandwidth Profile:** Bandwidth utilization thresholds can be applied per Media Realm. For example, if bandwidth thresholds are crossed, the device can reject any new new calls on this Media Realm. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 159.

You can also configure remote destination subnets per Media Realm and assign each subnet a Quality of Experience Profile and Bandwidth Profile. For configuring Remote Media Subnets, see "Configuring Remote Media Subnets" on page 166.

Notes:

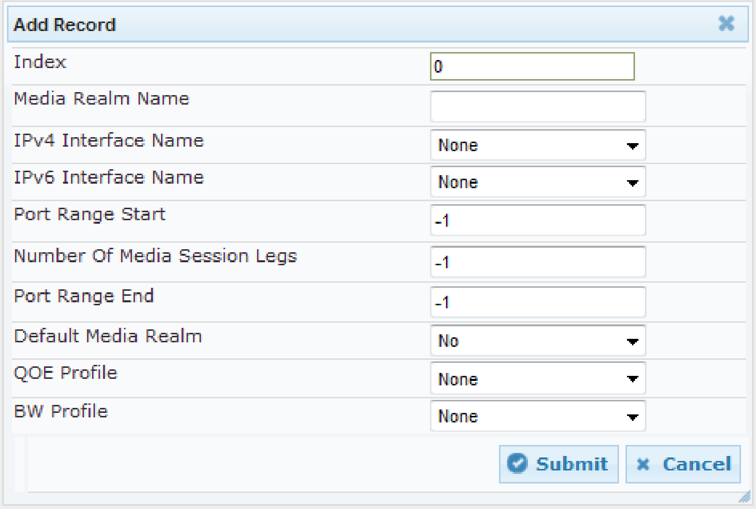
- If an IP Group is associated with an SRD and different Media Realms are assigned to the IP Group and SRD, the IP Group's Media Realm takes precedence.
- If you modify a Media Realm currently being used by a call, the device does not perform Quality of Experience for the call. If you delete the Media Realm during the call, the device maintains the call until the call parties end the call.

The procedure below describes how to configure Media Realms in the Web interface. You can also configure Media Realms using the table ini file parameter, CpMediaRealm or CLI command, configure voip/voip-network realm.

➤ To configure a Media Realm:

1. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **Media Realm Configuration**).
2. Click **Add**; the following dialog box appears:

Figure 74: Media Realm Page - Add Record Dialog Box



Add Record

Index	0
Media Realm Name	
IPv4 Interface Name	None
IPv6 Interface Name	None
Port Range Start	-1
Number Of Media Session Legs	-1
Port Range End	-1
Default Media Realm	No
QOE Profile	None
BW Profile	None

3. Configure the Media Realm according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 57: Media Realm Table Parameter Descriptions

Parameter	Description
Index **[CpMediaRealm_Index]**	Defines an index number for the new table record. The valid value is 0 to 63.
Media Realm Name CLI: name **[CpMediaRealm_MediaRealmName]**	Defines an arbitrary name to easily identify the Media Realm. The valid value is a string of up to 40 characters. **Notes:** ▪ This parameter is mandatory. ▪ The name assigned to the Media Realm must be unique.
IPv4 Interface Name CLI: ipv4 **[CpMediaRealm_IPv4IF]**	Assigns an IPv4 network interface to the Media Realm. This is the name of the interface as configured in the 'Interface Name' field of the Interface table.
IPv6 Interface Name CLI: ipv6if [CpMediaRealm_IPv6IF]	Assigns an IPv6 network interface to the Media Realm. This is the name of the interface as configured for the 'Interface Name' field of the Interface table.
Port Range Start CLI: port-range-start **[CpMediaRealm_PortRangeStart]**	Defines the starting port for the range of Media interface UDP ports. **Notes:** ▪ You must either configure all Media Realms with port ranges, or all without; not some with and some without. ▪ The available UDP port range is calculated using the BaseUDPport parameter: ✓ BaseUDPport to BaseUDPport + 4000*10 ▪ Port ranges over 60,000 must not be used. ▪ Media Realms must not have overlapping port ranges.
Number of Media Session Legs CLI: session-leg **[CpMediaRealm_MediaSessionLeg]**	Defines the number of media sessions associated with the range of ports. This is the number of media sessions available in the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.
Port Range End CLI: port-range-end **[CpMediaRealm_PortRangeEnd]**	(Read-only field) Displays the ending port for the range of media interface UDP ports. This field is calculated by adding the 'Media Session Leg' field (multiplied by the port chunk size) to the 'Port Range Start' field. A value appears once a row has been successfully added to the table.

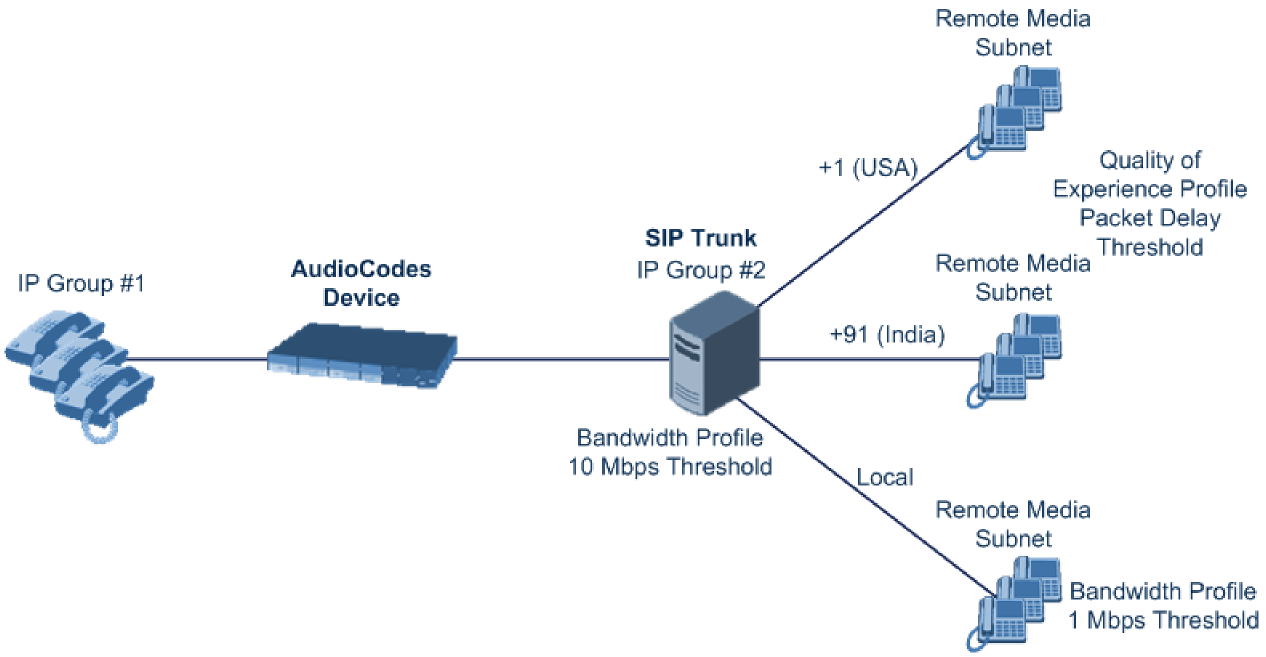
Parameter	Description
Default Media Realm CLI: is-default **[CpMediaRealm_IsDefault]**	Defines the Media Realm as the default Media Realm. This default Media Realm is used when no Media Realm is configured for an IP Group or SRD for a specific call. ▪ **[0]** No (default) ▪ **[1]** Yes **Notes:** ▪ This parameter can be set to Yes for only **one** defined Media Realm. ▪ If the parameter is not configured, then the first Media Realm in the table is used as default. ▪ If the table is not configured, the default Media Realm includes all the configured media interfaces.
QoE Profile CLI: qoe-profile **[CpMediaRealm_QoeProfile]**	Assigns a QoE Profile to the Media Realm. For configuring QoE Profiles, see "Configuring Quality of Experience Profiles" on page 155.
BW Profile CLI: bw-profile **[CpMediaRealm_BWProfile]**	Assigns a Bandwidth Profile to the Media Realm. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 159.

18.2 Configuring Remote Media Subnets

Remote Media Subnets define destination subnets for media (RTP/SRTP) traffic on a specific Media Realm. Each Remote Media Subnet can be assigned different call quality (Quality of Experience Profile) and bandwidth utilization (Bandwidth Profile) profiles. These profiles are configured in "Configuring Quality of Experience Profiles" on page 155 and "Configuring Bandwidth Profiles" on page 159, respectively. Thus, you can apply these profiles to remote media subnets instead of Media Realms or IP Groups. You can configure up to five Remote Media Subnets per Media Realm.

The figure below illustrates an example for implementing Remote Media Subnets. IP Group #2 represents a SIP Trunk which routes international (USA and India) and local calls. As international calls are typically more prone to higher delay than local calls, different Quality of Experience Profiles are assigned to them. This is done by creating Remote Media Subnets for each of these call destinations and assigning each Remote Media Subnet a different Quality of Experience Profile. A Quality of Experience Profile that defines a packet delay threshold is assigned to the international calls, which if crossed, a different IP Profile is used that defines higher traffic priority to voice over other traffic. In addition, IP Group #2 has a 10-Mbps bandwidth threshold and a "tighter" bandwidth limitation (e.g., 1 Mbps) is allocated to local calls. If this limit is exceeded, the device rejects new calls to this Remote Media Subnet.

Figure 75: Remote Media Subnets Example

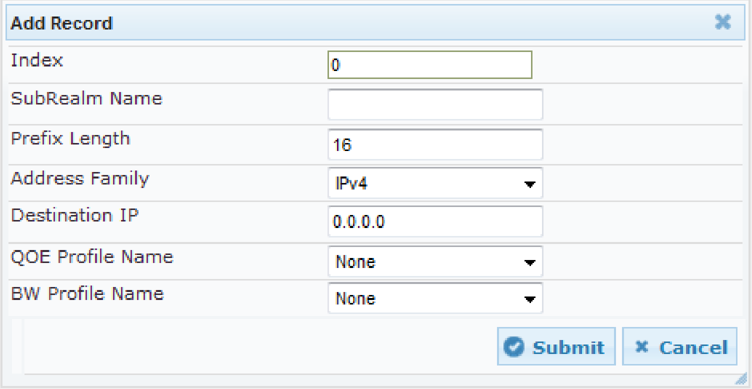


The procedure below describes how to configure Remote Media Subnets in the Web interface. You can also configure Remote Media Subnets using the table *ini* file parameter, RemoteMediaSubnet or CLI command, configure voip > voip-network realm remotemediasubnet.

➢ To configure a Remote Media Subnet:

1. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **Media Realm Configuration**).
2. Select the Media Realm index row for which you want to add Remote Media Subnets, and then click the **Remote Media Subnet** link located below the table; the Remote Media Subnet table appears.
3. Click **Add**; the following dialog box appears:

Figure 76: Remote Media Subnet - Add Record



Add Record

Index	0
SubRealm Name	
Prefix Length	16
Address Family	IPv4
Destination IP	0.0.0.0
QOE Profile Name	None
BW Profile Name	None

4. Configure the Remote Media Subnet according to the parameters described in the table below.
5. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 58: Remote Media Subnet Table Parameter Descriptions

Parameter	Description
Index **[RemoteMediaSubnet_RemoteMediaSubnetIndex]**	Defines an index number for the new table record.
Sub-Realm Name CLI: name **[RemoteMediaSubnet_RemoteMediaSubnetName]**	Defines an arbitrary name to easily identify the Remote Media Subnet. The valid value is a string of up to 20 characters.

Parameter	Description
Prefix Length CLI: prefix-length **[RemoteMediaSubnet_PrefixLength]**	Defines the subnet mask in Classless Inter-Domain Routing (CIDR) notation. For example, 16 denotes 255.255.0.0. The default is 16.
Address Family CLI: address-family **[RemoteMediaSubnet_AddressFamily]**	Defines the IP address protocol. ▪ **[2]** IPv4 Manual (default) ▪ **[10]** IPv6 Manual
Destination IP CLI: dst-ip-address **[RemoteMediaSubnet_DstIPAddress]**	Defines the IP address of the destination. The default is 0.0.0.0.
QOE Profile Name CLI: qoe-profile **[RemoteMediaSubnet_QOEProfileName]**	Assigns a Quality of Experience Profile to the Remote Media Subnet.
BW Profile Name CLI: bw-profile **[RemoteMediaSubnet_BWProfileName]**	Assigns a Bandwidth Profile to the Remote Media Subnet.

18.3 Configuring SRDs

The SRD table lets you configure up to 32 signaling routing domains (SRD). An SRD represents a logical VoIP network. Each logical or physical connection requires an SRD. For example, if the device interfaces with both the LAN and WAN, you would need to configure an SRD for each one.

The SRD is composed of the following:

- SIP Interface: The SIP Interface defines a listening port and type (TLS) for SIP signaling traffic on a specific logical IP network interface of the device.
- Media Realm: The Media Realm defines a UDP port range for RTP (media) traffic on a specific logical IP network interface of the device.

An SRD is a set of definitions together creating multiple, virtual multi-service IP gateways:

- Multiple and different SIP signaling interfaces (SRD associated with a SIP Interface) and RTP media (associated with a Media Realm) for multiple Layer-3 networks. Due to the B2BUA nature of the SBC application, different interfaces can be assigned to each leg of the call.
- Can operate with multiple gateway customers that may reside either in the same or in different Layer-3 networks as the device. This allows separation of signaling traffic between different customers. In such a scenario, the device is configured with multiple SRD's.

Typically, one SRD is defined for each SIP entity (e.g. proxies, IP phones, application servers, gateways, and softswitches) that communicate with each other. This provides these entities with VoIP services that reside on the same Layer-3 network (must be able to communicate without traversing NAT devices and must not have overlapping IP addresses). Routing from one SRD to another is possible, whereby each routing destination (IP Group or destination address) indicates the SRD to which it belongs.

Once configured, you can use the SRD as follows:

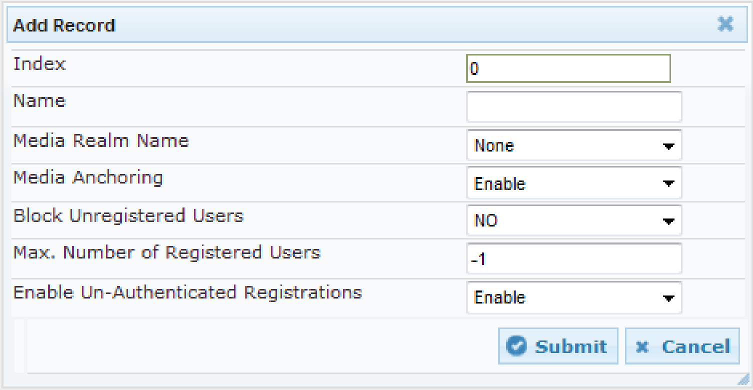
- Associate it with a SIP Interface (see "Configuring SIP Interfaces" on page 171)
- Associate it with an IP Group (see "Configuring IP Groups" on page 174)
- Associate it with a Proxy Set (see "Configuring Proxy Sets" on page 182)
- Associated it with an Admission Control rule (see Configuring Admission Control Table on page 244)
- Define it as a Classification rule for incoming SIP requests (see "Configuring Classification Rules" on page 249)
- Use it as a destination IP-to-IP routing rule (see Configuring SBC IP-to-IP Routing Rules on page 255)

The procedure below describes how to configure SRDs in the Web interface. You can also configure this using the table ini file parameter, SRD or CLI command, configure voip > voip-network srd.

➢ To configure an SRD:

1. Open the SRD Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **SRD Table**).
2. Click **Add**; the following dialog box appears:

Figure 77: SRD Settings Page



Add Record

Index	0
Name	
Media Realm Name	None
Media Anchoring	Enable
Block Unregistered Users	NO
Max. Number of Registered Users	-1
Enable Un-Authenticated Registrations	Enable

3. Configure an SRD according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 59: SRD Table Parameter Descriptions

Parameter	Description
Index **[SRD_Index]**	Defines an index for the new table record. The valid value is 0 to 31.
SRD Name CLI: name **[SRD_Name]**	Defines an arbitrary name to easily identify the SRD. The valid value can be a string of up to 21 characters. **Note:** This parameter is mandatory.
Media Realm Name CLI: media-realm **[SRD_MediaRealm]**	Assigns a Media Realm to the SRD. The listed Media Realms are the identifiable names that you configured for the Media Realms in the 'Media Realm Name' field of the Media Realm table (see "Configuring Media Realms" on page 164). **Note:** If the Media Realm is later deleted from the Media Realm table, this value becomes invalid in the SRD table.
Media Anchoring CLI: intra-srd-media-anchoring [SRD_IntraSRDMediaAnchoring]	Enables the Media Anchoring feature (Anti-Tromboning) per SRD, whereby RTP (media) flows directly between the call parties (i.e., does not traverse the device). ▪ [0] Enable = (Default) RTP traverses the device and each leg uses a different coder or coder parameters. ▪ [1] Disable = The RTP packet flow does not traverse the device; instead, the two SIP UAs establish a direct RTP/SRTP (media) flow between one another. **Notes:** ▪ If this parameter is enabled and the two call endpoints belong to the same SRD, calls cannot be established if the following scenario exists: a. One of the endpoints is defined as a foreign user (for example, “follow me service”) b. and one endpoint is located on the WAN and the other on the LAN. ▪ The reason for this is that in Media Anchoring, the device does not interfere in the SIP signaling such as manipulation of IP addresses, which is necessary for calls between LAN and WAN. ▪ When the global parameter SBCDirectMedia is disabled, Media Anchoring can only be enabled for calls between endpoints belonging to the same SRD. ▪ For more information on Media Anchoring, see No Media Anchoring (Anti-Tromboning) on page 222.
Block Unregistered Users CLI: block-un-reg-users	Determines whether the device blocks (rejects) incoming calls (INVITE requests) from unregistered users (pertaining to User-type IP Groups) for

Parameter	Description
[SRD_BlockUnRegUsers]	the SRD. ▪ [0] No = Calls from unregistered users are not blocked (default). ▪ [1] Yes = Blocks calls from unregistered users. **Note:** When the call is blocked, the device sends a SIP 500 "Server Internal Error" response to the remote end.
Max. Number of Registered Users CLI: max-reg-users [SRD_MaxNumOfRegUsers]	Maximum number of users belonging to this SRD that can register with the device. By default, no limitation exists for registered users
Enable Un-Authenticated Registrations CLI: enable-un-auth-registrs [SRD_EnableUnAuthenticatedRegistrations]	Determines whether the device blocks REGISTER requests from new users (i.e., users not registered in the device's registration database) when the destination is a User-type IP Group. ▪ [0] No = The device sends REGISTER requests to the SIP proxy server and only if authenticated by the server does the device add the user registration to its database. ▪ [1] Yes = The device adds REGISTER requests to its database even if the requests are not authenticated by a SIP proxy (default).

18.4 Configuring SIP Interfaces

The SIP Interface table lets you configure up to 32 SIP Interfaces. A SIP Interface defines a listening port and type (UDP, TCP, or TLS) for SIP signaling traffic on a specific logical IP network interface (configured in the Interface table). The SIP Interface can be configured for a specific application (i.e., , SBC) and associated with an SRD. For each SIP Interface, you can assign a SIP message policy, enable TLS mutual authentication, enable TCP keepalive, and determine the SIP response sent upon classification failure.

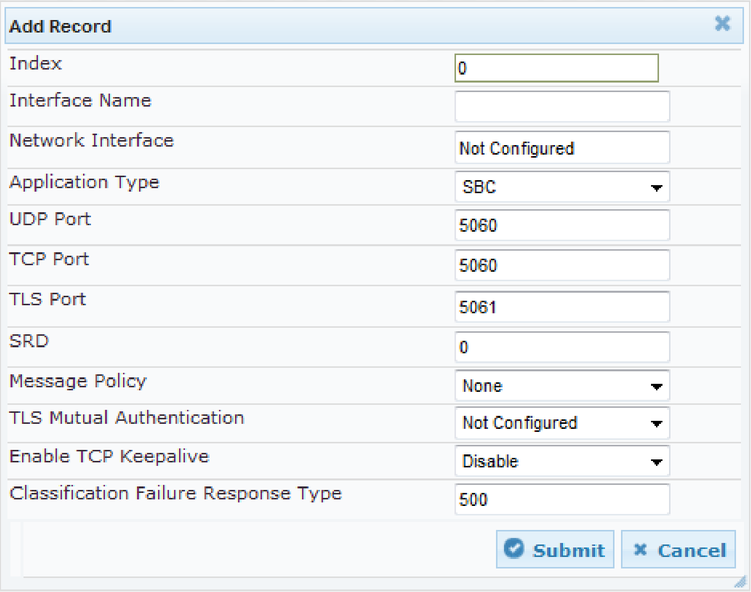
SIP Interfaces can be used, for example, for the following:

- Using SIP signaling interfaces per call leg (i.e., each SIP entity communicates with a specific SRD).
- Using different SIP listening ports for a single or for multiple IP network interfaces.
- Differentiating between applications by creating SIP Interfaces per application.
- Separating signaling traffic between networks (e.g., different customers) to use different routing tables, manipulations, SIP definitions, and so on.

The procedure below describes how to configure SIP interfaces in the Web interface. You can also configure this using the table ini file parameter, SIPInterface or the CLI command, configure voip > voip-network sip-interface.

➢ To configure a SIP Interface:

1. Open the SIP Interface Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **SIP Interface Table**).
2. Click **Add**; the following dialog box appears:



Index	0
Interface Name	
Network Interface	Not Configured
Application Type	SBC
UDP Port	5060
TCP Port	5060
TLS Port	5061
SRD	0
Message Policy	None
TLS Mutual Authentication	Not Configured
Enable TCP Keepalive	Disable
Classification Failure Response Type	500

3. Configure a SIP Interface according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 60: SIP Interface Table Parameter Descriptions

Parameter	Description
Index **[SIPInterface_Index]**	Defines an index for the new table record. The valid value is 0 to 31.
Interface Name CLI: interface-name **[SIPInterface_InterfaceName]**	Defines an arbitrary name to easily identify the SIP Interface. The valid value is a string of up to 21 characters.
Network Interface CLI: network-interface **[SIPInterface_NetworkInterface]**	Assigns a Control-type IP network interface to the SIP Interface. This string value must be identical (case-sensitive) to that configured in the 'Interface Name' field of the Interface table (see "Configuring IP Network Interfaces" on page 91). By default, no value is defined.
Application Type CLI: application-type **[SIPInterface_ApplicationType]**	Defines the application type associated with the SIP Interface. ▪ [2] SBC = SBC application.
UDP Port CLI: udp-port **[SIPInterface_UDPPort]**	Defines the listening and source UDP port. The valid range is 1 to 65534. The default is 5060. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
TCP Port CLI: tcp-port **[SIPInterface_TCPPort]**	Defines the listening TCP port. The valid range is 1 to 65534. The default is 5060. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
TLS Port CLI: tls-port **[SIPInterface_TLSPort]**	Defines the listening TLS port. The valid range is 1 to 65534. The default is 5061. **Notes:**

Parameter	Description
	▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
SRD CLI: srd **[SIPInterface_SRD]**	Assigns an SRD ID to the SIP Interface (configured in "Configuring SRDs" on page 169). The default is 0. **Notes:** ▪ You can assign the same SRD ID to up to two SIP Interfaces of the same application type. ▪ Each SIP Interface of the same application type (e.g., SBC) that is assigned to the same SRD must be configured with the same IP version (IPv4 or IPv6). ▪ All the SIP Interfaces that are assigned to the same SRD must have the same network interface (assigned in the 'Network Interface' parameter, above).
Message Policy CLI: message-policy **[SIPInterface_MessagePolicy]**	Assigns a SIP message policy to the SIP interface (configured in "Configuring SIP Message Policy Rules").
TLS Mutual Authentication CLI: tls-mutual-auth **[SIPInterface_TLSMutualAuthentication]**	Enables TLS mutual authentication for the SIP Interface. ▪ **[-1]** Not Configured = (Default) The SIPSRequireClientCertificate global parameter setting is applied. ▪ **[0]** Disable = Device does not request the client certificate for TLS connection on this SIP Interface. ▪ **[1]** Enable = Device requires receipt and verification of the client certificate to establish the TLS connection on this SIP Interface.
Enable TCP Keepalive CLI: tcp-keepalive-enable **[SIPInterface_TCPKeepaliveEnable]**	Enables the TCP Keep-Alive mechanism with the IP entity on this SIP Interface. TCP keep-alive can be used, for example, to keep a NAT entry open for clients located behind a NAT server, or simply to check that the connection to the IP entity is available. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For configuring TCP keepalive, use the following ini file parameters: TCPKeepAliveTime, TCPKeepAliveInterval, and TCPKeepAliveRetry.
Classification Failure Response Type CLI: classification_fail_response_type [SIPInterface_ClassificationFailureResponseType]	Defines the SIP response code that the device sends if a received SIP request (OPTIONS, REGISTER, or INVITE) has failed the SBC Classification process. The valid value can be a SIP response code from 400 through 699, or it can be set to 0 to not send any response at all. The default response code is 500 (Server Internal Error). This feature is important for preventing Denial of Service (DoS) attacks, typically initiated from the WAN. Malicious attackers can use SIP scanners to detect ports used by SIP devices. These scanners scan devices by sending UDP packets containing a SIP request to a range of specified IP addresses, listing those that return a valid SIP response. Once the scanner finds a device that supports SIP, it extracts information from the response and identifies the type of device (IP address and name) and can execute DoS attacks. A way to defend the device against such attacks is to not send a SIP reject response to these unclassified "calls" so that the attacker assumes that no device exists at such an IP address and port. **Note:** This parameter is applicable only if the device is set to reject unclassified calls. This is configured using the 'Unclassified Calls' parameter on the General Settings page (Configuration tab > VoIP menu > SBC > General Settings).

18.5 Configuring IP Groups

The IP Group table lets you configure up to 200 IP Groups. An IP Group represents a SIP entity in the network with which the device communicates. This can be a server (e.g., IP PBX or ITSP) or it can be a group of users (e.g., LAN IP phones). For servers, the IP Group is typically used to define the server's IP address by associating it with a Proxy Set (see Configuring Proxy Sets on page 182).

IP Groups can be used for the following:

- The IP Group table can be used to classify incoming SIP dialog-initiating requests (e.g., INVITE messages) to specific IP Groups. This classification is based on the Proxy Set ID associated with the IP Group. If the source address of the incoming SIP dialog is defined for this Proxy Set, the device assigns the SIP dialog to the associated IP Group. This feature is configured using the 'Classify by Proxy Set' parameter in the IP Group table.

Note: It is highly recommended to use the Classification table for classifying incoming SIP dialogs to IP Groups (see Configuring Classification Rules on page 249).

- IP Groups are used for configuring IP-to-IP routing rules where they represent the source and destination of the call (see Configuring SBC IP-to-IP Routing Rules on page 255).
- SIP dialog registration and authentication (digest user/password) of specific IP Groups (Served IP Group, e.g., corporate IP-PBX) with other IP Groups (Serving IP Group, e.g., ITSP). This is configured in the Account table (see "Configuring Registration Accounts" on page 187).

You can also apply the device's Quality of Experience feature to IP Groups:

- **Quality of Experience Profile:** Call quality monitoring based on thresholds for voice metrics (e.g., MOS) can be applied per IP Group. For example, if MOS is considered poor, calls belonging to this IP Group can be rejected. For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 155.
- **Bandwidth Profile:** Bandwidth utilization thresholds can be applied per IP Group. For example, if bandwidth thresholds are crossed, the device can reject any new calls on this IP Group. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 159.

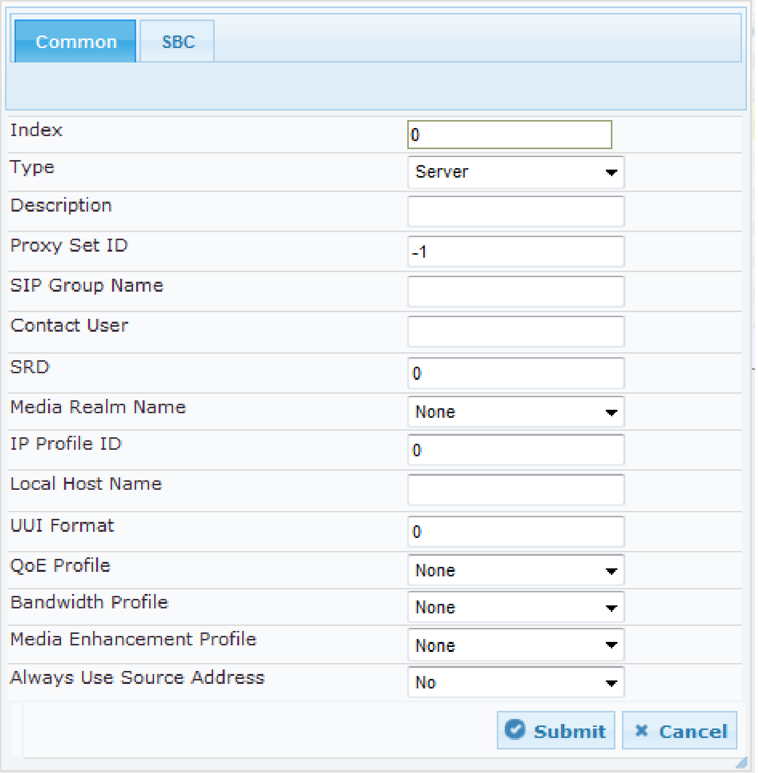
Notes:

- IP Group ID 0 cannot be used. This IP Group is set to default values and is used by the device when IP Groups are not implemented.
- When operating with multiple IP Groups, the default Proxy server must not be used (i.e., the IsProxyUsed parameter must be set to 0).
- If different SRDs are configured in the IP Group and Proxy Set tables, the SRD defined for the Proxy Set takes precedence.

The procedure below describes how to configure IP Groups in the Web interface. You can also configure IP Groups using the table ini file parameter, IPGroup or CLI command, configure voip > control-network ip-group.

➢ To configure an IP Group:

1. Open the IP Group Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **IP Group Table**).
2. Click **Add**; the following dialog box appears:



Index	0
Type	Server
Description	
Proxy Set ID	-1
SIP Group Name	
Contact User	
SRD	0
Media Realm Name	None
IP Profile ID	0
Local Host Name	
UUI Format	0
QoE Profile	None
Bandwidth Profile	None
Media Enhancement Profile	None
Always Use Source Address	No

3. Configure an IP Group according to to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 61: IP Group Table Parameter Descriptions

Parameter	Description
Common Parameters	
Index **[IPGroup_Index]**	Defines an index for the new table record.
Type CLI:type [IPGroup_Type]	Defines the type of IP Group: ▪ [0] Server = Used when the destination address, configured by the Proxy Set, of the IP Group (e.g., ITSP, Proxy, IP-PBX, or Application server) is known. ▪ [1] User = Represents a group of users such as IP phones and softphones where their location is dynamically obtained by the device when REGISTER requests and responses traverse (or are terminated) by the device. These users are considered remote (far-end) users. ▪ Typically, this IP Group is configured with a Serving IP Group that represents an IP-PBX, Application or Proxy server that serves this User-type IP Group. Each SIP request sent by a user of this IP Group is proxied to the Serving IP Group. For registrations, the device updates its internal database with the AOR and contacts of the users. Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group. The device forwards these responses directly to the SIP users. ▪ To route a call to a registered user, a rule must be configured in the SBC IP-to-IP Routing table. The device searches the dynamic database (by using the request URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained from this entry, and a SIP request is sent to the destination. ▪ The device also supports NAT traversal for the SIP clients located behind NAT. In this case, the device must be defined with a global IP address. ▪ [2] Gateway = This is applicable only to the SBC application in scenarios where the device receives requests to and from a gateway representing multiple users. This IP Group type is necessary as the other IP Group types are not suitable: ✓ The IP Group cannot be defined as a Server since its destination address is unknown during configuration. ✓ The IP Group cannot be defined as a User since the SIP Contact header of the incoming REGISTER does not represent a specific user. The Request-URI user part can change and therefore, the device is unable to identify an already registered user and therefore, adds an additional record to the database. ▪ The IP address of the Gateway IP Group is obtained dynamically from the host part of the Contact header in the REGISTER request received from the IP Group. Therefore, routing to this IP Group is possible only once a REGISTER request is received. If a REGISTER refresh request arrives, the device updates the new location (i.e., IP address) of the IP Group. If the REGISTER fails, no update is performed. If an UN-REGISTER request arrives, the IP address associated with the IP Group is deleted and therefore, no routing to the IP Group is done.
Description CLI: description **[IPGroup_Description]**	Defines a brief description for the IP Group. The valid value is a string of up to 29 characters. The default is an empty field.
Proxy Set ID CLI: proxy-set-id **[IPGroup_ProxySetId]**	Assigns a Proxy Set ID to the IP Group. All INVITE messages destined to this IP Group are sent to the IP address configured for the Proxy Set. **Notes:** ▪ Proxy Set ID 0 must **not** be used; this is the device's default Proxy.

Parameter	Description
	▪ The Proxy Set is applicable only to Server-type IP Groups. ▪ The SRD configured for this Proxy Set in the Proxy Set table is automatically assigned to this IP Group (see the 'SRD' field below). ▪ To configure Proxy Sets, see "Configuring Proxy Sets" on page 182.
SIP Group Name CLI: sip-group-name **[IPGroup_SIPGroupName]**	Defines the SIP Request-URI host name used in INVITE and REGISTER messages sent to this IP Group, or the host name in the From header of INVITE messages received from this IP Group. The valid value is a string of up to 100 characters. The default is an empty field. **Notes:** ▪ If this parameter is not configured, the value of the global parameter, ProxyName is used instead (see "Configuring Proxy and Registration Parameters" on page 190). ▪ If the IP Group is of User type, this parameter is used internally as a host name in the Request-URI for Tel-to-IP initiated calls. For example, if an incoming call from the device's T1 trunk is routed to a User-type IP Group, the device first creates the Request-URI (<destination_number>@<SIP Group Name>), and then it searches the internal database for a match.
Contact User CLI: contact-user **[IPGroup_ContactUser]**	Defines the user part of the From, To, and Contact headers of SIP REGISTER messages, and the user part of the Contact header of INVITE messages received from this IP Group and forwarded by the device to another IP Group. **Notes:** ▪ This parameter is applicable only to Server-type IP Groups. ▪ This parameter is overridden by the 'Contact User' parameter in the 'Account' table (see "Configuring Registration Accounts" on page 187).
SRD CLI: srd [IPGroup_SRD]	Assigns an SRD to the IP Group. The default is 0. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ To configure SRDs, see Configuring SRDs on page 169. ▪ For Server-type IP Groups, if you assign the IP Group with a Proxy Set ID (in the 'Proxy Set ID' field), the SRD field is automatically set to the SRD value assigned to the Proxy Set in the Proxy Set table.
Media Realm Name CLI: media-realm-name [IPGroup_MediaRealm]	Assigns a Media Realm to the IP Group. The string value must be identical (including case-sensitive) to the Media Realm name defined in the Media Realm table (see Configuring Media Realms on page 164). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If the Media Realm is deleted from the Media Realm table, this value becomes invalid.
IP Profile ID CLI: ip-profile-id **[IPGroup_ProfileId]**	Assigns an IP Profile to the IP Group. To configure IP Profiles, see "Configuring IP Profiles" on page 201. The default is 0.
Local Host Name CLI: local-host-name **[IPGroup_ContactName]**	Defines the host name (string) that the device uses in the SIP message's Via and Contact headers. This is typically used to define an FQDN as the host name. The device uses this string for Via and Contact headers in outgoing INVITE messages sent to a specific IP Group, and the Contact header in SIP 18x and 200 OK responses for incoming INVITE messages received from a specific IP Group. The Inbound IP Routing table can be used to identify the source IP Group from where the INVITE message was

Parameter	Description
	received. If this parameter is not configured (default), these headers are populated with the device's dotted-decimal IP address of the network interface on which the message is sent. **Note:** To ensure proper device handling, this parameter should be a valid FQDN.
UUI Format CLI: uui-format **[IPGroup_UUIFormat]**	Enables the generation of the Avaya UCID value, adding it to the outgoing INVITE sent to this IP Group. ▪ **[0]** Disabled (default) ▪ **[1]** Enabled This provides support for interworking with Avaya equipment by generating Avaya's UCID value in outgoing INVITE messages sent to Avaya's network. The device adds the UCID in the User-to-User SIP header. Avaya's UCID value has the following format (in hexadecimal): 00 + FA + 08 + node ID (2 bytes) + sequence number (2 bytes) + timestamp (4 bytes) This is interworked in to the SIP header as follows: `User-to-User: 00FA080019001038F725B3;encoding=hex` **Note:** To define the Network Node Identifier of the device for Avaya UCID, use the 'Network Node ID' (NetworkNodeId) parameter.
QoE Profile CLI: qoe-profile **[IPGroup_QOEProfile]**	Assigns a Quality of Experience Profile rule. For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 155.
Bandwidth Profile CLI: bandwidth-profile **[IPGroup_BWProfile]**	Assigns a Bandwidth Profile rule. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 159.
Media Enhancement Profile CLI: media-enhancement-profile **[IPGroup_MediaEnhancementProfile]**	Assigns a Media Enhancement Profile rule. For configuring Media Enhancement Profiles, see "Configuring Media Enhancement Profiles" on page 161.
Always Use Source Address CLI: always-use-source-addr **[IPGroup_AlwaysUseSourceAddr]**	Enables the device to always send SIP requests and responses, within a SIP dialog, to the source IP address received in the previous SIP message packet. This feature is especially useful in scenarios where the IP Group endpoints are located behind a NAT firewall (and the device is unable to identify this using its regular NAT mechanism). ▪ **[0]** No = (Default) The device sends SIP requests according to the settings of the global parameter, SIPNatDetection. ▪ **[1]** Yes = The device sends SIP requests and responses to the source IP address received in the previous SIP message packet. For information on NAT traversal, see Remote UA behind NAT.
CLI: Msg-Man-User-Defined-String1 **[IPGroup_MsgManUserDef1]**	Defines a value for the SIP user part that can be used in Message Manipulation rules configured in the Message Manipulations table. The Message Manipulation rule obtains this value from the IP Group, by using the following syntax: param.ipg.<src|dst>.user-defined.<0>. The valid value is a string of up to 30 characters. For configuring Message Manipulation rules, see "Configuring SIP Message Manipulation" on page 193.
CLI: Msg-Man-User-Defined-String2 **[IPGroup_MsgManUserDef2]**	Defines a value for the SIP user part that can be used in Message Manipulation rules configured in the Message Manipulations table. The Message Manipulation rule obtains this value from the IP Group, by using the following syntax: param.ipg.<src|dst>.user-defined.<1>. The valid value is a string of up to 30 characters.

Parameter	Description
	For configuring Message Manipulation rules, see "Configuring SIP Message Manipulation" on page 193.
SBC Parameters	
Classify By Proxy Set CLI: classify-by-proxy-set [IPGroup_ClassifyByProxySet]	Defines whether the incoming INVITE is classified to the IP Group according to its associated Proxy Set. ▪ [0] Disable ▪ [1] Enable (default) This classification occurs only if classification according to the device's database fails to (i.e., received INVITE is not from a registered user). The classification proceeds with checking whether the INVITE's IP address (if host names, then according to the dynamically resolved IP address list) is defined for a Proxy Set ID (in the Proxy Set table). If a Proxy Set ID has such an IP address, the device classifies the INVITE as belonging to the IP Group associated with this Proxy Set. The Proxy Set ID is assigned to the IP Group using the IP Group table's 'Proxy Set ID' parameter (see above). **Notes:** ▪ In cases where multiple IP Groups are associated with the same Proxy Set ID, do not enable this feature. If enabled, the device is unable to correctly classify the incoming INVITEs to their appropriate IP Groups. ▪ To enhance security, it is highly recommended to disable this parameter so that the device can use the Classification table rules to classify the call. If this parameter is enabled, the Classification table is not used if an associated Proxy Set is found. ▪ This parameter is applicable only to Server-type IP Groups.
Max. Number of Registered Users CLI: max-num-of-reg-users [IPGroup_MaxNumOfRegUsers]	Defines the maximum number of users in this IP Group that can register with the device. By default, no limitation exists for registered users. **Note:** This field is applicable only to User-type IP Groups.
Inbound Message Manipulation Set CLI: inbound-mesg-manipulation-set [IPGroup_InboundManSet]	Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the inbound message. To configure Message Manipulation rules, see Configuring SIP Message Manipulation on page 193.
Outbound Message Manipulation Set CLI: outbound-mesg-manipulation-set [IPGroup_OutboundManSet]	Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the outbound message. To configure Message Manipulation rules, see Configuring SIP Message Manipulation on page 193.
Registration Mode CLI: registration-mode [IPGroup_RegistrationMode]	Defines the registration mode for the IP Group: ▪ [0] User Initiates Registration (default) ▪ [1] SBC Initiates Registration = Used when the device serves as a client (e.g., with an IP PBX). This functions only with the User Info file. ▪ [2] Registrations not Needed = The device adds users to its database in active state.
Authentication Mode CLI: authentication-mode [IPGroup_AuthenticationMode]	Defines the authentication mode. ▪ [0] User Authenticates = (Default) The device does not handle the authentication, but simply passes the authentication messages between the SIP user agents. ▪ [1] SBC as Client = The device authenticates as a client. It receives the 401/407 response from the proxy requesting for authentication. The device sends the proxy the authorization credentials (i.e., username and password) according to one of the following: 1) account defined in the Account table (only if authenticating Server-type IP Group), 2) global username and password parameters (only if authenticating Server-type IP Group), 3) User Information file, or 4) sends request to users

Parameter	Description
	requesting credentials (only if authenticating User-type IP Group). ▪ [2] SBC as Server = The device acts as an Authentication server: ✓ Authenticates SIP clients, using the usernames and passwords in the User Information table (see SBC User Information for SBC User Database on page 305). This is applicable only to User-type IP Groups. ✓ Authenticates SIP severs. This is applicable only to Server-type IP Groups.
Authentication Method List CLI: authentication-method-list [IPGroup_MethodList]	Defines SIP methods received from the IP Group that must be challenged by the device, when the device acts as an Authentication server. If this parameter is not defined (i.e., empty value), no methods are challenged. The default value is null. Multiple entries are separated by a backslash "\", for example, INVITE\REGISTER. **Note:** This parameter is applicable only if the 'Authentication Mode' parameter is set to SBC as Server [2].
SBC Client Forking Mode CLI: enable-sbc-client-forking [IPGroup_EnableSBCClientForking]	Defines call forking of INVITE messages to up to five separate SIP outgoing legs for User-type IP Groups. This occurs if multiple contacts are registered under the same AOR in the device's registration database. ▪ [0] Sequential = (Default) Sequentially sends the INVITE to each contact. If there is no answer from the first contact, it sends the INVITE to the second contact, and so on until a contact answers. If no contact answers, the call fails or is routed to an alternative destination, if configured. ▪ [1] Parallel = Sends the INVITE simultaneously to all contacts. The call is established with the first contact that answers. ▪ [2] Sequential Available Only = Sequentially sends the INVITE only to available contacts (i.e., not busy). If there is no answer from the first available contact, it sends the INVITE to the second contact, and so on until a contact answers. If no contact answers, the call fails or is routed to an alternative destination, if configured. **Note:** The device can also fork INVITE messages received for a Request-URI of a specific contact (user) registered in the database to all other users located under the same AOR as the specific contact. This is configured using the SBCSendInviteToAllContacts parameter.
Source URI Input CLI: src-uri-input [IPGroup_SourceUriInput]	Defines the SIP header in the incoming INVITE that is used for call matching characteristics based on source URIs. ▪ [-1] Not Configured (default) ▪ [0] From ▪ [1] To ▪ [2] Request-URI ▪ [3] P-Asserted - First Header ▪ [4] P-Asserted - Second Header ▪ [5] P-Preferred ▪ [6] Route ▪ [7] Diversion ▪ [8] P-Associated-URI ▪ [9] P-Called-Party-ID ▪ [10] Contact ▪ [11] Referred-by **Notes:** ▪ This parameter is applicable only when classification is done according to the Classification table.

Parameter	Description
	- If the configured SIP header does not exist in the incoming INVITE message, the classification of the message to a source IP Group fails. - If the device receives an INVITE as a result of a REFER request or a 3xx response, then the incoming INVITE is routed according to the Request-URI. The device identifies such INVITEs according to a specific prefix in the Request-URI header, configured by the SBCXferPrefix parameter. Therefore, in this scenario, the device ignores this parameter setting.
Destination URI Input CLI: dst-uri-input [IPGroup_DestUriInput]	Defines the SIP header in the incoming INVITE that is used for call matching characteristics based on destination URIs. - [-1] Not Configured (default) - [0] From - [1] To - [2] Request-URI - [3] P-Asserted - First Header - [4] P-Asserted - Second Header - [5] P-Preferred - [6] Route - [7] Diversion - [8] P-Associated-URI - [9] P-Called-Party-ID - [10] Contact - [11] Referred-by **Notes:** - This parameter is applicable only when classification is done according to the Classification table. - If the configured SIP header does not exist in the incoming INVITE message, the classification of the message to a source IP Group fails. - If the device receives an INVITE as a result of a REFER request or a 3xx response, then the incoming INVITE is routed according to the Request-URI. The device identifies such INVITEs according to a specific prefix in the Request-URI header, configured by the SBCXferPrefix parameter. Therefore, in this scenario, the device ignores this parameter setting.
Username CLI: username [IPGroup_Username]	Defines the shared username for authenticating the IP Group, when the device acts as an Authentication server. The valid value is a string of up to 51 characters. By default, no username is defined. **Notes:** - This parameter is applicable only to Server-type IP Groups and when the 'Authentication Mode' parameter is set to SBC as Server (i.e., authentication of servers). - To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.
Password CLI: password IPGroup_Password]	Defines the shared password for authenticating the IP Group, when the device acts as an Authentication server. The valid value is a string of up to 51 characters. By default, no password is defined. **Notes:** - This parameter is applicable only to Server-type IP Groups and when the 'Authentication Mode' parameter is set to SBC as Server (i.e., authentication of servers).

Parameter	Description
	▪ To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.

18.6 Configuring Proxy Sets

The Proxy Sets table lets you configure up to 200 Proxy Sets. A Proxy Set defines the destination address (IP address or FQDN) and transport type (e.g., UDP) of a SIP server (e.g., Proxy). Each Proxy Set can be configured with up to 10 addresses, enabling you to implement load balancing and redundancy between multiple servers. If you configure the address as an FQDN, you can configure the method for resolving the domain name to an IP address.

Proxy Sets are later assigned to **Server-type** IP Groups, in the IP Group table. When the device sends an INVITE message to an IP Group, it sends it to the address configured for the Proxy Set. You can also enable the classification of incoming SBC SIP dialogs to IP Groups based on Proxy Set. If the source address of the incoming SIP dialog is the same as the address of a Proxy Set that is assigned to an IP Group, the device classifies the SIP dialog as belonging to that IP Group. This feature is configured using the 'Classify by Proxy Set' parameter in the IP Group table. For configuring IP Groups, see "Configuring IP Groups" on page 174.

Note: For classifying incoming SIP dialogs to IP Groups, it is highly recommended to use ONLY the Classification table (see Configuring Classification Rules on page 249).

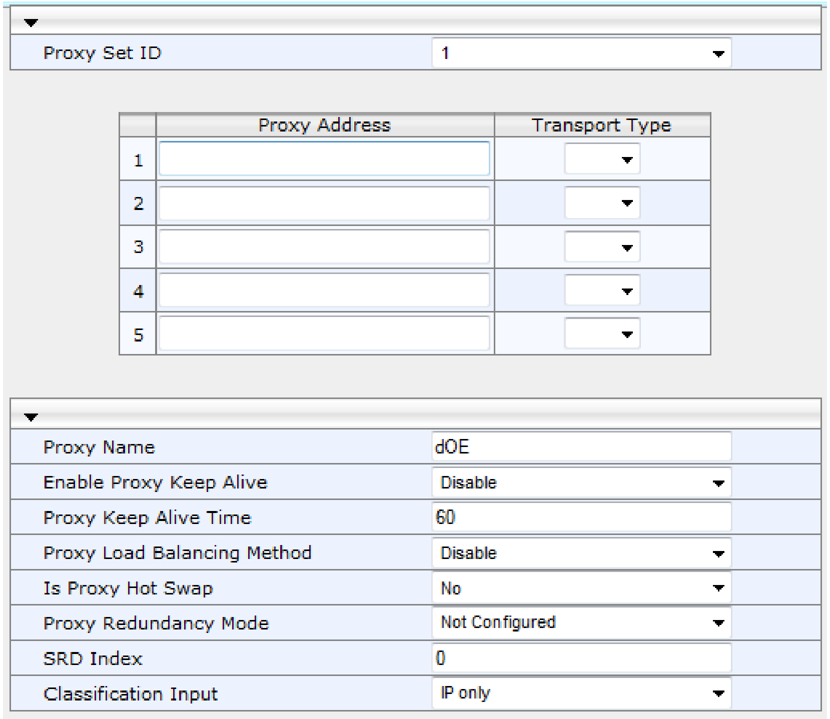
The procedure below describes how to configure Proxy Sets in the Web interface. You can also configure Proxy Sets using the following management tools:

- Proxy Set ID with IP addresses: table ini file parameter, ProxyIP or CLI command, configure voip > voip-network proxy-ip > proxy-set-id
- Attributes for the Proxy Set: table ini file parameter, ProxySet or CLI command, configure voip > voip-network proxy-set

➢ To configure a Proxy Set:

1. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **Proxy Sets Table**).

Figure 78: Proxy Sets Table Page



Proxy Set ID	1

	Proxy Address	Transport Type
1		
2		
3		
4		
5		

Proxy Name	dOE
Enable Proxy Keep Alive	Disable
Proxy Keep Alive Time	60
Proxy Load Balancing Method	Disable
Is Proxy Hot Swap	No
Proxy Redundancy Mode	Not Configured
SRD Index	0
Classification Input	IP only

2. Configure a Proxy Set according to the parameters described in the table below.
3. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 62: Proxy Sets Table Parameter Description

Parameter	Description
Web: Proxy Set ID CLI: configure voip > voip-network proxy-set **[ProxySet_Index]**	Defines an index number for the new table record. To summarize, if the default Proxy Set is used, the INVITE message is sent according to the following preferences: Typically, when IP Groups are used, there is no need to use the default Proxy and all routing and registration rules can be configured using IP Groups and the Account tables (see "Configuring Registration Accounts" on page 187). **Note:** Proxy Set ID 0 is the default Proxy Set.
Proxy Address CLI: voip-network proxy-ip > proxy-address **[ProxyIp_IpAddress]**	Defines the address of the Proxy server. Up to 10 addresses can be configured per Proxy Set. The address can be defined as an IP address in dotted-decimal notation (e.g., 201.10.8.1) or FQDN. You can also specify the port in the following format: ▪ IPv4 address: <IP address>:<port> (e.g., 201.10.8.1:5060) ▪ IPv6 address: <[IPV6 address]>:<port> (e.g., [2000::1:200:200:86:14]:5060)
Transport Type CLI: voip-network proxy-ip > transport-type **[ProxyIp_TransportType]**	Defines the transport type for communicating with the Proxy server. ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS ▪ **[-1]** = Undefined **Note:** If this parameter is not defined, the settings of the global parameter SIPTransportType is used.
Proxy Name CLI: proxy-name **[ProxySet_ProxyName]**	Defines an arbitrary name to easily identify the Proxy Set. The valid value is a string of up to 20 characters.

Parameter	Description
DNS Resolve Method CLI: dns-resolve-method **[ProxySet_DNSResolveMethod]**	Defines the DNS query record type for resolving the Proxy server's host name into an IP address. ▪ **[-1]** = DNS resolving is done according to the settings of the global parameter, Proxy DNS Query Type. ▪ **[0]** A-Record = (Default) A-record DNS query. ▪ **[1]** SRV = If the Proxy address is configured with a domain name without a port (e.g., domain.com), an SRV query is done. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Thus, if the first SRV query returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed. If the configured Proxy address contains a domain name with a port (e.g., domain.com:5080), the device performs a regular DNS A-record query. ▪ **[2]** NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the configured Proxy address contains a domain name with a port (e.g., domain.com:5080), the device performs a regular DNS A-record query. If the transport type is configured for the Proxy address, a NAPTR query is not performed.
Web: Enable Proxy Keep Alive CLI: voip-network proxy-set > proxy-enable-keep-alive **[ProxySet_EnableProxyKeepAlive]**	Enables the device's Proxy Keep-Alive mechanism, which checks communication with the Proxy server. ▪ **[0]** Disable (default). ▪ **[1]** Using Options = Enables the Proxy Keep-Alive mechanism using SIP OPTIONS messages. The device sends these message every user-defined interval, configured by the 'Proxy Keep Alive Time' parameter. If the device receives a SIP response code that is also configured in the 'Keep-Alive Failure Responses' parameter (below), the device considers the Proxy as down. ▪ **[2]** Using Register = Enables the Proxy Keep-Alive mechanism using SIP REGISTER messages. The device sends the REGISTER message every user-defined interval, configured by the SBCProxyRegistrationTime parameter. Any SIP response from the Proxy - success (200 OK) or failure (4xx response) - is considered as if the Proxy is "alive". If the Proxy does not respond to INVITE messages sent by the device, the Proxy is considered as down (offline). If you enable Proxy Keep-Alive mechanism, the device can operate with multiple Proxy servers (addresses) for redundancy and load balancing (configured by the 'Proxy Load Balancing Method' parameter). **Notes:** ▪ For Survivability mode for User-type IP Groups, this parameter must be enabled (1 or 2). ▪ If this parameter is enabled and the Proxy uses the TCP/TLS transport type, you can enable CRLF Keep-Alive mechanism, using the UsePingPongKeepAlive parameter.
Web: Proxy Keep Alive Time CLI: voip-network proxy-set > proxy-keep-alive-time **[ProxySet_ProxyKeepAliveTime]**	Defines the interval (in seconds) between Keep-Alive messages sent by the device when the Keep-Alive mechanism is enabled. The valid range is 5 to 2,000,000. The default is 60. **Note:** This parameter is applicable only if the 'Enable Proxy Keep Alive' parameter is set to **Using Options**.
Web: Keep-Alive Failure Responses CLI: keepalive-fail-resp	Defines SIP response codes that if any is received in response to a keep-alive message using SIP OPTIONS, the device considers the Proxy as

Parameter	Description
[ProxySet_KeepAliveFailureResp]	down. Up to three response codes can be configured, where each code is separated by a comma (e.g., 407,404). By default, no responses are defined. If no responses are configured or responses received are not those configured, the proxy is considered "alive". **Note:** The SIP 200 response code is not supported by this feature.
Web: Proxy Load Balancing Method CLI: voip-network proxy-set > proxy-load-balancing-method **[ProxySet_ProxyLoadBalancingMethod]**	Enables the Proxy Load Balancing mechanism per Proxy Set. ▪ **[0]** Disable = Load Balancing is disabled (default) ▪ **[1]** Round Robin = A list of all possible Proxy IP addresses is compiled. This list includes all IP addresses per Proxy Set after necessary DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive mechanism (according to parameters EnableProxyKeepAlive and ProxyKeepAliveTime) tags each entry as 'offline' or 'online'. Load balancing is only performed on Proxy servers that are tagged as 'online'. All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP addresses list is refreshed according to ProxyIPListRefreshTime. If a change in the order of the entries in the list occurs, all load statistics are erased and balancing starts over again. ▪ **[2]** Random Weights = The outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server, using SRV records. The device sends the requests in such a fashion that each Proxy receives a percentage of the requests according to its' assigned weight. A single FQDN should be configured as a Proxy IP address. Random Weights Load Balancing is not used in the following scenarios: ✓ The Proxy Set includes more than one Proxy IP address. ✓ The only Proxy defined is an IP address and not an FQDN. ✓ SRV is not enabled (DNSQueryType). ✓ The SRV response includes several records with a different Priority value.
Web: Is Proxy Hot Swap CLI: voip-network proxy-set > is-proxy-hot-swap **[ProxySet_IsProxyHotSwap]**	Enables the Proxy Hot-Swap redundancy mechanism, which provides real-time switching from the primary Proxy server to redundant Proxies when no response is received from the primary. ▪ **[0]** No (default) ▪ **[1]** Yes = The device sends the SIP INVITE/REGISTER message to the first address (Proxy/Registrar server) listed in the Proxy Set. If a SIP response is received and this response code is defined in the 'Keep Alive Failure Response' parameter (above), the device assumes the Proxy as down and sends the message again; otherwise, the device assumes the proxy "alive" and does not send the message again. Each time a defined response code is received, the device re-sends the message. This can occur until a user-defined maximum number of retransmissions, configured by the HotSwapRtx parameter, after which the device sends the same message to the next address (redundant Proxy/Registrar), and so on. If there is no response from any of the Proxies, the device goes through the address list again until a "live" Proxy is located.
Web: Proxy Redundancy Mode CLI: voip-network proxy-set > proxy-redundancy-mode **[ProxySet_ProxyRedundancyMode]**	Determines whether the device switches from a redundant Proxy to the primary Proxy when it becomes available again. ▪ **[-1]** Not configured = (Default) The global parameter, ProxyRedundancyMode applies. ▪ **[0]** Parking = The device continues operating with the redundant (now active) Proxy until the next failure, after which it operates with the next

Parameter	Description
	redundant Proxy. - **[1]** Homing = The device always attempts to operate with the primary Proxy. The device switches back to the primary Proxy whenever it becomes available. **Notes:** - To enable this functionality, you must also enable the Proxy Keep-Alive mechanism (using the 'Enable Proxy Keep Alive' parameter). - The **Homing** option can only be used if the 'Enable Proxy Keep Alive' parameter is set to **Using Options.**
Web: SRD Index CLI: voip-network proxy-set > srd-id [ProxySet_ProxySet_SRD]	Assigns an SRD to the Proxy Set ID. The default is SRD 0. **Notes:** - For this parameter to take effect, a device reset is required. - To configure SRDs, see Configuring SRDs on page 169.
Web: Classification Input CLI: voip-network proxy-set > classification-input [ClassificationInput]	Defines how the device classifies IP calls to the Proxy Set. - [0] IP Only = (Default) The call is classified to the Proxy Set according to its IP address only. - [1] IP + Port + Transport = The call is classified to the Proxy Set according to its IP address, port, and transport type. **Note:** This parameter is applicable only if the IP Group table's parameter, 'Classify by Proxy Set' is set to Enable.

19 SIP Definitions

This section describes configuration of SIP parameters.

19.1 Configuring SIP Parameters

Many of the stand-alone SIP parameters associated with various features can be configured in the following pages:

- **SIP General Parameters page:** Provides SIP parameters for configuring general SIP features. To access this page, use the following path: **Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**.
- **SIP Advanced Parameters page:** Provides SIP parameters for configuring advanced SIP features. To access this page, use the following path: **Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**.

For a description of these parameters, refer to the section corresponding to the feature or see "Configuration Parameters Reference" on page 380.

19.2 Configuring Registration Accounts

The Account table lets you configure up to 200 Accounts. An Account defines registration information for registering and authenticating (digest) "served" IP Groups (e.g., IP PBX) with a "serving" IP Group (e.g., ITSP). Registration information includes a username, password, host name (AOR), and contact user name (AOR). The device includes this information in the REGISTER message sent to the "serving" IP Group. Up to 10 Accounts can be configured per "served" IP Group.

A "served" IP Group can register to more than one "serving" IP Group (e.g., multiple ITSPs). This is done by configuring multiple entries in the Account table for the same "served" IP Group, but with different "serving" IP Groups, user name/password, host name, and contact user values.

Note: If no match is found in the Account table for incoming or outgoing calls, the username and password is taken from:

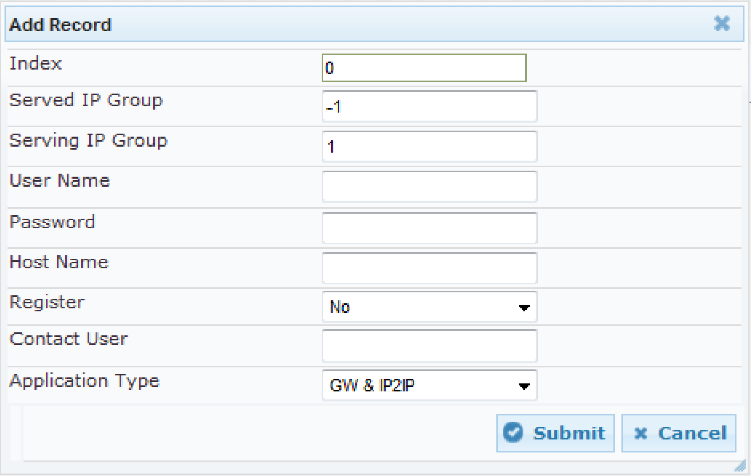
-
- 'UserName' and 'Password' parameters on the Proxy & Registration page

The procedure below describes how to configure Accounts in the Web interface. You can also configure Accounts using the table ini file parameter, Account or CLI command, configure voip > sip-definition account.

➢ To configure an Account:

1. Open the Account Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Account Table**).
2. Click **Add**; the following dialog box appears:

Figure 79: Account Table - Add Record



Add Record

Index	0
Served IP Group	-1
Serving IP Group	1
User Name	
Password	
Host Name	
Register	No
Contact User	
Application Type	GW & IP2IP

3. Configure an account according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Once you have configured Accounts, you can register or un-register them, as described below:

➢ To register or un-register an Account:

1. In the table, select the required Account entry row.
2. From the **Action** drop-down list, choose one of the following commands:
   - **Register** to register the Account.
   - **Un-Register** to un-register an Account.

To view Account registration status, see "Viewing Registration Status" on page 341.

Table 63: Account Table Parameter Descriptions

Parameter	Description
Index	Defines an index for the new table record.
Served IP Group CLI: served-ip-group [Account_ServedIPGroup]	Defines the IP Group (e.g., IP-PBX) that you want to register and/or authenticate.
Serving IP Group CLI: serving-ip-group **[Account_ServingIPGroup]**	Defines the IP Group to where the device sends the SIP REGISTER requests (if enabled) for registration and authentication.
User Name CLI: user-name **[Account_Username]**	Defines the digest MD5 Authentication username. The valid value is a string of up to 50 characters.
Password CLI: password **[Account_Password]**	Defines the digest MD5 Authentication password. The valid value is a string of up to 50 characters.
Host Name CLI: host-name **[Account_HostName]**	Defines the Address of Record (AOR) host name. The host name appears in SIP REGISTER From/To headers as ContactUser@HostName. For a successful registration, the host name is also included in the URI of the INVITE From header. The valid value is a string of up to 49 characters. **Note:** If this parameter is not configured or if registration fails, the 'SIP Group Name' parameter value configured in the IP Group table is used instead.
Register	Enables registration.

Parameter	Description
CLI: register **[Account_Register]**	▪ **[0]** No (Default) ▪ **[1]** Regular = Regular registration process. For more information, see "Regular Registration Mode" on page 189. ▪ **[2]** GIN = Registration for legacy PBXs, using Global Identification Number (GIN). For more information, see "Single Registration for Multiple Phone Numbers using GIN" on page 189. **Note:** The account registration is not affected by the IsRegisterNeeded parameter.
Contact User CLI: contact-user **[Account_ContactUser]**	Defines the AOR username. This appears in REGISTER From/To headers as ContactUser@HostName, and in INVITE/200 OK Contact headers as ContactUser@<device's IP address>. **Notes:** ▪ If this parameter is not configured, the 'Contact User' parameter in the IP Group table is used instead. ▪ If registration fails, the user part in the INVITE Contact header contains the source party number.
Application Type CLI: application-type **[Account_ApplicationType]**	Defines the application type: ▪ [2] SBC = SBC application.

19.2.1 Regular Registration Mode

When you configure the registration mode in the Account table to **Regular**, the device sends REGISTER requests to the Serving IP Group. The host name (in the SIP From/To headers) and contact user (user in From/To and Contact headers) are taken from the configured Account table upon successful registration. See the example below:

```
REGISTER sip:xyz SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac1397582418
From: <sip:ContactUser@HostName>;tag=1c1397576231
To: <sip: ContactUser@HostName >
Call-ID: 1397568957261200022256@10.33.37.78
CSeq: 1 REGISTER
Contact: <sip:ContactUser@10.33.37.78>;expires=3600
Expires: 3600
User-Agent: Sip-Gateway/v.6.00A.008.002
Content-Length: 0
```

19.2.2 Single Registration for Multiple Phone Numbers using GIN

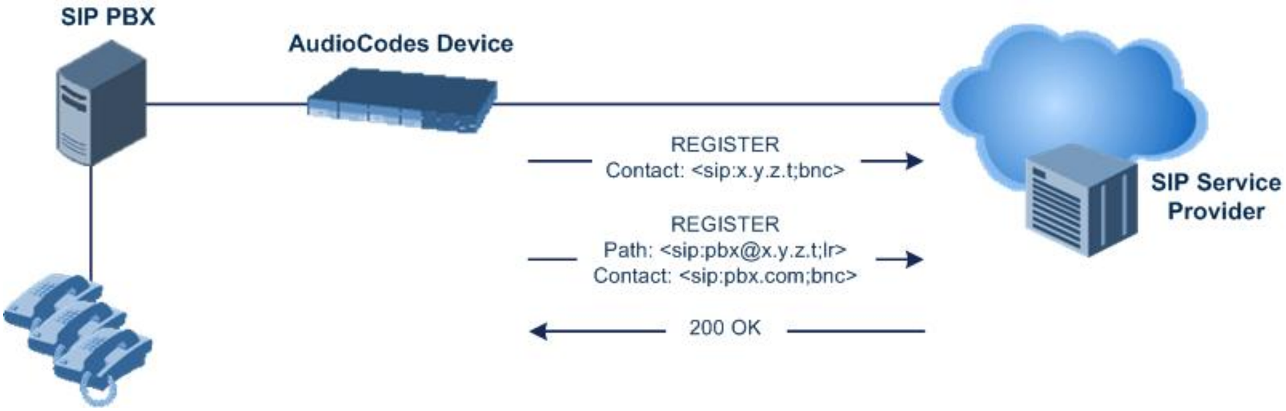
When you configure the registration mode in the Account table to **GIN**, the Global Identifiable Number (GIN) registration method is used, according to RFC 6140. The device performs GIN-based registration of users to a SIP registrar on behalf of a SIP PBX. In effect, the PBX registers with the service provider, just as a directly hosted SIP endpoint would register. However, because a PBX has multiple user agents, it needs to register a contact address on behalf of each of these. Rather than performing a separate registration procedure for each user agents, GIN registration mode does multiple registrations using a single REGISTER transaction.

According to this mechanism, the PBX delivers to the service provider in the Contact header field of a REGISTER request a template from which the service provider can construct contact URIs for each of the AORs assigned to the PBX and thus, can register these contact URIs within its location service. These registered contact URIs can then be used to deliver to the PBX inbound requests targeted at the AORs concerned. The mechanism can be used with AORs comprising SIP URIs based on global E.164 numbers and the service provider's domain name or sub-domain name.

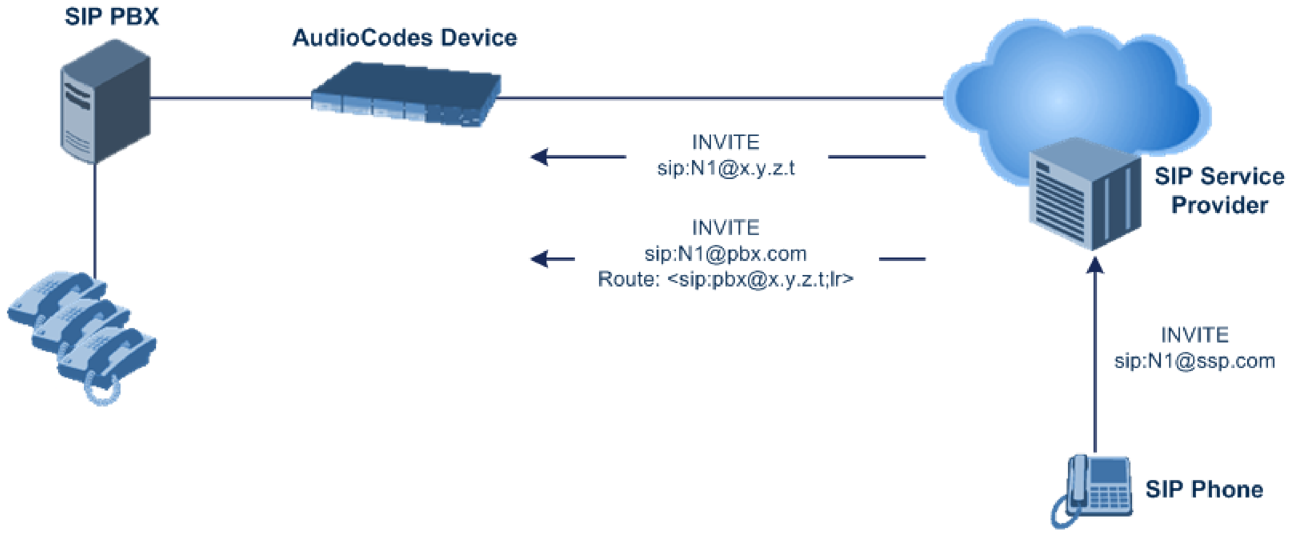
The SIP REGISTER request sent by the device for GIN registration with a SIP server provider contains the Require and Proxy-Require headers. These headers contain the token 'gin'. The Supported header contains the token 'path' and the URI in the Contact header contains the parameter 'bnc' without a user part:

```
Contact: <sip:198.51.100.3;bnc>;
```

The figure below illustrates the GIN registration process:


SIP PBX
AudioCodes Device
REGISTER
Contact: <sip:x.y.z.t;bnc>
REGISTER
Path: <sip:pbx@x.y.z.t;lr>
Contact: <sip:pbx.com;bnc>
200 OK
SIP Service Provider

The figure below illustrates an incoming call using GIN:



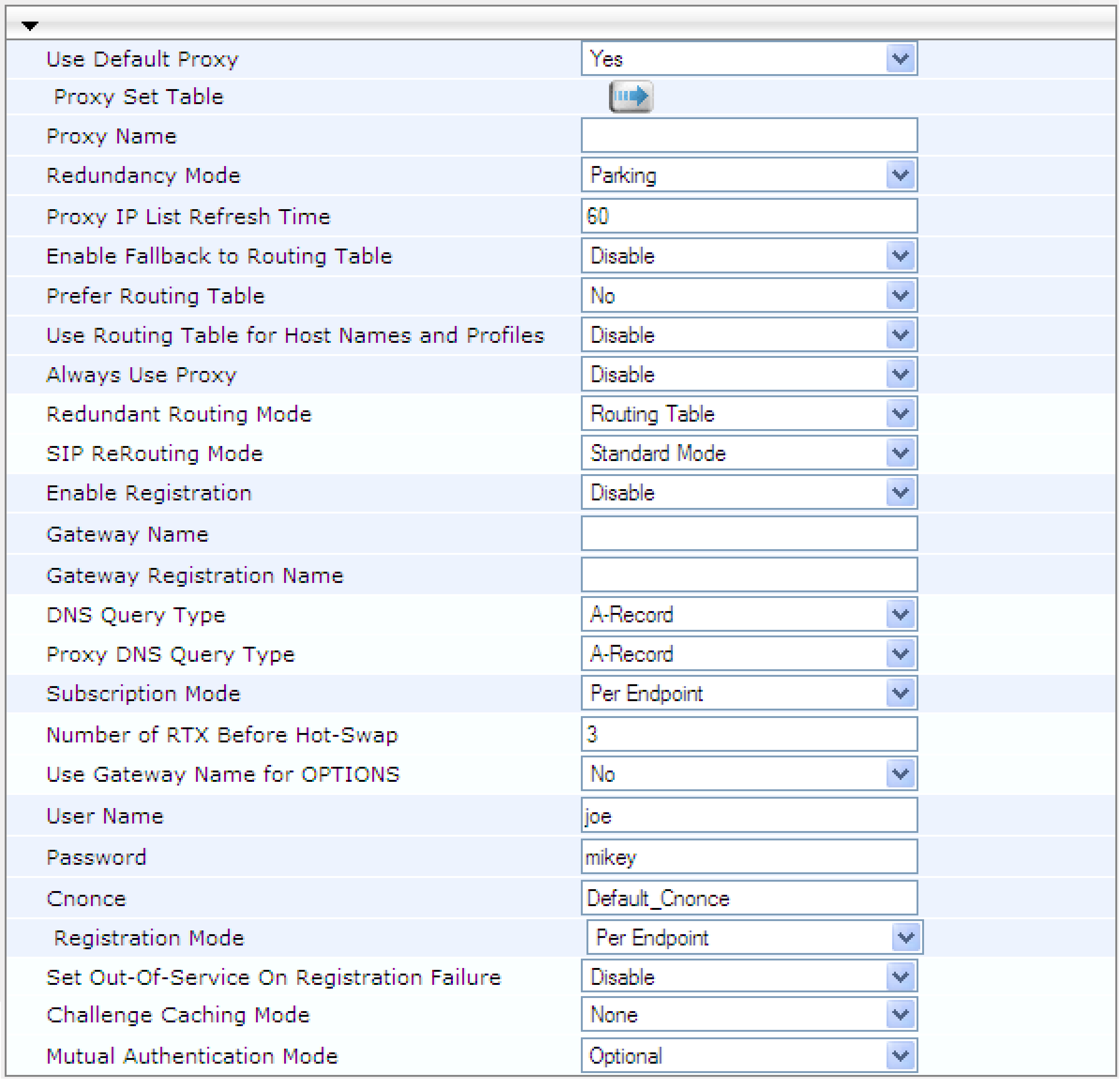
19.3 Configuring Proxy and Registration Parameters

The Proxy & Registration page allows you to configure the Proxy server and registration parameters. For a description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 380.

Note: To view the registration status of endpoints with a SIP Registrar/Proxy server, see "Viewing Registration Status" on page 341.

➢ To configure the Proxy and registration parameters:

1. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Proxy & Registration**).



Use Default Proxy	Yes
Proxy Set Table	➡
Proxy Name	
Redundancy Mode	Parking
Proxy IP List Refresh Time	60
Enable Fallback to Routing Table	Disable
Prefer Routing Table	No
Use Routing Table for Host Names and Profiles	Disable
Always Use Proxy	Disable
Redundant Routing Mode	Routing Table
SIP ReRouting Mode	Standard Mode
Enable Registration	Disable
Gateway Name	
Gateway Registration Name	
DNS Query Type	A-Record
Proxy DNS Query Type	A-Record
Subscription Mode	Per Endpoint
Number of RTX Before Hot-Swap	3
Use Gateway Name for OPTIONS	No
User Name	joe
Password	mikey
Cnonce	Default_Cnonce
Registration Mode	Per Endpoint
Set Out-Of-Service On Registration Failure	Disable
Challenge Caching Mode	None
Mutual Authentication Mode	Optional

2. Configure the parameters as required.
3. Click **Submit**.

➢ To register or un-register the device to a Proxy/Registrar:

- Click the **Register** button to register.
- Click **Un-Register** button to un-register.

Instead of registering the entire device, you can register specific entities as listed below by using the **Register** button located on the page in which these entities are configured:

- Accounts - Account table (see "Configuring Registration Accounts" on page 187)

Click the **Proxy Set Table** ➡ button to Open the Proxy Sets Table page to configure groups of proxy addresses. Alternatively, you can open this page from the **Proxy Sets Table** page item (see "Configuring Proxy Sets" on page 182 for a description of this page).

19.3.1 SIP Message Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261 standard. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then re-send the INVITE with a Proxy-Authorization header containing the credentials.

User agents, Redirect or Registrar servers typically use the SIP 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example shows the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to a Registrar/Proxy server for registration:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c17940
To: <sip: 122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Sip-Gateway/Mediant Software E-SBC/v.6.60.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
```

2. Upon receipt of this request, the Registrar/Proxy returns a 401 Unauthorized response:

```
SIP/2.0 401 Unauthorized
Via: SIP/2.0/UDP 10.2.1.200
From: <sip:122@10.2.2.222 >;tag=1c17940
To: <sip:122@10.2.2.222 >
Call-ID: 634293194@10.1.1.200
Cseq: 1 REGISTER
Date: Mon, 30 Jul 2012 15:33:54 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
WWW-Authenticate: Digest realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
stale=FALSE,
algorithm=MD5
```

3. According to the sub-header present in the WWW-Authenticate header, the correct REGISTER request is created.
4. Since the algorithm is MD5:
   - The username is equal to the endpoint phone number "122".
   - The realm return by the proxy is "audiocodes.com".
   - The password from the *ini* file is "AudioCodes".
   - The equation to be evaluated is "122:audiocodes.com:AudioCodes". According to the RFC, this part is called A1.
   - The MD5 algorithm is run on this equation and stored for future usage.
   - The result is "a8f17d4b41ab8dab6c95d3c14e34a9e1".
5. The par called A2 needs to be evaluated:
   - The method type is "REGISTER".

- Using SIP protocol "sip".
- Proxy IP from *ini* file is "10.2.2.222".
- The equation to be evaluated is "REGISTER:sip:10.2.2.222".
- The MD5 algorithm is run on this equation and stored for future usage.
- The result is "a9a031cfddcb10d91c8e7b4926086f7e".

6. Final stage:
   - A1 result: The nonce from the proxy response is "11432d6bce58ddf02e3b5e1c77c010d2".
   - A2 result: The equation to be evaluated is "A1:11432d6bce58ddf02e3b5e1c77c010d2:A2".
   - The MD5 algorithm is run on this equation. The outcome of the calculation is the response needed by the device to register with the Proxy.
   - The response is "b9c45d0234a5abf5ddf5c704029b38cf".

   At this time, a new REGISTER request is issued with the following response:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Server: Audiocodes-Sip-Gateway/Mediant Software E-SBC/v.6.60.010.006
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200:
Expires:3600
Authorization: Digest, username: 122,
realm="audiocodes.com",
nonce="11432d6bce58ddf02e3b5e1c77c010d2",
uri="10.2.2.222",
response="b9c45d0234a5abf5ddf5c704029b38cf"
```

7. Upon receiving this request and if accepted by the Proxy, the Proxy returns a 200 OK response, completing the registration transaction:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c23940
To: <sip: 122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Cseq: 1 REGISTER
Date: Thu, 26 Jul 2012 09:34:42 GMT
Server: Columbia-SIP-Server/1.17
Content-Length: 0
Contact: <sip:122@10.1.1.200>; expires="Thu, 26 Jul 2012 10:34:42 GMT";
action=proxy; q=1.00
Contact: <122@10.1.1.200:>; expires="Tue, 19 Jan 2038 03:14:07 GMT"; action=proxy;
q=0.00
Expires: Thu, 26 Jul 2012 10:34:42 GMT
```

19.4 Configuring SIP Message Manipulation

The Message Manipulations table lets you configure up to 200 Message Manipulation rules. A Message Manipulation rule defines a manipulation sequence for SIP messages. SIP message manipulation enables the normalization of SIP messaging fields between communicating network segments. For example, it allows service providers to design their own policies on the SIP messaging fields that must be present before a SIP call enters their network. Similarly, enterprises and small businesses may have policies for the information that can enter or leave their networks for policy or security reasons from a service provider. SIP message manipulations can also be implemented to resolve incompatibilities between SIP devices inside the enterprise network.

Each Message Manipulation rule is configured with a Manipulation Set ID. You can create groups (sets) of Message Manipulation rules by assigning each of the relevant Message Manipulation rules to the same Manipulation Set ID.

The Manipulation Set ID is used to assign the rules to calls:

To apply your Message Manipulation rule(s) to specific calls (inbound and/or outbound SIP messages), you need to assign a Manipulation Set ID to the relevant IP Group in the IP Group table (see Configuring IP Groups on page 174).

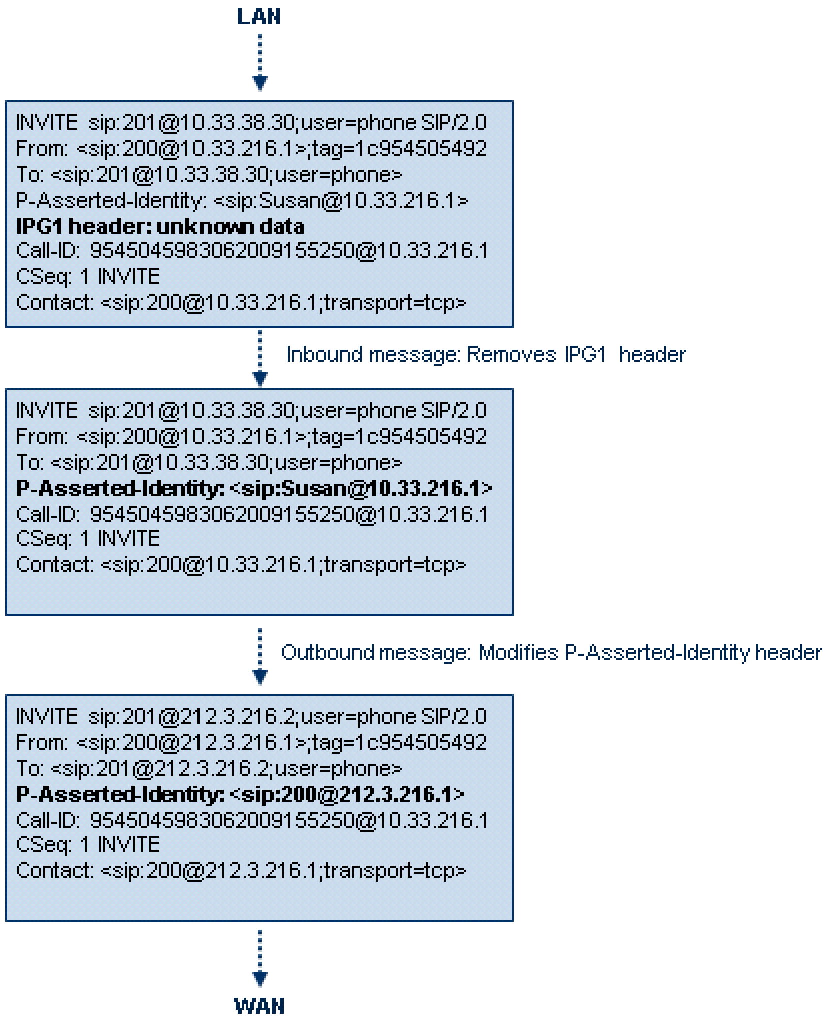
The device also supports a built-in SIP message normalization feature that can be enabled per Message Manipulation rule. The normalization feature removes unknown SIP message elements before forwarding the message. These elements can include SIP headers, SIP header parameters, and SDP body fields.

The SIP message manipulation feature supports the following:

- Manipulation on SIP message type (Method, Request/Response, and Response type)
- Addition of new SIP headers
- Removal of SIP headers ("black list")
- Modification of SIP header components such as values, header values (e.g., URI value of the P-Asserted-Identity header can be copied to the From header), call's parameter values
- Deletion of SIP body (e.g., if a message body is not supported at the destination network this body is removed)
- Translating one SIP response code to another
- Topology hiding (generally present in SIP headers such as Via, Record Route, Route and Service-Route).
- Configurable identity hiding (information related to identity of subscribers, for example, P-Asserted-Identity, Referred-By, Identity and Identity-Info)
- Apply conditions per rule - the condition can be on parts of the message or call’s parameters
- Multiple manipulation rules on the same SIP message

The figure below illustrates a SIP message manipulation example:

Figure 80: SIP Header Manipulation Example


LAN
Inbound message: Removes IPG1 header
Outbound message: Modifies P-Asserted-Identity header
WAN

```
INVITE sip:201@10.33.38.30;user=phone SIP/2.0
From: <sip:200@10.33.216.1>;tag=1c954505492
To: <sip:201@10.33.38.30;user=phone>
P-Asserted-Identity: <sip:Susan@10.33.216.1>
IPG1 header: unknown data
Call-ID: 9545045983062009155250@10.33.216.1
CSeq: 1 INVITE
Contact: <sip:200@10.33.216.1;transport=tcp>
```

```
INVITE sip:201@10.33.38.30;user=phone SIP/2.0
From: <sip:200@10.33.216.1>;tag=1c954505492
To: <sip:201@10.33.38.30;user=phone>
P-Asserted-Identity: <sip:Susan@10.33.216.1>
Call-ID: 9545045983062009155250@10.33.216.1
CSeq: 1 INVITE
Contact: <sip:200@10.33.216.1;transport=tcp>
```

```
INVITE sip:201@212.3.216.2;user=phone SIP/2.0
From: <sip:200@212.3.216.1>;tag=1c954505492
To: <sip:201@212.3.216.2;user=phone>
P-Asserted-Identity: <sip:200@212.3.216.1>
Call-ID: 9545045983062009155250@10.33.216.1
CSeq: 1 INVITE
Contact: <sip:200@212.3.216.1;transport=tcp>
```

Notes:

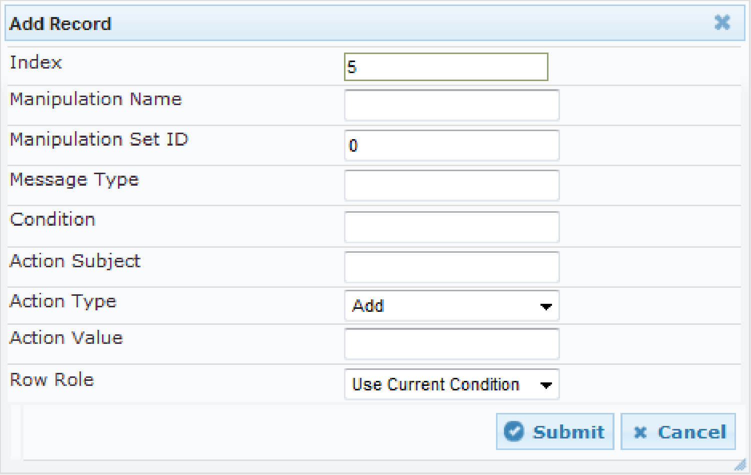
- For a detailed description of the syntax used for configuring Message Manipulation rules, refer to the *SIP Message Manipulations Quick Reference Guide*.
- Inbound message manipulation is done only after the Classification, inbound/outbound number manipulations, and routing processes.
- Each message can be manipulated twice - on the source leg and on the destination leg (i.e., source and destination IP Groups).
- Unknown SIP parts can only be added or removed.
- SIP manipulations do not allow you to remove or add mandatory SIP headers. They can only be modified and only on requests that initiate new dialogs. Mandatory SIP headers include To, From, Via, CSeq, Call-Id, and Max-Forwards.

The procedure below describes how to configure Message Manipulation rules in the Web interface. You can also configure Message Manipulation rules using the table ini file parameter, MessageManipulations or CLI command, configure voip > sbc manipulations message-manipulations.

➢ To configure SIP message manipulation rules:

1. Open the Message Manipulations page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Msg Policy & Manipulation** > **Message Manipulations**).
2. Click **Add**; the following dialog box appears:

Figure 81: Message Manipulations Table - Add Record Dialog Box



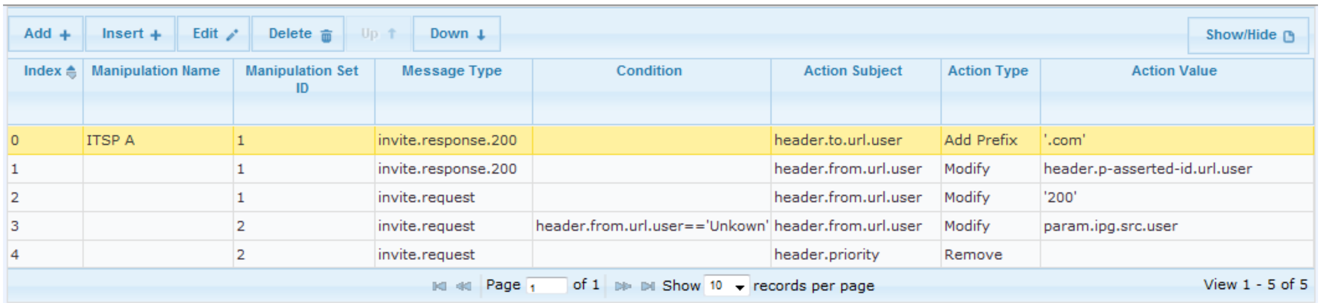
Add Record

Index	5
Manipulation Name	
Manipulation Set ID	0
Message Type	
Condition	
Action Subject	
Action Type	Add
Action Value	
Row Role	Use Current Condition

3. Configure a Message Manipulation rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

An example of configured message manipulation rules are shown in the figure below:

Figure 82: Message Manipulations Page



Index	Manipulation Name	Manipulation Set ID	Message Type	Condition	Action Subject	Action Type	Action Value
0	ITSP A	1	invite.response.200		header.to.url.user	Add Prefix	'.com'
1		1	invite.response.200		header.from.url.user	Modify	header.p-asserted-id.url.user
2		1	invite.request		header.from.url.user	Modify	'200'
3		2	invite.request	header.from.url.user=='Unkown'	header.from.url.user	Modify	param.ipg.src.user
4		2	invite.request		header.priority	Remove	

Page 1 of 1 Show 10 records per page View 1 - 5 of 5

- Index 0: Adds the suffix ".com" to the host part of the To header.
- Index 1: Changes the user part of the From header to the user part of the P-Asserted-ID.
- Index 2: Changes the user part of the SIP From header to "200".
- Index 3: If the user part of the From header equals "unknown", then it is changed according to the srcIPGroup call's parameter.
- Index 4: Removes the Priority header from an incoming INVITE message.

Table 64: Message Manipulations Parameter Descriptions

Parameter	Description
Index **[MessageManipulations_Index]**	Defines an index number for the new table record. **Note:** Each rule must be configured with a unique index.
Manipulation Name CLI: manipulation-name **[MessageManipulations_ManipulationName]**	Defines an arbitrary name to easily identify the Message Manipulation rule. The valid value is a string of up to 16 characters.
Manipulation Set ID CLI: manipulation-set-id **[MessageManipulations_ManSetID]**	Defines a Manipulation Set ID for the rule. You can define the same Manipulation Set ID for multiple rules to create a group of rules. The Manipulation Set ID is used to assign the manipulation rules to an IP Group (in the IP Group table) for inbound and/or outbound messages. The valid value is 0 to 19. The default is 0.
Matching Characteristics	
Message Type	Defines the SIP message type that you want to manipulate.

Parameter	Description
CLI: message-type **[MessageManipulations_MessageType]**	The valid value is a string (case-insensitive) denoting the SIP message. For example: ▪ Empty = rule applies to all messages ▪ Invite = rule applies to all INVITE requests and responses ▪ Invite.Request = rule applies to INVITE requests ▪ Invite.Response = rule applies to INVITE responses ▪ subscribe.response.2xx = rule applies to SUBSCRIBE confirmation responses **Note:** Currently, SIP 100 Trying messages cannot be manipulated.
Condition CLI: condition **[MessageManipulations_Condition]**	Defines the condition that must exist for the rule to apply. The valid value is a string (case-insensitive). For example: ▪ header.from.url.user== '100' (indicates that the user part of the From header must have the value "100") ▪ header.contact.param.expires > '3600' ▪ header.to.url.host contains 'domain' ▪ param.call.dst.user != '100'
Operation	
Action Subject CLI: action-subject **[MessageManipulations_ActionSubject]**	Defines the SIP header upon which the manipulation is performed. The valid value is a string (case-insensitive).
Action Type CLI: action-type **[MessageManipulations_ActionType]**	Defines the type of manipulation. ▪ **[0]** Add (default) = Adds new header/param/body (header or parameter elements). ▪ **[1]** Remove = Removes header/param/body (header or parameter elements). ▪ **[2]** Modify = Sets element to the new value (all element types). ▪ **[3]** Add Prefix = Adds value at the beginning of the string (string element only). ▪ **[4]** Add Suffix = Adds value at the end of the string (string element only). ▪ **[5]** Remove Suffix = Removes value from the end of the string (string element only). ▪ **[6]** Remove Prefix = Removes value from the beginning of the string (string element only). ▪ **[7]** Normalize = Removes unknown SIP message elements before forwarding the message.
Action Value CLI: action-value **[MessageManipulations_ActionValue]**	Defines a value that you want to use in the manipulation. The default value is a string (case-insensitive) in the following syntax: ▪ string/<message-element>/<call-param> + ▪ string/<message-element>/<call-param> For example: ▪ 'itsp.com' ▪ header.from.url.user ▪ param.call.dst.user ▪ param.call.dst.host + '.com' ▪ param.call.src.user + '<' + header.from.url.user + '@' + header.p-asserted-id.url.host + '>' **Note:** Only single quotation marks must be used.

Parameter	Description
Row Role CLI: row-role **[MessageManipulations_RowRole]**	Determines which condition must be used for the rule of this table row. ▪ **[0]** Use Current Condition = The condition entered in this row must be matched in order to perform the defined action (default). ▪ **[1]** Use Previous Condition = The condition of the rule configured directly above this row must be used in order to perform the defined action. This option allows you to configure multiple actions for the same condition. **Note:** When multiple manipulations rules apply to the same header, the next rule applies to the result string of the previous rule.

19.5 Configuring SIP Message Policy Rules

The Message Policy table lets you configure up to 20 SIP Message Policy rules. SIP Message Policy rules are used to block (blacklist) unwanted incoming SIP messages or permit (whitelist) receipt of desired SIP messages. You can configure legal and illegal characteristics of a SIP message. This feature is helpful against VoIP fuzzing (also known as robustness testing), which sends different types of packets to its "victims" for finding bugs and vulnerabilities. For example, the attacker might try sending a SIP message containing either an oversized parameter or too many occurrences of a parameter.

You can apply Message Policy rules globally (default) or per signaling domain by assigning it to a SIP interface (see "Configuring SIP Interfaces" on page 171).

Each Message Policy rule can be configured with the following:

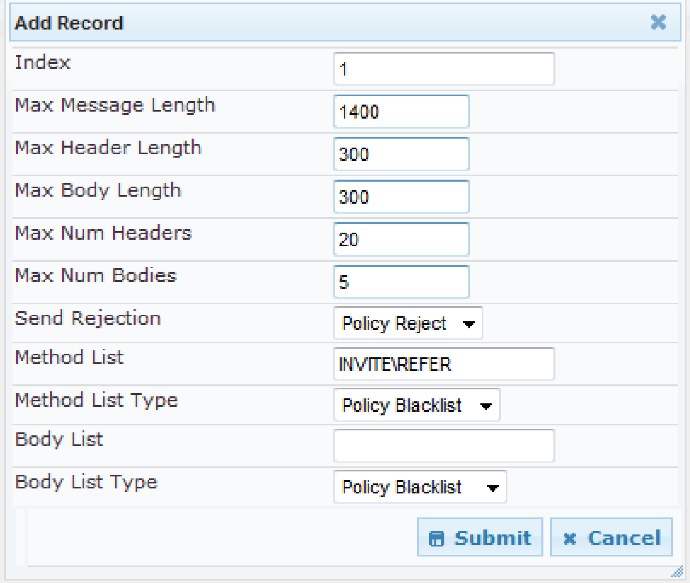
- Maximum message length
- Maximum header length
- Maximum message body length
- Maximum number of headers
- Maximum number of bodies
- Option to send 400 "Bad Request" response if message request is rejected
- Blacklist and whitelist for defined methods (e.g., INVITE)
- Blacklist and whitelist for defined bodies

The procedure below describes how to configure Message Policy rules in the Web interface. You can also configure Message Policy rules using the table ini file parameter, MessagePolicy or the CLI command, configure voip > sbc message-policy.

➢ To configure SIP Message Policy rules:

1. Open the Message Policy Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Msg Policy & Manipulation** > **Message Policy Table**).
2. Click **Add**; the following dialog box appears:

Figure 83: Message Policy Table - Add Record Dialog Box



Add Record

Index	1
Max Message Length	1400
Max Header Length	300
Max Body Length	300
Max Num Headers	20
Max Num Bodies	5
Send Rejection	Policy Reject
Method List	INVITE\REFER
Method List Type	Policy Blacklist
Body List	
Body List Type	Policy Blacklist

The figure above shows a configuration example where the Message Policy rule limits SIP messages to 32,768 characters, headers to 256 characters, bodies to 512 characters, number of headers to 16, and only permits two bodies. Invalid requests are rejected. Only INVITE and BYE requests are permitted and there are no restrictions on bodies.

3. Configure a Message Policy rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 65: Message Policy Table Parameter Descriptions

Parameter	Description
Index **[MessagePolicy_Index]**	Defines an index number for the new table record.
Max Message Length CLI: max-message-length **[MessagePolicy_MaxMessageLength]**	Defines the maximum SIP message length. The valid value is up to 32,768 characters. The default is 32,768.
Max Header Length CLI: max-header-length **[MessagePolicy_MaxHeaderLength]**	Defines the maximum SIP header length. The valid value is up to 512 characters. The default is 512.
Max Body Length CLI: max-body-length **[MessagePolicy_MaxBodyLength]**	Defines the maximum SIP message body length. This is the value of the Content-Length header. The valid value is up to 1,024 characters. The default is 1,024.
Max Num Headers CLI: max-num-headers **[MessagePolicy_MaxNumHeaders]**	Defines the maximum number of SIP headers. The valid value is any number up to 32. The default is 32. **Note:** The device supports up to 20 SIP Record-Route headers that can be received in a SIP INVITE request or 200 OK response. If it receives more than this, it responds with a SIP 513 'Message Too Large' response.
Max Num Bodies CLI: max-num-bodies **[MessagePolicy_MaxNumBodies]**	Defines the maximum number of bodies (e.g., SDP) in the SIP message. The valid value is any number up to 8. The default is 8.
Send Rejection CLI: send-rejection **[MessagePolicy_SendRejection]**	Determines whether the device sends a 400 "Bad Request" response if a message request is rejected. ▪ **[0]** Policy Reject = (Default) If the message is a request, then the device sends a response to reject the request. ▪ **[1]** Policy Drop = The device ignores the message without sending any response.

Parameter	Description
Method List CLI: method-list **[MessagePolicy_MethodList]**	Defines the SIP methods (e.g., INVITE\BYE) to which the rule applies. The syntax for entering the methods is as follows: ▪ Methods must be separated by a backslash (\). ▪ The entered value is not case sensitive.
Method List Type CLI: method-list-type **[MessagePolicy_MethodListType]**	Determines the policy for the SIP methods. ▪ **[0]** Policy Blacklist = The specified methods (in the 'Method List' field) are rejected by the policy. ▪ **[1]** Policy Whitelist = (Default) The specified methods (in the 'Method List' field) are allowed by the policy.
Body List CLI: body-list **[MessagePolicy_BodyList]**	Defines the SIP body (i.e., value of the Content-Type header) to which the rule applies.
Body List Type CLI: body-list-type **[MessagePolicy_BodyListType]**	Determines the policy for the defined SIP body. ▪ **[0]** Policy Blacklist =The specified SIP body (in the 'Body List' field) is rejected by the policy. ▪ **[1]** Policy Whitelist = (Default) The specified SIP body (in the 'Body List' field) is allowed by the policy.

20 Coders and Profiles

This section describes configuration of the coders and SIP profiles parameters.

20.1 Configuring IP Profiles

The IP Profile Settings table lets you configure up to 40 IP Profiles. An IP Profile is a set of parameters with user-defined settings relating to signaling (e.g., SIP message terminations such as REFER) and media (e.g., coder type). An IP Profile can later be assigned to specific IP calls (inbound and/or outbound). Thus, IP Profiles provide high-level adaptation when the device interworks between different IP entities, each of which may require different handling by the device. For example, if a specific IP entity uses the G.711 coder only, you can configure an IP Profile with G.711 for this IP entity.

To use your IP Profile for specific calls, you need to assign it to any of the following:

- IP Groups - see "Configuring IP Groups" on page 174

Many of the parameters in the IP Profile table have a corresponding "global" parameter. For calls that are not associated with any IP Profile, the settings of the "global" parameters are applied.

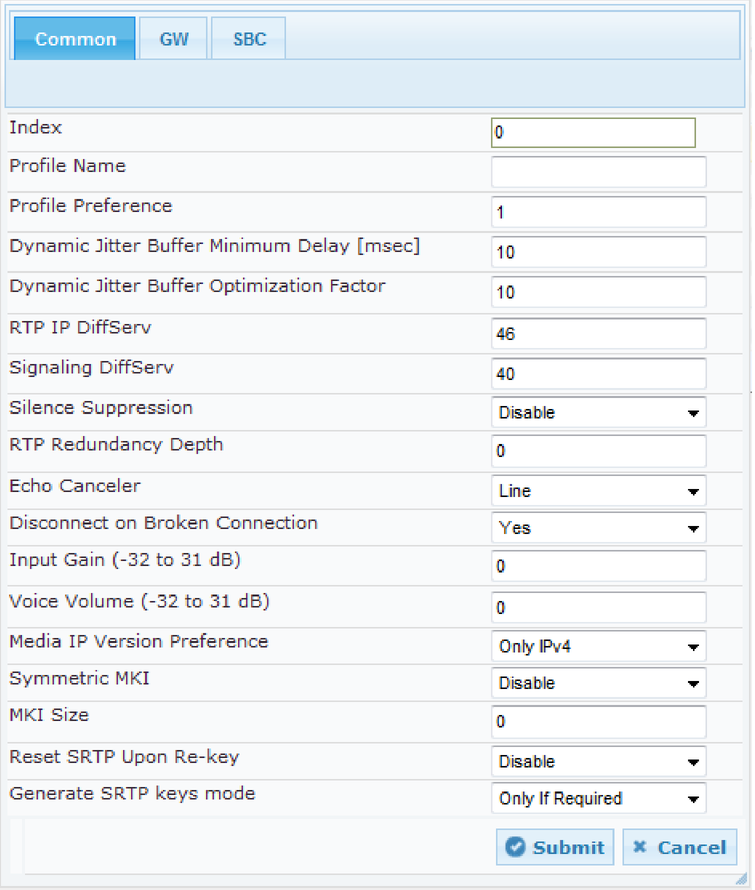
> **Note:** IP Profiles can also be implemented when using a Proxy server (when the AlwaysUseRouteTable parameter is set to 1).

The procedure below describes how to configure IP Profiles in the Web interface. You can also configure IP Profiles using the table ini file parameter, IPProfile or the CLI command, configure voip > coders-and-profiles ip-profile.

➢ To configure an IP Profile:

1. Open the IP Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **IP Profile Settings**).
2. Click **Add**; the following dialog box appears:

Figure 84: IP Profile Table - Add Record



Index	0
Profile Name	
Profile Preference	1
Dynamic Jitter Buffer Minimum Delay [msec]	10
Dynamic Jitter Buffer Optimization Factor	10
RTP IP DiffServ	46
Signaling DiffServ	40
Silence Suppression	Disable
RTP Redundancy Depth	0
Echo Canceler	Line
Disconnect on Broken Connection	Yes
Input Gain (-32 to 31 dB)	0
Voice Volume (-32 to 31 dB)	0
Media IP Version Preference	Only IPv4
Symmetric MKI	Disable
MKI Size	0
Reset SRTP Upon Re-key	Disable
Generate SRTP keys mode	Only If Required

3. Configure an IP Profile according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 66: IP Profile Settings Table Parameter Descriptions

Parameter	Description
Common	
Web: Index **[IpProfile_Index]**	Defines an index number for the new table record.
Web: Profile Name CLI: profile-name **[IpProfile_ProfileName]**	Defines an arbitrary name to easily identify the IP Profile. The valid value is a string of up to 20 characters.
Web: RTP IP DiffServ CLI: rtp-ip-diffserv **[IpProfile_IPDiffServ]**	Defines the DiffServ value for Premium Media class of service (CoS) content. The valid range is 0 to 63. The default is 46. **Note:** The corresponding global parameter is PremiumServiceClassMediaDiffServ.
Web: Signaling DiffServ CLI: signaling-diffserv **[IpProfile_SigIPDiffServ]**	Defines the DiffServ value for Premium Control CoS content (Call Control applications). The valid range is 0 to 63. The default is 40. **Note:** The corresponding global parameter is PremiumServiceClassControlDiffServ.
Web: RTP Redundancy Depth CLI: rtp-redundancy-depth **[IpProfile_RTPRedundancyDepth]**	Enables the device to generate RFC 2198 redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver's end from the redundant data that arrives in subsequent packets. This is required, for example, in wireless networks where a high percentage (up to 50%) of packet loss can be experienced.

Parameter	Description
	▪ **[0]** 0 = (Default) Disable. ▪ **[1]** 1 = Enable - previous voice payload packet is added to current packet. **Notes:** ▪ When enabled, you can configure the payload type, using the RFC2198PayloadType parameter. ▪ The RTP redundancy dynamic payload type can be included in the SDP, by using the EnableRTPRedundancyNegotiation parameter. ▪ The corresponding global parameter is RTPRedundancyDepth.
Web: Disconnect on Broken Connection CLI: disconnect-on-broken-connection **[IpProfile_DisconnectOnBrokenConnection]**	Enables the device to release the call if RTP packets are not received within a user-defined timeout, configured by the BrokenConnectionEventTimeout parameter. ▪ **[0]** No ▪ **[1]** Yes (default) **Notes:** ▪ This feature is applicable only if the RTP session is used without Silence Compression. If Silence Compression is enabled, the device doesn't detect a broken RTP connection. ▪ During a call, if the source IP address (from where the RTP packets are received) is changed without notifying the device, the device filters these RTP packets. To overcome this, set the DisconnectOnBrokenConnection parameter to 0; the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address. ▪ The corresponding global parameter is DisconnectOnBrokenConnection.
Web: Media IP Version Preference CLI: media-ip-version-preference [IpProfile_MediaIPVersionPreference]	Defines the preferred RTP media IP addressing version for outgoing SIP calls. This is indicated in the "c=" field (Connection Information) of the SDP. ▪ [0] Only IPv4 = (Default) SDP offer includes only IPv4 media IP addresses. ▪ [1] Only IPv6 = SDP offer includes only IPv6 media IP addresses. ▪ [2] Prefer IPv4 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first media is IPv4. ▪ [3] Prefer IPv6 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first media is IPv6. **Notes:** ▪ This parameter is applicable only when the device offers an SDP. ▪ The IP addressing version is determined according to the first SDP "m=" field. ▪ The corresponding global parameter is MediaIPVersionPreference.

Web: Symmetric MKI
CLI: enable-symmetric-mki
[IpProfile_EnableSymmetricMKI]
	Enables symmetric MKI negotiation.
	▪ **[0]** Disable = (Default) The device includes the MKI in its SIP 200 OK response according to the SRTPTxPacketMKISize parameter (if set to 0, it is not included; if set to any other value, it is included with this value).
	▪ **[1]** Enable = The answer crypto line contains (or excludes) an MKI value according to the selected crypto line in the offer. For example, assume that the device receives an INVITE containing the following two crypto lines in SDP:
	`a=crypto:2 AES CM 128 HMAC SHA1 80 inline:TAaxNnQt8/qLQMnDuG4vxYfWl6K7eBK/ufk04pR4|2^31|1:1`
	`a=crypto:3 AES CM 128 HMAC SHA1 80 inline:bnuYZnMxSfUiGitviWJZmzr7OF3AiRO0l5Vnh0kH|2^31`
	▪ The first crypto line includes the MKI parameter "1:1". In the 200 OK response, the device selects one of the crypto lines (i.e., '2' or '3'). Typically, it

Parameter	Description
	selects the first line that supports the crypto suite. However, for SRTP-to-SRTP in SBC sessions, it can be determined by the remote side on the outgoing leg. If the device selects crypto line '2', it includes the MKI parameter in its answer SDP, for example: `a=crypto:2 AES CM 128 HMAC SHA1 80 inline:R1VyA1xV/qwBjkEklu4kSJyl3wCtYeZLq1/QFuxw|2^31|1:1` ▪ If the device selects a crypto line that does not contain the MKI parameter, then the MKI parameter is not included in the crypto line in the SDP answer (even if the SRTPTxPacketMKISize parameter is set to any value other than 0). **Note:** The corresponding global parameter is EnableSymmetricMKI.
Web: MKI Size CLI: mki-size **[IpProfile_MKISize]**	Defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. The valid value is 0 to 4. The default is 0 (i.e., new keys are generated without MKI). **Notes:** ▪ The device can forward MKI size as is for SRTP-to-SRTP flows or override the MKI size during negotiation. This can be done on the inbound or outbound leg. ▪ The corresponding global parameter is SRTPTxPacketMKISize.
Web: Reset SRTP Upon Re-key CLI: reset-srtp-upon-re-key **[IpProfile_ResetSRTPStateUponRekey]**	Enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. This feature ensures that the roll-over counter (ROC), one of the parameters used in the SRTP encryption/decryption process of the SRTP packets, is synchronized on both sides for transmit and receive packets. ▪ **[0]** Disable = (Default) ROC is not reset on the device side. ▪ **[1]** Enable = If the session expires causing a session refresh through a re-INVITE, the device or server generates a new key and the device resets the ROC index (and other SRTP fields) as done by the server, resulting in a synchronized SRTP. **Notes:** ▪ If this feature is disabled and the server resets the ROC upon a re-key generation, one-way voice may occur. ▪ The corresponding global parameter is ResetSRTPStateUponRekey.
Generate SRTP keys mode CLI: generate-srtp-keys **[IpProfile_GenerateSRTPKeys]**	Enables the device to generate a new SRTP key upon receipt of a re-INVITE with this SIP entity. ▪ **[0]** Only If Required= (Default) The device generates an SRTP key only if necessary. ▪ **[1]** Always = The device always generates a new SRTP key.
SBC	
Allowed Media Types CLI: sbc-allowed-media-types [IPProfile_SBCAllowedMediaTypes]	Defines media types permitted for this SIP entity. The media type appears in the SDP 'm=' line (e.g., 'm=audio'). The device permits only media types that appear in both the SDP offer and this configured list. If no common media types exist between the SDP offer and this list, the device drops the call. The valid value is a string of up to 64 characters. To configure multiple media types, separate the strings with a comma, e.g., "media, audio" (without quotes). By default, no media types are configured (i.e., all media types are permitted).
Web: Allowed Coders Group ID CLI: sbc-allowed-coders-group-id [IpProfile_SBCAllowedCodersGroupI	Assigns an Allowed Coders Group to this SIP entity. This defines audio (voice) coders that can be used for this SIP entity. To configure Allowed Coders Groups, see Configuring Allowed Audio Coder

Parameter	Description
D]	Groups on page 247. For a description of the Allowed Coders feature, see "Restricting Coders" on page 223.
Web: Allowed Video Coders Group ID CLI: sbc-allowed-video-coders-group-id [IPProfile_SBCAllowedVideoCodersGroupID]	Assigns an Allowed Video Coders Group to this SIP entity. This defines permitted video coders when forwarding video streams to the SIP entity. The video coders are listed in the "video" media type in the SDP (i.e., 'm=video' line). For this SIP entity, the device uses only video coders that appear in both the SDP offer and the Allowed Video Coders Group ID. By default, no Allowed Video Coders Group is assigned (i.e., all video coders are allowed). To configure Allowed Video Coders Groups, see Configuring Allowed Video Coder Groups on page 248.
Web: Allowed Coders Mode CLI: sbc-allowed-coders-mode [IpProfile_SBCAllowedCodersMode]	Defines the mode of the Allowed Coders feature for this SIP entity. ▪ [0] Restriction = In the incoming SDP offer, the device uses only Allowed coders; the rest are removed from the SDP offer (i.e., only coders common between those in the received SDP offer and the Allowed coders are used). ▪ [1] Preference = The device re-arranges the priority (order) of the coders in the incoming SDP offer according to their order of appearance in the Allowed Coders Group or Allowed Video Coders tables. The coders received in the SDP offer are listed adter the Allowed coders. ▪ [2] Restriction and Preference = Performs both Restriction and Preference. **Notes:** ▪ This parameter is applicable only if Allowed coders are assigned to the IP Profile (using the 'Allowed Coders Group ID' or 'Allowed Video Coders Group ID' parameters). ▪ For more information on the Allowed Coders feature, see Restricting Coders on page 223.
Web: SBC Media Security Behavior CLI: sbc-media-security-behaviour [IpProfile_SBCMediaSecurityBehaviour]	Defines the handling of RTP and SRTP for this SIP entity. ▪ [0] As is = (Default) No special handling for RTP\SRTP is done. ▪ [1] SRTP = SBC legs negotiate only SRTP media lines, and RTP media lines are removed from the incoming SDP offer\answer. ▪ [2] RTP = SBC legs negotiate only RTP media lines, and SRTP media lines are removed from the incoming offer\answer. ▪ [3] Both = Each offer\answer is extended (if not already) to two media lines - one RTP and the other SRTP. If two SBC legs (after offer\answer negotiation) use different security types (i.e., one RTP and the other SRTP), the device performs RTP-SRTP transcoding. To transcode between RTP and SRTP, the following prerequisites must be met: ▪ At least one supported SDP "crypto" attribute and parameters. ▪ EnableMediaSecurity must be set to 1. If one of the above transcoding prerequisites is not met, then: ▪ any value other than "As is" is discarded. ▪ if the incoming offer is SRTP, force transcoding, coder transcoding, and DTMF extensions are not applied.
Web: P-Asserted-Identity CLI: sbc-assert-identity [IpProfile_SBCAssertIdentity]	Defines the device's handling of the SIP P-Asserted-Identity header for this SIP entity. This header indicates how the outgoing SIP message asserts identity. ▪ [0] As Is = (Default) P-Asserted Identity header is not affected and the device uses the same P-Asserted-Identity header (if present) in the

Parameter	Description
	incoming message for the outgoing message. ▪ [1] Add = Adds a P-Asserted-Identity header. The header's values are taken from the source URL. ▪ [2] Remove = Removes the P-Asserted-Identity header. **Notes:** ▪ This parameter affects only the initial INVITE request. ▪ The corresponding global parameter is SBCAssertIdentity.
Web: Diversion Mode CLI: sbc-diversion-mode [IpProfile_SBCDiversionMode]	Defines the device's handling of the SIP Diversion header for this SIP entity. For more information on interworking of the History-Info and Diversion headers, see Interworking SIP Diversion and History-Info Headers on page 228. ▪ [0] As Is = (Default) Diversion header is not handled. ▪ [1] Add = History-Info header is converted to a Diversion header. ▪ [2] Remove = Removes the Diversion header and the conversion to the History-Info header depends on the settings of the SBCHistoryInfoMode parameter.
Web: History-Info Mode CLI: sbc-history-info-mode [IpProfile_SBCHistoryInfoMode]	Defines the device's handling of the SIP History-Info header for this SIP entity. For more information on interworking of the History-Info and Diversion headers, see Interworking SIP Diversion and History-Info Headers on page 228. ▪ [0] As Is = (Default) History-Info header is not handled. ▪ [1] Add = Diversion header is converted to a History-Info header. ▪ [2] Remove = History-Info header is removed from the SIP dialog and the conversion to the Diversion header depends on the settings of the SBCDiversionMode parameter.
Web: PRACK Mode CLI: sbc-prack-mode [IpProfile_SbcPrackMode]	Defines the device's handling of SIP PRACK messages for this SIP entity. ▪ [1] Optional = PRACK is optional. If required, the device performs the PRACK process on behalf of the SIP entity. ▪ [2] Mandatory = PRACK is required for this SIP entity. Calls from endpoints that do not support PRACK are rejected. Calls destined to these endpoints are also required to support PRACK. ▪ [3] Transparent (default) = The device does not intervene with the PRACK process and forwards the request as is.
Web: Session Expires Mode CLI: sbc-session-expires-mode [IpProfile_SBCSessionExpiresMode]	Defines the required session expires mode for this SIP entity. ▪ [0] Transparent = (Default) The device does not interfere with the session expires negotiation. ▪ [1] Observer = If the SIP Session-Expires header is present, the device does not interfere, but maintains an independent timer for each leg to monitor the session. If the session is not refreshed on time, the device disconnects the call. ▪ [2] Not Supported = The device does not allow a session timer with this SIP entity. ▪ [3] Supported = The device enables the session timer with this SIP entity. If the incoming SIP message does not include any session timers, the device adds the session timer information to the sent message. You can configure the value of the Session-Expires and Min-SE headers, using the SBCSessionExpires and SBCMinSE parameters, respectively.
Web: Remote Update Support CLI: sbc-rmt-update-supp [IpProfile_SBCRemoteUpdateSupport]	Defines whether this SIP entity supports the SIP UPDATE message. ▪ [0] Not Supported = UPDATE message is not supported. ▪ [1] Supported Only After Connect = UPDATE message is supported only after the call is connected. ▪ [2] Supported = (Default) UPDATE message is supported during call

Parameter	Description
	setup and after call establishment.
Web: Remote Re-INVITE Support CLI: sbc-rmt-re-invite-supp [IpProfile_SBCRemoteReinviteSupport]	Defines whether the destination UA of the re-INVITE request supports re-INVITE messages and if so, whether it supports re-INVITE with or without SDP. ▪ [0] Not Supported = re-INVITE is not supported and the device does not forward re-INVITE requests. The device sends a SIP response to the re-INVITE request, which can either be a success or a failure, depending on whether the device can bridge the media between the endpoints. ▪ [1] Supported only with SDP = re-INVITE is supported, but only with SDP. If the incoming re-INVITE arrives without SDP, the device creates an SDP and adds it to the outgoing re-INVITE. ▪ [2] Supported = (Default) re-INVITE is supported with or without SDP.
Web: Remote Delayed Offer Support CLI: sbc-rmt-delayed-offer [IpProfile_SBCRemoteDelayedOfferSupport]	Defines whether the remote endpoint supports delayed offer (i.e., initial INVITEs without an SDP offer). ▪ [0] Not Supported = Initial INVITE requests without SDP are not supported. ▪ [1] Supported = (Default) Initial INVITE requests without SDP are supported. **Note:** For this parameter to function, you need to configure a valid Extension Coders Group ID for IP Profiles that do not support delayed offer.
Web: Remote REFER Behavior CLI: sbc-rmt-refer-behavior [IpProfile_SBCRemoteReferBehavior]	Defines the device's handling of REFER requests for this SIP entity. ▪ [0] Regular = (Default) Refer-To header is unchanged and the device forwards the REFER as is. ▪ [1] Database URL = Changes the Refer-To header so that the re-routed INVITE is sent through the SBC: **a.** Before forwarding the REFER request, the device changes the host part to the device's IP address and adds a special prefix ("T~&R_") to the Contact user part. **b.** The incoming INVITE is identified as a REFER-resultant INVITE according to this special prefix. **c.** The device replaces the host part in the Request-URI with the host from the REFER contact. The special prefix remains in the user part for regular classification, manipulation, and routing. The special prefix can also be used for specific routing rules for REFER-resultant INVITEs. **d.** The special prefix is removed before the resultant INVITE is sent to the destination. ▪ [2] IP Group Name = Sets the host part in the REFER message to the name defined for the IP Group (in the IP Group table). ▪ [3] Handle Locally = Handles the incoming REFER request itself without forwarding the REFER. The device generates a new INVITE to the alternative destination according to the rules in the IP-to-IP Routing table (the 'Call Trigger' field must be set to REFER). **Note:** The corresponding global parameter is SBCReferBehavior.
Web: Remote 3xx Behavior CLI: sbc-rmt-3xx-behavior [IpProfile_SBCRemote3xxBehavior]	Defines the device's handling of SIP 3xx redirect responses for this SIP entity. By default, the device's handling of SIP 3xx responses is to send the Contact header unchanged. However, some SIP entities may support different versions of the SIP 3xx standard while others may not even support SIP 3xx. When enabled, the device handles SIP redirections between different subnets (e.g., between LAN and WAN sides). This is required when the new address provided by the redirector (Redirect sever) may not be reachable by the far-end user (FEU) located in another subnet. For example, a far-end user (FEU) in the WAN sends a SIP request via the device to a Redirect

Parameter	Description
	server in the LAN, and the Redirect server replies with a SIP 3xx response to a PBX in the LAN in the Contact header. If the device sends this response as is (i.e., with the original Contact header), the FEU is unable to reach the new destination. ▪ [0] Transparent = (Default) The device forwards the received SIP 3xx response as is, without changing the Contact header (i.e.,transparent handling). ▪ [1] Database URL = The device changes the Contact header so that the re-route request is sent through the device. The device changes the URI in the Contact header of the received SIP 3xx response to its own URI and adds a special user prefix ("T~&R_"), which is then sent to the FEU. The FEU then sends a new INVITE to the device, which the device then sends to the correct destination. ▪ [2] Handle Locally = The device handles SIP 3xx responses on behalf of the dialog-initiating UA and retries the request (e.g., INVITE) using one or more alternative URIs included in the 3xx response. The device sends the new request to the alternative destination according to the IP-to-IP Routing table (the 'Call Trigger' field must be set to 3xx). **Notes:** ▪ When this parameter is changed from 1 to 0, new 3xx Contact headers remain unchanged. However, requests with the special prefix continue using the device's database to locate the new destination. ▪ Only one database entry is supported for the same host, port, and transport combination. For example, the following URLs cannot be distinguished by the device: ✓ sip:10.10.10.10:5060;transport=tcp;param=a ✓ sip:10.10.10.10:5060;transport=tcp;param=b ▪ The database entry expires two hours after the last use. ▪ The maximum number of destinations (i.e., database entries) is 50. ▪ The corresponding global parameter is SBC3xxBehavior.
Web: Remote Multiple 18x CLI: sbc-rmt-mltple-18x-supp [IpProfile_SBCRemoteMultiple18xSupport]	Defines whether multiple 18x responses including 180 Ringing, 181 Call is Being Forwarded, 182 Call Queued, and 183 Session Progress are forwarded to the caller, for this SIP entity. ▪ [0] Not Supported = Only the first 18x response is forwarded to the caller. ▪ [1] Supported = (Default) Multiple 18x responses are forwarded to the caller.
Web: Remote Early Media Response Type CLI: sbc-rmt-early-media-resp [IpProfile_SBCRemoteEarlyMediaResponseType]	Defines the SIP provisional response type - 180 or 183 - for forwarding early media to the caller, for this SIP entity. ▪ [0] Transparent = (Default) All early media response types are supported; the device forwards all responses as is (unchanged). ▪ [1] 180 = Early media is sent as 180 response only. ▪ [2] 183 = Early media is sent as 183 response only.
Web: Remote Early Media CLI: sbc-rmt-early-media-supp [IpProfile_SBCRemoteEarlyMediaSupport]	Defines whether the remote side can accept early media or not. ▪ [0] Not Supported = Early media is not supported. ▪ [1] Supported = (Default) Early media is supported.
Web: Enforce MKI Size CLI: sbc-enforce-mki-size [IpProfile_SBCEnforceMKISize]	Enables MKI length negotiation for SRTP-to-SRTP flows between SIP networks (i.e., IP Groups). This includes the capability of modifying the MKI length on the inbound or outbound SBC call leg for this SIP entity. ▪ [0] Don't enforce = (Default) Device forwards the MKI size as is. ▪ [1] Enforce = Device changes the MKI length according to the settings of the IP Profile parameter, MKISize.
Web: Remote Early Media RTP	Defines whether the destination UA sends RTP immediately after it sends

Parameter	Description
Behavior CLI: sbc-rmt-early-media-rtp [IpProfile_SBCRemoteEarlyMediaRTP]	18x response. - [0] Immediate = (Default) Remote client sends RTP immediately after it sends 18x response with early media. Device forwards 18x and RTP as is. - [1] Delayed = After sending 18x response, the remote client waits before sending RTP (e.g., Microsoft Lync environment). For the device's handling of this remote UA support, see Interworking SIP Early Media on page 230.
Web: Remote RFC 3960 Gateway Model Support CLI: sbc-rmt-rfc3960-supp [IpProfile_SBCRemoteSupportsRFC3960]	Defines whether the destination UA is capable of receiving 18x messages with delayed RTP. - [0] Not Supported = (Default) UA does not support receipt of 18x messages with delayed RTP. For the device's handling of this remote UA support, see Interworking SIP Early Media on page 230. - [1] Supported = UA is capable of receiving 18x messages with delayed RTP.
Web: Remote Can Play Ringback CLI: sbc-rmt-can-play-ringback [IpProfile_SBCRemoteCanPlayRingback]	Defines whether the destination UA can play a local ringback tone. - [0] No = UA does not support local ringback tone. The device sends 18x with delayed SDP to the UA. - [1] Yes = (Default) UA supports local ringback tone. For the device's handling of this remote UA support, see Interworking SIP Early Media on page 230.
Web: RFC 2833 DTMF Payload Type CLI: sbc-2833dtmf-payload [IpProfile_SBC2833DTMFPayloadType]	Defines the payload type of DTMF digits for this SIP entity. This enables the interworking of the DTMF payload type for RFC 2833 between different SBC call legs. For example, if two entities require different DTMF payload types, the SDP offer received by the device from one entity is forwarded to the destination entity with its payload type replaced with the configured payload type, and vice versa. The value range is 0 to 200. The default is 0 (i.e., the device forwards the received payload type as is).
Web: User Registration Time CLI: sbc-usr-reg-time [IpProfile_SBCUserRegistrationTime]	Defines the duration (in seconds) of the periodic registrations that occur between the users of this SIP entity and the device (the device responds with this value to the user). The valid range is 0 to 2,000,000 seconds. The default is 0. When set to 0, the device does not change the Expires header's value received in the user's REGISTER request. If no Expires header is received in the REGISTER message and this parameter is set to 0, the Expires header's value is set to 180 seconds, by default. **Note:** The corresponding global parameter is SBCUserRegistrationTime.
Web: Reliable Held Tone Source CLI: reliable-heldtone-source [IPProfile_ReliableHoldToneSource]	Enables the device to consider the received call-hold request (re-INVITE/UPDATE) with SDP containing 'a=sendonly', as genuine. - [0] No (default) = Even if the received SDP contains 'a=sendonly', the device plays a held tone to the held party. This is useful in cases where the initiator of the call hold does not support the generation of held tones. - [1] Yes = If the received SDP contains 'a=sendonly', the device does not play a held tone to the held party (and assumes that the initiator of the call hold plays the held tone). **Note:** The device plays a held tone only if the 'SBC Play Held Tone' parameter is set to Yes.
Web: Play Held Tone CLI: play-held-tone [IpProfile_SBCPlayHeldTone]	Enables the device to play a held tone to the held party. This is useful if the held party does not support playing a local held tone, or for IP entities initiating call hold that do not support the generation of held tones. - [0] No (default) - [1] Yes **Note:** If this parameter is set to Yes, the device plays the tone only if the

Parameter	Description
	'SBC Remote Hold Format' parameter is set to transparent, send-only, send only 0.0.0.0, or not supported.
Web: Remote Hold Format CLI: remote-hold-Format [IPProfile_SBCRemoteHoldFormat]	Defines the format of the SDP in the re-INVITE for call hold that the device sends to the held party. ▪ [0] Transparent = Device forwards SDP as is. ▪ [1] Send Only = Device sends SDP with 'a=sendonly'. ▪ [2] Send Only Zero ip = Device sends SDP with 'a=sendonly' and 'c=0.0.0.0'. ▪ [3] Inactive = Device sends SDP with 'a=inactive'. ▪ [4] Inactive Zero ip = Device sends SDP with 'a=inactive' and 'c=0.0.0.0'. ▪ [5] Not Supported = Used when remote side cannot identify a call-hold message. The device terminates the received call-hold message (re-INVITE / UPDATE) and sends a 200 OK to the initiator of the call hold. The device plays a held tone to the held party if the 'SBC Play Held Tone' parameter is set to Yes.
Web: Remote Replaces Behavior CLI: sbc-rmt-replaces-behavior [IpProfile_SBCRemoteReplacesBehavior]	Enables the device to handle an incoming INVITE with the Replaces header for this SIP entity (which does not support this header). The Replaces header is used to replace an existing SIP dialog with a new dialog such as in call transfer or call pickup ▪ [0] Transparent = (Default) SIP entity supports the Replaces header. The device forwards the received INVITE with the Replaces header as is to the SIP entity. ▪ [1] Handle Locally = The SIP entity does not support INVITE with the Replaces header. The device terminates the received INVITE with the Replaces header and establishes a new call between the SIP entity and the new call party. It then disconnects the call with the initial call party, by sending it a SIP BYE request. For example, assume that the device establishes a call between A and B. If B initiates a call transfer to C, the device receives an INVITE with the Replaces header from C. If A supports the Replaces header, the device simply forwards the INVITE as is to A; a new call is established between A and C and the call between A and B is disconnected. However, if A does not support the Replaces header, the device uses this feature to terminate the INVITE with Replaces header and handles the transfer for A. The device does this by connecting A to C, and disconnecting the call between A and B, by sending a SIP BYE request to B. Note that if media transcoding is required, the device sends an INVITE to C on behalf of A with a new SDP offer.
Web: SDP Ptime Answer CLI: sbc-sdp-ptime-ans [IpProfile_SBCSDPPtimeAnswer]	Defines the packetization time (ptime) of the coder in RTP packets for this SIP entity. This is useful when implementing transrating. ▪ [0] Remote Answer (Default) = Use ptime according to SDP answer. ▪ [1] Original Offer = Use ptime according to SDP offer. ▪ [2] Preferred Value= Use preferred ptime for negotiation, if configured by the 'Preferred Ptime' parameter.
Web: Preferred Ptime CLI: sbc-preferred-ptime [IpProfile_SBCPreferredPTime]	Defines the packetization time (in msec) for this SIP entity if the 'SBC SDP Ptime Answer' parameter is set to Preferred Value. The valid range is 0 to 200. The default is 0 (i.e., preferred ptime is not used).
Web: Use Silence Suppression CLI: sbc-use-silence-supp [IpProfile_SBCUseSilenceSupp]	Defines silence suppression support for this SIP entity. ▪ [0] Transparent (default) = Forward as is. ▪ [1] Add = Enable silence suppression for each relevant coder listed in the SDP. ▪ [2] Remove = Disable silence suppression for each relevant coder listed in the SDP.

Parameter	Description
Web: Play RBT To Transferee CLI: sbc-play-rbt-to-xferee [IpProfile_SBCPlayRBTToTransferee]	Enables the device to play a ringback tone to the transferred party (transferee) during a blind call transfer, for this SIP entity (which does not support such a tone generation during call transfer). The ringback tone indicates to the transferee of the ringing of the transfer target (to where the transferee is being transferred). ▪ [0] No (Default) ▪ [1] Yes Typically, the transferee hears a ringback tone only if the transfer target sends it early media. However, if the transferee is put on-hold before being transferred, no ringback tone is heard. When this feature is enabled, the device generates a ringback tone to the transferee during call transfer in the following scenarios: ▪ Transfer target sends a SIP 180 (Ringing) to the device. ▪ For non-blind transfer, if the call is transferred while the transfer target is ringing and no early media occurs. ▪ The 'Remote Early Media RTP Behavior parameter is set to Delayed (used in the Lync environment), and transfer target sends a 183 Session progress with SDP offer. If early media from the transfer target has already been detected, the transferee receives RTP stream from the transfer target. If it has not been detected, the device generates a ringback tone to the transferee and stops the tone generation once RTP has been detected from the transfer target. For any of these scenarios, if the transferee is put on-hold by the transferor, the device retrieves the transferee from hold, sends a re-INVITE if necessary, and then plays the ringback tone. **Note:** For the device to play the ringback tone, it must be loaded with a Prerecorded Tones (PRT) file. For more information, see Prerecorded Tones File on page 302.
Web: RTCP Mode CLI: sbc-rtcp-mode [IPProfile_SBCRTCPMode]	Defines how the device handles RTCP packets during call sessions for this SIP entity. This is useful for interworking RTCP between SIP entities. For example, this may be necessary when incoming RTCP is not compatible with the destination SIP entity's (this IP Profile) RTCP support. In such a scenario, the device can generate the RTCP and send it to the SIP entity. ▪ [0] Transparent (default) = RTCP is forwarded as is. ▪ [1] Generate Always = Generates RTCP packets during active and inactive (e.g., during call hold) RTP periods (i.e., media is 'a=recvonly' or 'a=inactive' in the INVITE SDP). ▪ [2] Generate only if RTP Active = Generates RTCP packets only during active RTP periods. In other words, the device does not generate RTCP when there is no RTP traffic (such as when a call is on hold). **Note:** The corresponding global parameter is SBCRTCPMode.
Web: Jitter Compensation CLI: sbc-jitter-compensation [IpProfile_SBCJitterCompensation]	Enables the on-demand jitter buffer for SBC calls. This jitter buffer is useful when incoming packets are received at inconsistent intervals (i.e., packet delay variation). The jitter buffer stores the packets and sends them out at a constant rate (according to the coder's settings). ▪ [0] Disable (default) ▪ [1] Enable **Note:** The jitter buffer parameters, 'Dynamic Jitter Buffer Minimum Delay' (DJBufMinDelay) and 'Dynamic Jitter Buffer Optimization Factor' (DJBufOptFactor) can be used to configure minimum packet delay only when transcoding is employed.

Reader's Notes

Part V

Session Border Controller Application

21 SBC Overview

This section provides a detailed description of the device's SBC application.

Notes:

- For guidelines on how to deploy your E-SBC device, refer to the E-*SBC Design Guide* document.
- The SBC feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 309.
- For the maximum number of supported SBC sessions, and SBC users than can be registered in the device's registration database, see "Technical Specifications" on page 458.

The SBC application supports the following main features:

- NAT traversal: The device supports NAT traversal, allowing, for example, communication with ITSPs with globally unique IP addresses, for LAN-to-WAN VoIP signaling (and bearer), using two independent legs. This also enables communication for "far-end" users located behind a NAT on the WAN. The device supports this by:
  - Continually registering far-end users in its dynamic database.
  - Maintaining remote NAT binding state by frequent registrations, thereby, off-loading far-end registrations from the LAN IP PBX.
  - Using Symmetric RTP (RFC 4961) to overcome bearer NAT traversal.
- VoIP firewall and security for signaling and media:
  - SIP signaling:
    - Deep and stateful inspection of all SIP signaling packets.
    - SIP dialog initiations may be rejected based on values of incoming SIP INVITE message and other Layer-3 characteristics.
    - Packets not belonging to an authorized SIP dialog are discarded.
  - RTP:
    - Opening pinholes (ports) in the device's firewall based on Offer-Answer SDP negotiations.
    - Deep packet inspection of all RTP packets.
    - Late rogue detection - if a SIP session was gracefully terminated and someone tries to "ride on it" with rogue traffic from the already terminated RTP and SIP context, the VoIP Firewall prevents this from occurring.
    - Disconnects call (after user-defined time) if RTP connection is broken.
    - Black/White lists for both Layer-3 firewall and SIP classification.
- Topology hiding: The device intrinsically supports topology hiding, limiting the amount of topology information displayed to external parties. For example, IP addresses of ITSPs' equipment (e.g. proxies, gateways, and application servers) can be hidden from outside parties. The device's topology hiding is provided by implementing back-to-back user agent (B2BUA) leg routing:
  - Strips all incoming SIP Via header fields and creates a new Via value for the outgoing message.
  - Each leg has its own Route/Record Route set.
  - Modifies SIP To, From, and Request-URI host names (must be configured using the Message Manipulations table).
  - Generates a new SIP Call-ID header value (different between legs).
  - Changes the SIP Contact header to the device's own address.
  - Layer-3 topology hiding by modifying source IP address in the SIP IP header.
- SIP normalization: The device supports SIP normalization, whereby the SBC application

can overcome interoperability problems between SIP user agents. This is achieved by the following:

  - Manipulation of SIP URI user and host parts.
  - Connection to ITSP SIP trunks on behalf of an IP-PBX - the device can register and utilize user and password to authenticate for the IP-PBX.

- Survivability:
  - Routing calls to alternative routes such as the PSTN.
  - Routing calls between user agents in the local network using a dynamic database (built according to registrations of SIP user agents).
- Routing:
  - IP-to-IP routing translations of SIP, UDP, TCP, TLS (when extensive transcoding is not required).
  - Load balancing and redundancy of SIP servers.
  - Routing according to Request-URI\Specific IP address\Proxy\FQDN.
  - Alternative routing.
  - Routing between different Layer-3 networks (e.g., LAN and WAN).
- Load balancing\redundancy of SIP servers.
- ITSP accounts.
- SIP URI user and host name manipulations.
- Coder transcoding.

21.1 SIP Network Definitions

The device's SBC application can implement multiple SIP signaling and RTP (media) interfaces.

21.2 SIP Dialog Initiation Process

The device's SIP dialog initiation process concerns all incoming SIP dialog initiation requests. This includes SIP methods such as INVITE, SUBSCRIBE, OPTIONS, REFER, INFO, UNSOLICITED NOTIFY, MESSAGE, and REGISTER.

The SIP dialog initiation process consists of the following stages:

1. **Determining source and destination URL:** The SIP protocol has more than one URL in a dialog-establishing request that may represent the source and destination URLs. When handling an incoming request, the device uses specific SIP headers for obtaining the source and destination URLs. Once these URLs are determined, their user and host parts are used as input for the classification process, message manipulation, and call routing.
   - **All SIP requests (e.g., INVITE) except REGISTER dialogs:**
     - Source URL: The source URL is obtained from the SIP header according to the following logic:
       - ✓ The source URL is obtained from the From header.
       - ✓ If the From header contains the value 'Anonymous', the source URL is obtained from the P-Preferred-Identity header.
       - ✓ If the P-Preferred-Identity header does not exist, the source URL is obtained from the P-Asserted-Identity header.
     - Destination URL: The destination URL is obtained from the Request-URI.
   - **REGISTER dialogs:**
     - Source URL: The source URL is obtained from the To header.
     - Destination URL: The destination URL is obtained from the Request-URI.

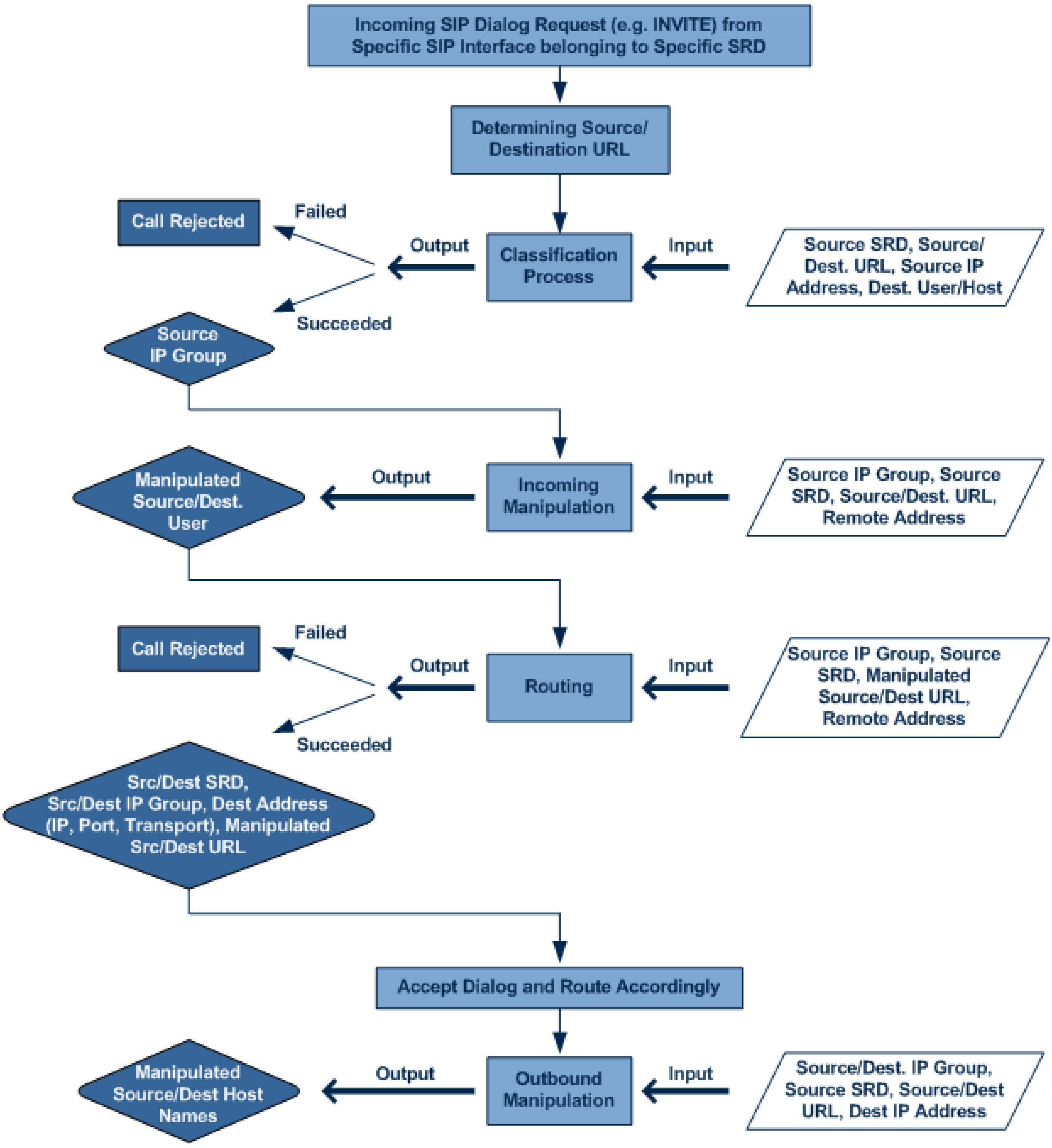
Note: You can determine the SIP header from where the device obtains the source URL in the incoming SIP request. This is done in the IP Group table using the 'Source URI Input' parameter.

2. **Classifying incoming SIP dialog-initiating requests to a source IP Group:** The classification identifies the incoming SIP dialog request as belonging to a specific IP Group (from where the SIP dialog request originated). For more information, see "Configuring Classification Rules" on page 249.
3. **SBC IP-to-IP routing:** The device routes the call to a destination that can be configured to one of the following:
    - Registered user Contact listed in the device's database (only for User-type IP Groups).
    - IP Group - the destination is the address configured for the Proxy Set associated with the IP Group (allows redundancy/load balancing).
    - Specified destination address (can be based on IP address, host name, port, transport type, and/or SRD). Routing to a host name can be resolved using NAPTR/SRV/A-Record.
    - Request-URI of incoming SIP dialog initiating requests.
    - ENUM query.
    - Hunt Group - used for call survivability.
    - IP address (in dotted-decimal notation or FQDN - NAPTR/SRV/A-Record resolutions) according to a specified Dial Plan index listed in the loaded Dial Plan file.
    - LDAP server or LDAP query result.

    For more information, see "Configuring SBC IP-to-IP Routing Rules" on page 255.
4. **Manipulating SIP URI user part (source and destination) of inbound and/or outbound SIP dialog requests:** You can configure rules for manipulating the SIP URI user part (source and destination) on the inbound and/or outbound leg. For more information, see "SBC Manipulations" on page 264.
5. **SIP message manipulations:** You can configure SIP message manipulation rules that can add, remove, and/or modify SIP headers and parameters. For more information, see "Configuring SIP Message Manipulation" on page 193.

The flowchart below illustrates the SBC process:

Figure 85: Routing Process


Incoming SIP Dialog Request (e.g. INVITE) from Specific SIP Interface belonging to Specific SRD
Determining Source/ Destination URL
Call Rejected
Failed
Output
Classification Process
Input
Source SRD, Source/ Dest. URL, Source IP Address, Dest. User/Host
Succeeded
Source IP Group
Manipulated Source/Dest. User
Output
Incoming Manipulation
Input
Source IP Group, Source SRD, Source/Dest. URL, Remote Address
Call Rejected
Failed
Output
Routing
Input
Source IP Group, Source SRD, Manipulated Source/Dest URL, Remote Address
Succeeded
Src/Dest SRD, Src/Dest IP Group, Dest Address (IP, Port, Transport), Manipulated Src/Dest URL
Accept Dialog and Route Accordingly
Manipulated Source/Dest Host Names
Output
Outbound Manipulation
Input
Source/Dest. IP Group, Source SRD, Source/Dest URL, Dest IP Address

21.3 User Registration

To allow registrations to traverse the SBC, the device must be configured with at least one User-type IP Group. These IP Groups represent a group of user agents that share the following characteristics:

- Perform registrations and share the same serving proxy\registrar
- Possess identical SIP and media behavior
- Reside on the same Layer-3 network and are associated with the same SRD

Typically, the device is configured as the user agent's outbound proxy and the device is configured (using the IP-to-IP Routing table) to route requests received from this IP Group to the serving proxy and vice versa. Survivability can be achieved using the alternative routing feature.

21.3.1 Initial Registration Request Processing

The device's handling of registration requests (REGISTER messages) are as follows:

- The device obtains the source URL from the SIP To header and the destination URL from the Request-URI.
- The device's classification process for REGISTER requests is the same as for other SIP messages. However, the REGISTER request must be received from **User-type** IP Groups only. If classification fails or the IP Group is not a User-type, the device rejects the registration request.
- The device's routing of REGISTER requests is done using the IP-to-IP Routing table. If the destination is a User-type IP Group, the device does not forward the registration; instead, it accepts (replies with a SIP 200 OK response) or rejects (SIP 4xx) the request, according to the user's IP Group configuration.
- If registration succeeds (replied with 200 OK by the IP PBX), the device adds a record to its Users Registration database that identifies the specific contact of the specific user (AOR). This record is used by the device to route subsequent requests to the specific user (in normal or in survivability modes).
- Alternative routing can be configured for REGISTER requests, in the IP-to-IP Routing table.
- The record in the device's database includes the SIP Contact header. Every REGISTER request is added to the database before manipulation, allowing correct user identification in the Classification process for the next received request.
- Call Admission Control (CAC) can be configured for incoming and outgoing REGISTER requests. For example, limiting REGISTER requests from a certain IP Group/SRD. Note that this is only for concurrent register dialogs and not concurrent registrations in the device's Users Registration database.
- The device can retain the original value of the SIP Expires header received from the user or proxy, in the outgoing REGISTER message. This feature also applies when the device is in survivability mode (i.e., REGISTER requests cannot be forwarded to the proxy and is terminated by the device). This is configured by the SBCUserRegistrationTime, SBCProxyRegistrationTime, and SBCSurvivabilityRegistrationTime parameters.
- By default, the Contact header in the outgoing REGISTER is populated with a unique contact generated by the device and associated with the specific registration. Alternatively, the original user can be retained in the Contact header and used in the outgoing REGISTER request (using the SBCKeepContactUserinRegister parameter).

21.3.2 SBC Users Registration Database

The device manages a dynamic Users Registration database that is updated according to registration requests that traverse it. Each database entry for a user represents a binding between an AOR (obtained from the SIP To header) and one or more contact (obtained from the SIP Contact headers). Database bindings are added upon successful registration responses.

Database bindings are removed in the following cases:

- Successful de-registration responses (REGISTER with Expires header that equals zero).
- Registration failure responses.
- Timeout of the Expires header value (in scenarios where the user agent did not send a refresh registration request).

Note: The device's Users Registration database poses the following restrictions:

- The same contact cannot belong to more than one AOR.
- Contacts with identical URIs and different ports and transport types are not supported (same key is created).
- Multiple contacts in a single REGISTER is not supported.
- One database is shared between all User-type IP Groups.

21.3.3 Routing using Users Registration Database

The device uses the Users Registration database when routing calls of registered users. The device tries to locate a match for the IP-to-IP Routing rule between the incoming Request-URI and the following, listed in chronological order:

1. Unique Contact: the contact generated by the device and sent in the initial registration request to the serving proxy.
2. Registered AOR in the Users Registration database: the AOR of the incoming REGISTER request.
3. Registered Contact in the Users Registration database: the Contact of the incoming REGISTER request.

If registrations are destined to the database (using the above rules), the device does not attempt to find a database match, but instead replies with a SIP 200 OK (used for Survivability). Once a match is found, the request is routed either to the contact received in the initial registration or (if the device identifies that the user agent is behind a NAT) to the source IP address of the initial registration.

21.3.4 Registration Refreshes

Registration refreshes are incoming REGISTER requests that are associated with a registered user in the Users Registration database. These refreshes are routed to the serving proxy only if the serving proxy Expires time is about to expire; otherwise, the device responds with a 200 OK without routing the REGISTER. Each such refreshes also refresh the internal timer set on the device for this specific registration.

The device automatically notifies SIP Proxy / Registrar servers of users that are registered in the device's Users Registration database whose registration timeout has expired. When a user's registration timer expires, the device removes the user record from the database and sends an un-register notification (REGISTER message with the Expires header set to 0) to the Proxy/Registrar. This occurs only if a REGISTER message is sent to an IP Group destination type (in the IP-to-IP Routing table).

The device can be configured to add extra time (grace period) to the expiration timer of registered users in the database. If you configure this grace period, the device keeps the user in the database (and does not send an un-register to the Registrar server), allowing the user to send a "late" re-registration to the device. The device removes the user from the database only when this additional time expires. This featured is configured using the 'User Registration Grace Time' parameter (SBCUserRegistrationGraceTime).

The device keeps registered users in its Users Registration database even if connectivity with the SIP proxy server is lost (i.e., proxy does not respond to users' registration refresh requests). The device removes users from the database only when their registration expiry time is reached (with the additional grace period, if configured).

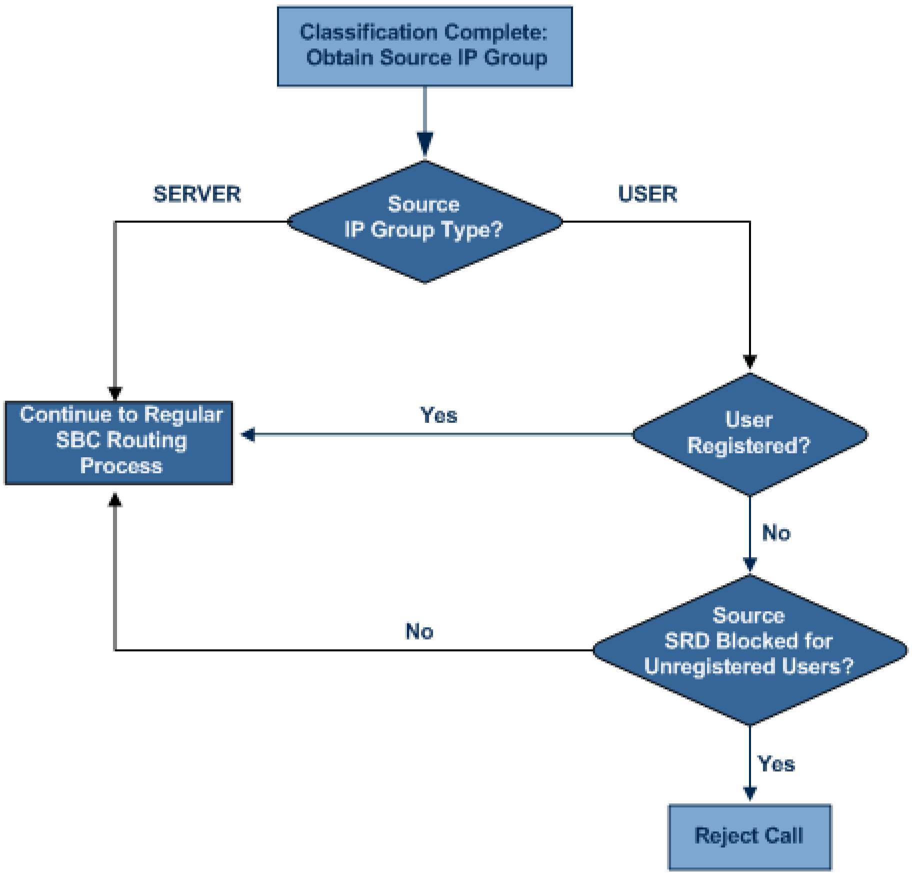
21.3.5 Registration Restriction Control

The device provides flexibility in controlling user registration:

- **Limiting Number of Registrations:** You can limit the number of users that can register with the device per IP Group and/or SRD. By default, no limitation exists for registered users. This is configured in the SRD and IP Group tables.

- **Blocking Incoming Calls from Unregistered Users:** You can block incoming calls (INVITE requests) from unregistered users (pertaining to User-type IP Groups). By default, calls from unregistered users are not blocked. This is configured in the SRD table. The flowchart below depicts the process for blocking unregistered users. When the call is rejected, the device sends a SIP 500 (Server Internal Error) response to the remote end.

Figure 86: Blocking Incoming Calls from Unregistered Users



21.4 SBC Media Handling

Media behavior includes anything related to the establishment, management and termination of media sessions within the SIP protocol. Media sessions are created using the SIP "offer"/"answer" mechanism. If successful, the result is a bi-directional media (RTP) flow (e.g. audio, fax, modem, DTMF). Each offer/answer may create more than one media session of different types (e.g. audio and fax). In a SIP dialog, multiple offer/answer transactions may occur, each may change the media sessions characteristics (e.g. IP address, port, coders, media types, and RTP mode). The media capabilities exchanged in an offer/answer transaction include the following:

- Media types (Audio, Secure Audio, Video, Fax, Text...)
- IP addresses and ports of the media flow
- Media flow mode (send receive, receive only, send only, inactive)
- Media coders (coders and their characteristics used in each media flow)
- Other (standard or proprietary) media and session characteristics

Even though the device usually does not change the negotiated media capabilities (mainly performed by the remote user agents), it does examine the media exchange to control negotiated media types (if necessary) and to know how to open the RTP media channels (IP addresses, coder type, payload type etc.). The device forwards multiple video streams and text, as is.

The device interworks (normalization) the media (RTP-to-RTP, SRTP-to-RTP, and SRTP-to-SRTP) between its SBC legs. It "re-builds" specific fields in the RTP header when forwarding media packets. The main fields include the sequence number, SSRC, and timestamp.

The device is aware and sometimes active in the offer\answer process due to the following:

- NAT traversal: the device changes the SDP address to be its own address, thereby, resolving NAT problems.
- Firewall and security:
  - RTP pin holes - only RTP packets related to a successful offer\answer negotiation traverse the device: When the device initializes, there are no RTP pin holes opened, this means that each RTP\RTCP packets destined to the device are discarded. Once an offer\answer transaction ends successfully, an RTP pin hole is opened and RTP\RTCP flows between the two remote user agents. Once a pin hole is opened, the payload type and RTP header version is validated for each packet. RTP pin holes close if one of the associated SIP dialogs is closed (may also be due to broken connection).
  - Late rogue detection - once a dialog is disconnected, the related pin holes also disconnect.
  - Deep Packet inspection of the RTP that flows through the opened pin holes.
- Adding of media functionality to SIP user agents:
  - Transcoding (for a description on the transcoding modes, see Transcoding Modes)
  - Broken connection

According to the above functionalities, the call can be configured to operate in one of the following modes:

- **Media Anchoring without Transcoding (Transparent):** RTP traverses the device with minimal RTP packet changes (no DSP resources needed). This is typically used to solve NAT, firewall, and security issues. In this mode, all the "audio" coders in the received offer are included in the SBC outgoing offer. The Coder Table configuration has no effect on the coders in the outgoing offer. For more information, see "Media Anchoring without Transcoding (Transparent)" on page 221.
- **Media Anchoring with Transcoding:** RTP traverses the device and each leg uses a different coder or coder parameters (DSP resources are required). For more information, see Media Anchoring with Transcoding.
- **No Media Anchoring:** The RTP packet flow does not traverse the device. Instead, the two SIP UA's establish a direct RTP/SRTP flow between one another (see "No Media Anchoring" on page 222).

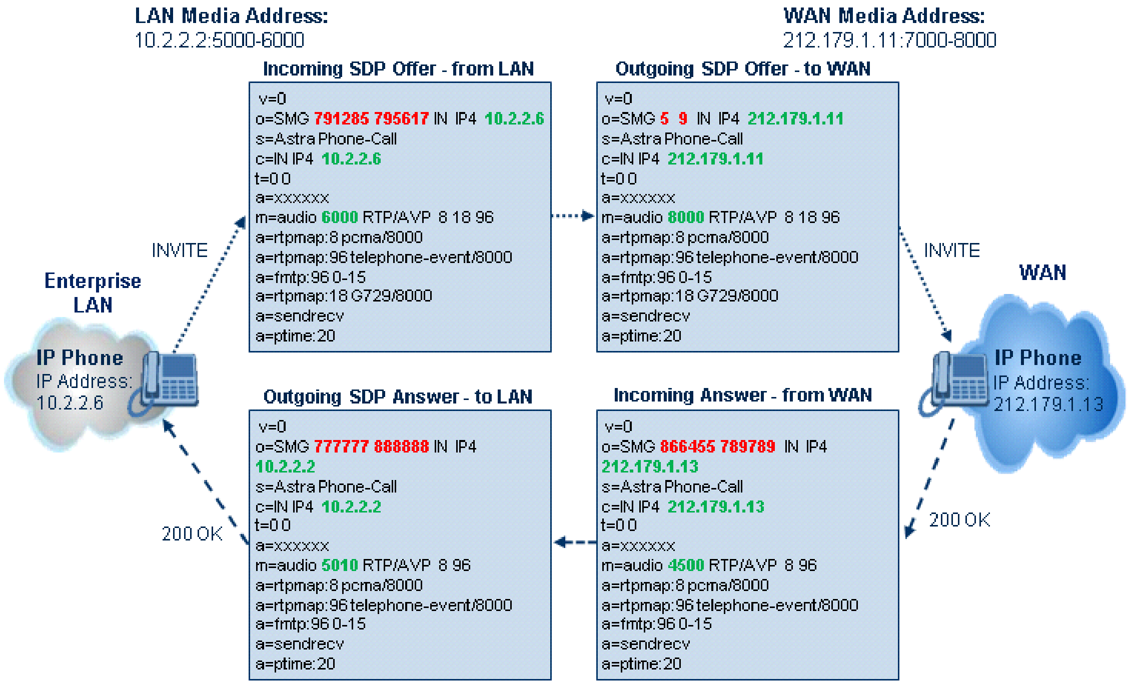
21.4.1 Media Anchoring without Transcoding (Transparent)

To direct the RTP to flow through the device (for NAT traversal, firewall and security), all IP address fields in the SDP are modified:

- Origin: IP address, session and version id
- Session connection attribute ('c=' field)
- Media connection attribute ('c=' field)
- Media port number
- RTCP media attribute IP address and port

Each SBC leg allocates and uses the device's local ports (e.g., for RTP\RTCP\fax). The local ports are allocated from a Media Realm associated with each leg. The legs are associated with a Media Realm as follows: If the leg's IP Group is configured with a Media Realm, then this is the associated Media Realm; otherwise, the leg's SRD Media Realm is the associated one. The figure below illustrates an example of SDP handling for a call between a LAN IP Phone 10.2.2.6 and a remote IP Phone 212.179.1.13 on the WAN.

Figure 87: SDP Offer/Answer Example



Incoming SDP Offer - from LAN

```
v=0
o=SMG 791285 795617 IN IP4 10.2.2.6
s=Astra Phone-Call
c=IN IP4 10.2.2.6
t=0 0
a=xxxxxx
m=audio 6000 RTP/AVP 8 18 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=rtpmap:18 G729/8000
a=sendrecv
a=ptime:20
```

Outgoing SDP Offer - to WAN

```
v=0
o=SMG 5 9 IN IP4 212.179.1.11
s=Astra Phone-Call
c=IN IP4 212.179.1.11
t=0 0
a=xxxxxx
m=audio 8000 RTP/AVP 8 18 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=rtpmap:18 G729/8000
a=sendrecv
a=ptime:20
```

Outgoing SDP Answer - to LAN

```
v=0
o=SMG 777777 888888 IN IP4
10.2.2.2
s=Astra Phone-Call
c=IN IP4 10.2.2.2
t=0 0
a=xxxxxx
m=audio 5010 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=sendrecv
a=ptime:20
```

Incoming Answer - from WAN

```
v=0
o=SMG 866455 789789 IN IP4
212.179.1.13
s=Astra Phone-Call
c=IN IP4 212.179.1.13
t=0 0
a=xxxxxx
m=audio 4500 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=sendrecv
a=ptime:20
```

21.4.2 No Media Anchoring

The No Media Anchoring (commonly referred to as Anti-Tromboning) feature enables the use of SBC signaling capabilities without handling the media (RTP/SRTP) flow between remote SIP user agents (UA). The media flow does not traverse the device. Instead, the two SIP UAs establish a direct media flow (i.e., direct call) between one another. Signaling continues to traverse the device with minimal intermediation and involvement to enable certain SBC abilities such as routing.

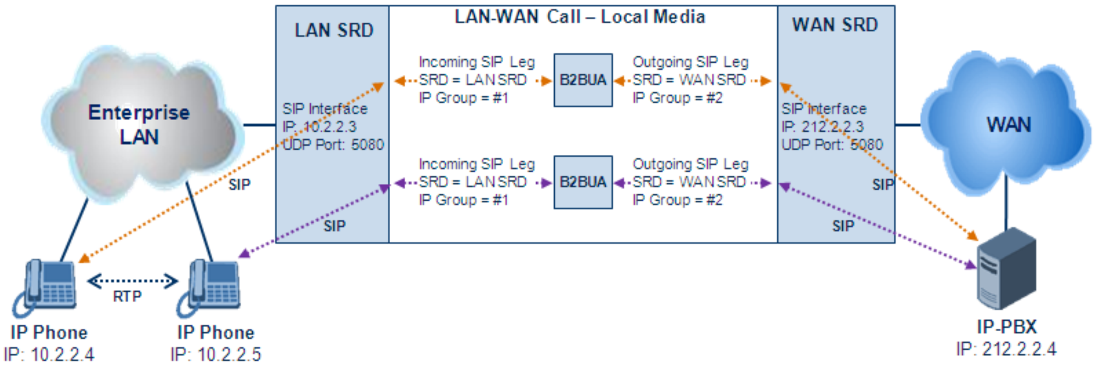
By default, media packets traverse the device to solve NAT problems, enforce media security policy, perform media transcoding between the two legs, and media monitoring. In certain deployments, specific calls do not require media anchoring, for example, when there is no need for NAT, security, or transcoding. This is typical for calls between users in the LAN:

- Internal LAN calls: When the SBC routes a call between two UAs within the same LAN, the SBC can forward the SDP directly between caller and callee, and direct the media to flow between the UAs without traversing the SBC.
- Internal LAN calls via WAN: In this setup, the SBC dynamically identifies the call as between UAs located in the same network (i.e., LAN) and thereby, directs the media to flow between these UAs without traversing the SBC.

The No Media Anchoring feature is typically implemented in the following scenarios:

- The device is located within the LAN.
- Calls between two SIP UAs in the same LAN and signaling is sent to a SIP proxy server (or hosted IP PBX) located in the WAN.
- The device does not need to perform NAT traversal (for media) and all the users are in the same domain.

Figure 88: SBC SIP Signaling without RTP Media Flow



The benefits of implementing the No Media Anchoring feature include the following:

- Saves network bandwidth
- Reduces CPU usage (no media handling)
- Avoids interference in SDP negotiation and header manipulation on RTP/SRTP

The device handles the No Media Anchoring process as follows:

1. Identifies a No Media Anchoring call according to configuration and the call's properties (such as source, destination, IP Group, and SRD).
2. Handles the identified No Media Anchoring call.

The No Media Anchoring feature is enabled for all calls (regardless of SRD), using the global parameter, SBCDirectMedia. You can also enable No Media Anchoring per SRD (in the SRD table), whereby calls belonging to this same SRD (source and destination) are handled as No Media Anchoring (direct media) calls. This occurs even if the global parameter is disabled.

Notes:

- No Media Anchoring can be used when the SBC does not do NAT traversal (for media) where all the users are in the same domain.
- No Media Anchoring calls cannot operate with the following features:
  - √ Manipulation of SDP data (offer/answer transaction) such as ports, IP address, coders
  - √ Extension of RFC 2833 / out-of-band DTMF / in-band DTMF
  - √ Extension of SRTP/RTP
- All restriction features (Allowed Coders, restrict SRTP/RTP, restrict RFC 2833) can operate with No Media Anchoring calls. Restricted coders are removed from the SDP offer message.
- For No Media Anchoring, opening of voice channels and allocation of IP media ports are not required.
- When two UAs belong to the same SRD which is enabled for No Media Anchoring, and one of the UAs is defined as a foreign user (example, "follow me service") located in the WAN while the other UA is located in the LAN: calls between these two UAs cannot be established until the No Media Anchoring for the SRD is disabled, as the device does not interfere in the SIP signaling. In other words, parameters such as IP addresses are not manipulated for calls between LAN and WAN (although required).
- When the global parameter SBCDirectMedia is disabled, No Media Anchoring can only occur for calls between UAs belonging to the same SRD that is configured for No Media Anchoring in the SRD table.

21.4.3 Restricting Coders

The SBC Allowed Coders (coders restriction) feature determines the coders that can be used for a specific SBC leg. This provides greater control over bandwidth by enforcing the use of specific coders (*allowed* *coders groups*) while preventing the use of other coders. This is done by defining a group of allowed coders for the SBC leg, as described below:

1. Configure a Coders Group for allowed coders, using the AllowedCodersGroup parameter.
2. Select this Coders Group using the SBCAllowedCodersGroupID parameter of the IP Profile table.
3. Enable this feature by setting the SBCAllowedCodersMode parameter of the IP Profile table to **Restriction**.

Coders that are not listed (including unknown coders) in the Allowed Coders Group are removed from the SDP offer. Therefore, only coders common between the SDP offer and Allowed Coders Group are used. If the SDP offer does not list any of the Allowed Coders, the call is rejected.

Notes:

- For a list of supported coders, see Configuring Default Coders.
- Allowed Coder Groups are applicable only to audio media.

The Allowed Coders process is as follows:

a. The device receives an incoming SIP message with SDP (offer) and checks the offered coders.
b. The source (first) leg may have Allowed Coders (i.e. list of coders that can be used - enforced).
c. The device checks for common coders between the SDP offered coders and the Allowed Coders Group list.

For example, assume the following:

- The SDP coder offer includes the following coders: G.729, G.711, and G.723.
- The source (first) leg includes the following Allowed Coders: G.711 and G.729.

The device selects the common coders, i.e., G.711 and G.729 (with changed preferred coder priority - highest for G.711). In other words, it removes the coders that are not in the Allowed Coders list and the order of priority is first according to the Allowed Coders list.

21.4.4 Prioritizing Coder List in SDP Offer

In addition to restricting the use of coders with Allowed coders, you can prioritize the coders listed in the SDP offer. This feature is referred to as *Coder Preference*. This is done on both SBC legs:

- **Incoming SDP offer:** The device arranges the coder list according to the order in the Allowed Coders Group table. The coders listed higher up in the table take preference over ones listed lower down in the table. This feature is enabled by setting the 'Allowed Coders Mode' parameter in the IP Profile table to **Preference** or **Restriction and Preference**. If set to **Preference**, in addition to the Allowed coders that are listed first in the SDP offer, the original coders received in the SDP are retained and listed after the Allowed coders. Thus, this mode does not necessarily restrict coder use to Allowed coders, but uses (prefers) the Allowed coders whenever possible.
- **Outgoing SDP offer:** If only Allowed coders are used, the coders are arranged in the SDP offer as described above.

21.4.5 SRTP-RTP and SRTP-SRTP Transcoding

The device supports transcoding between SRTP and RTP. The device can also enforce specific SBC legs to use SRTP and/or RTP. The device's handling of SRTP/RTP is configured using the IP Profile parameter, SBCMediaSecurityBehaviour, which provides the following options:

- SBC passes the media as is, regardless of whether it's RTP or SRTP (default).
- SBC legs negotiate only SRTP media lines (m=); RTP media lines are removed from the incoming SDP offer\answer.

- SBC legs negotiate only RTP media lines; SRTP media lines are removed from the incoming offer\answer.
- Each SDP offer\answer is extended (if not already) to two media lines for RTP and SRTP.

If after SDP offer\answer negotiation, one SBC leg uses RTP while the other uses SRTP, then the device performs RTP-SRTP transcoding. To translate between RTP and SRTP, the following prerequisites must be met:

- At least one supported SDP "crypto" attribute.
- The EnableMediaSecurity parameter must be set to 1.

Transcoding where both legs are configured for SRTP is typically required to trans-encrypt and trans-decrypt. This is relevant when the MKI and Symmetric MKI parameters are enabled. In other words, both sides need to both encrypt and decrypt the outgoing and incoming SRTP packets, respectively.

21.4.6 Multiple RTP Media Streams per Call Session

The device's SBC application supports multiple RTP media streams per SBC call session. Up to five different media types can be included in a session:

- Audio (m=audio)
- Video (m=video)
- Text (m=text)
- Fax (m=image)

Therefore, the device can provide transcoding of various attributes in the SDP offer/answer (e.g., codec, port, and packetization time) per media type. If the device is unable to perform transcoding (for example, does not support the codec), it relays the SBC dialog transparently.

21.5 Limiting SBC Call Duration

You can define a maximum allowed duration (in minutes) for SBC calls. If an established call reaches this user-defined limit, the device terminates the call. This feature ensures calls are properly terminated, allowing available resources for new calls. This feature is configured using the MaxCallDuration parameter.

21.6 SBC Authentication

The device can authenticate SIP servers and SBC users (clients). The different methods of support for this functionality is described in the following subsections.

21.6.1 SIP Authentication Server Functionality

The device can function as an Authentication server for authenticating received SIP message requests, based on HTTP authentication Digest with MD5. Alternatively, such requests can be authenticated by an external, third-party server.

When functioning as an Authentication server, the device can authenticate the following SIP entities:

- **SIP servers:** This is applicable to Server-type IP Groups. This provides protection from rogue SIP servers, preventing unauthorized usage of device resources and functionality. To authenticate remote servers, the device challenges the server with a user-defined username and password that is shared with the remote server. When the device receives an INVITE request from the remote server, it challenges the server by replying with a SIP 401 Unauthorized response containing the WWW-Authenticate header. The remote server then re-sends the INVITE containing an Authorization header with authentication information based on this username-password combination to confirm its identity. The device uses the username and password to authenticate the message prior to processing it.

- **SIP clients:** These are clients belonging to a User-type IP Group. This support prevents unauthorized usage of the device's resources by rogue SIP clients. When the device receives an INVITE or REGISTER request from a client (e.g., SIP phone) for SIP message authorization, the device processes the authorization as follows:
  1. The device challenges the received SIP message only if it is configured as a SIP method (e.g., INVITE) for authorization. This is configured in the IP Group table, using the 'Authentication Method List' parameter.
  2. If the message is received without a SIP Authorization header, the device "challenges" the client by sending a SIP 401 or 407 response. The client then resends the request with an Authorization header (containing the user name and password).
  3. The device validates the SIP message according to the AuthNonceDuration, AuthChallengeMethod and AuthQOP parameters.
     - If validation fails, the device rejects the message and sends a 403 (Forbidden) response to the client.
     - If validation succeeds, the device verifies client identification. It checks that the username and password received from the client is the same username and password in the device's User Information table / database (see "SBC User Information for SBC User Database" on page 305). If the client is not successfully authenticated after three attempts, the device sends a SIP 403 (Forbidden) response to the client. If the user is successfully identified, the device accepts the SIP message request.

The device's Authentication server functionality is configured per IP Group, using the 'Authentication Mode' parameter in the IP Group table (see "Configuring IP Groups" on page 174).

21.6.2 User Authentication based on RADIUS

The device can authenticate SIP clients (users) using a remote RADIUS server. The device supports the RADIUS extension for digest authentication of SIP clients, according to draft-sterman-aaa-sip-01. Based on this standard, the device generates the nonce (in contrast to RFC 5090, where it is done by the RADIUS server).

RADIUS based on draft-sterman-aaa-sip-01 operates as follows:

1. The device receives a SIP request without an Authorization header from the SIP client.
2. The device generates the nonce and sends it to the client in a SIP 407 (Proxy Authentication Required) response.
3. The SIP client sends the SIP request with the Authorization header to the device.
4. The device sends an Access-Request message to the RADIUS server.
5. The RADIUS server verifies the client's credentials and sends an Access-Accept (or Access-Reject) response to the device.
6. The device accepts the SIP client's request (sends a SIP 200 OK or forwards the authenticated request) or rejects it (sends another SIP 407 to the SIP client).

To configure this feature, set the SBCServerAuthMode ini file parameter to 2.

21.7 Interworking SIP Signaling

The device supports interworking of SIP signaling messages to ensure interoperability between communicating SIP UAs or entities. This is critical in network environments where the UAs on opposing SBC legs have different SIP signaling support. For example, some UAs may support different versions of a SIP method while others may not event support a specific SIP method. The configuration method for assigning specific SIP message handling modes to UAs, includes configuring an IP Profile with the required interworking mode, and then assigning the IP Profile to the relevant IP Group.

This section describes some of the device's support for handling SIP methods to ensure interoperability.

21.7.1 Interworking SIP 3xx Redirect Responses

The device supports interworking of SIP 3xx redirect responses. By default, the device's handling of SIP 3xx responses is to send the Contact header unchanged. However, some SIP UAs may support different versions of the SIP 3xx standard while others may not even support SIP 3xx.

The handling of SIP 3xx can be configured for all calls, using the global parameter SBC3xxBehavior. For configuring different SIP 3xx handling options for different UAs (i.e., per IP Group), use the IP Profile table parameter, 'SBC Remote 3xx Behavior'.

21.7.1.1 Resultant INVITE Traversing Device

The device can handle SIP 3xx responses so that the new INVITE message sent as a result of the 3xx traverses the device. The reasons for enforcing resultant INVITEs to traverse the device may vary:

- The user that receives the 3xx is unable to route to the 3xx contact (i.e., the user is on the LAN and the new contact is on the WAN). In such a scenario, the device enables the user to reach the WAN contact and overcome NAT problems.
- Enforce certain SBC policies (e.g., call admission control, header manipulation, and transcoding) on the resultant INVITE.

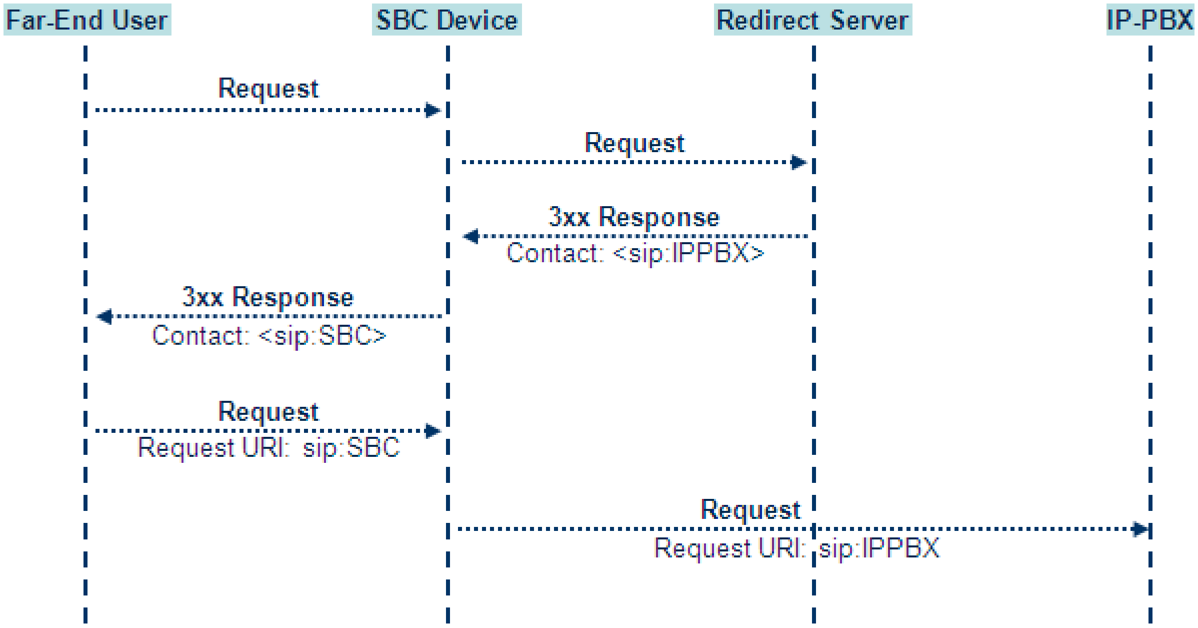
The device enforces this by modifying each Contact in the 3xx response as follows:

- Changes the host part to the device's IP address – this change causes the remote user agent to send the INVITE to the device.
- Adds a special prefix ("T~&R_") to the Contact user part – to identify the new INVITE as a 3xx resultant INVITE.

The SBC handling for the 3xx resultant INVITE is as follows:

1. The incoming INVITE is identified as a 3xx resultant INVITE according to the special prefix.
2. The device automatically replaces the SBC host part (in the Request-URI) with the host from the 3xx Contact.
3. The prefix ("T~&R_") remains in the user part for the classification, manipulation, and routing mechanisms.
4. The classification, manipulation, and routing processes are done exactly like any other INVITE handling. The special prefix can be used for specific routing rules for 3xx resultant INVITEs.
5. The prefix is removed before the resultant INVITE is sent to the destination.

Figure 89: SIP 3xx Response Handling



The process of this feature is described using an example:

1. The device receives the Redirect server's SIP 3xx response (e.g., Contact: <sip:User@IPPBX:5060;transport=tcp;param=a>;q=0.5).
2. The device replaces the Contact header value with the special prefix and database key value as user part, and with the device's URL as host part (e.g., Contact: <sip:Prefix_Key_User@SBC:5070;transport=udp>;q=0.5).
3. The device sends this manipulated SIP 3xx response to the Far-End User (FEU).
4. The FEU sends a new request with the Request-URI set to the value of the received 3xx response's Contact header (e.g., RequestURI: sip:Prefix_Key_User@SBC:5070;transport=udp).
5. Upon receipt of the new request from the FEU, the device replaces the Request-URI with the new destination address (e.g., RequestURI: sip:Prefix_User@IPPBX:5070;transport=tcp;param=a).
6. The device removes the user prefix from the Request-URI, and then sends this Request-URI to the new destination (e.g., RequestURI: sip:User@IPPBX:5070;transport=tcp;param=a).

21.7.1.2 Local Handling of SIP 3xx

The device can handle SIP 3xx responses on behalf of the dialog-initiating UA and retry the request (e.g., INVITE) using one or more alternative URIs included in the 3xx response. The new request includes SIP headers from the initial request such as Diversion, History-Info, P-Asserted-Id, and Priority. The source and destination URIs can be manipulated using the regular manipulation mechanism.

The device sends the new request to the alternative destination according to the IP-to-IP Routing table rules. (where the 'Call Trigger' field is set to **3xx**). It is also possible to specify the IP Group that sent the 3xx request as matching criteria for the re-routing rule in this table ('ReRoute IP Group ID' field).

21.7.2 Interworking SIP Diversion and History-Info Headers

This device can be configured to interwork between the SIP Diversion and History-Info headers. This is important, for example, to networks that support the Diversion header but not the History-Info header, or vice versa. Therefore, mapping between these headers is crucial for preserving the information in the SIP dialog regarding how and why (e.g., call redirection) the call arrived at a certain SIP UA.

This feature is configured in the IP Profile table (IPProfile parameter) using the following new parameters:

- SBCDiversionMode - defines the device's handling of the Diversion header
- SBCHistoryInfoMode - defines the device’s handling of the History-Info header

The handling of the SIP Diversion and History-Info headers is described in the table below:

Table 67: Handling of SIP Diversion and History-Info Headers

Parameter Value	SIP Header Present in Received SIP Message		
	Diversion	History-Info	Diversion and History-Info
HistoryInfoMode = Add **DiversionMode = Remove**	Diversion converted to History-Info. Diversion removed.	Not present	Diversion removed.
HistoryInfoMode = Remove **DiversionMode = Add**	Not present.	History-Info converted to Diversion. History-Info removed.	History-Info added to Diversion. History-Info removed.
HistoryInfoMode = Disable **DiversionMode = Add**	Diversion converted to History-Info.	Not present.	Diversion added to History-Info.

Parameter Value	SIP Header Present in Received SIP Message		
HistoryInfoMode = Disable **DiversionMode = Add**	Not present.	History-Info converted to Diversion.	History-Info added to Diversion.
HistoryInfoMode = Add **DiversionMode = Add**	Diversion converted to History-Info.	History-Info converted to Diversion.	Headers are synced and sent.
HistoryInfoMode = Remove **DiversionMode = Remove**	Diversion removed.	History-Info removed.	Both removed.

21.7.3 Interworking SIP REFER Messages

The device supports interworking of SIP REFER messages. SIP UAs may support different versions of the REFER standard while others may not even support REFER.

This feature supports the following:

- Attended, unattended, and semi-attended call transfers
- Sending INVITE, REFER-notifications, BYE, PRACK and Session Timer on behalf of peer PBXs
- Advanced routing rules for the new, initiated INVITE
- Forwarding early media after REFER while attempting to avoid transcoding (by sending session update)
- Interoperate with environments were different SIP UAs lack basic SIP functionality such as re-INVITE, UPDATE, PRACK, Delayed Offer, re-INVITE without SDP
- Session updates after connect to avoid transcoding

The handling of REFER can be configured for all calls, using the global parameter SBCReferBehavior. For configuring different REFER handling options for different UAs (i.e., IP Groups), use the IP Profile table parameter, 'SBC Remote Refer Behavior'.

- Local handling of REFER: This option is used for UAs that do not support REFER. Upon receipt of a REFER request, instead of forwarding it to the IP Group, the device handles it locally. It generates a new INVITE to the alternative destination according to the rules in the IP-to-IP Routing table (where the 'Call Trigger' field is set to **REFER**). It is also possible to specify the IP Group that sent the REFER request, as matching criteria for the re-routing rule in this table ('ReRoute IP Group ID' field).
- Transparent handling: The device forwards the REFER with the Refer-To header unchanged.
- Re-routing through SBC: The device changes the Refer-To header so that the re-routed INVITE is sent through the SBC application.
- IP Group Name: The device sets the host part in the REFER message to the name configured for the IP Group in the IP Group table.

21.7.4 Interworking SIP PRACK Messages

The device supports interworking of SIP Provisional Response ACKnowledgement (PRACK) messages (18x). While some UAs may not support PRACK (RFC 3262) others may require it. The device can be configured to resolve this interoperable issue and enable sessions between such endpoints. SIP PRACK handling is configured using the IP Profile parameter, 'SBC Prack Mode':

- Optional: PRACK is optional for these UAs. If required, the device performs the PRACK process on behalf of the destination UA.

- Mandatory: PRACK is required for these UAs. Calls from UAs that do not support PRACK are rejected. Calls destined to these UAs are also required to support PRACK.
- Transparent (default): The device does not intervene with the PRACK process and forwards the request as is.

21.7.5 Interworking SIP Session Timer

The device supports interworking of the SIP signaling keep-alive mechanism. The SIP standard provides a signaling keep-alive mechanism using re-INVITE and UPDATE messages. In certain setups, keep-alive may be required by some SIP UAs while for others it may not be supported. The device can resolve this mismatch by performing the keep-alive process on behalf of SIP UAs that do not support it.

For configuring the handling of session expires, use the IP Profile parameter, 'SBC Session Expires Mode'.

21.7.6 Interworking SIP Early Media

The device supports various interworking modes for SIP early media between SIP UAs (i.e., IP Groups):

- **Early Media Enabling:** The device supports the interworking of early media between SIP UAs that support early media and those that do not support receipt of early media. Early media can arrive in provisional responses to an INVITE request. The device forwards the request of early media for IP Groups that support this capability; otherwise, the device terminates it. Provisional responses whose SDP are suppressed are changed to a SIP 180 response. This feature is also supported for delayed offers. This is configured using the IP Profile parameter, 'SBC Remote Early Media Support'. The device refers to this parameter also for features that require early media such as playing ringback tone.
- **Early Media Response Type:** The device supports the interworking of different SIP provisional response types between UAs for forwarding the early media to the caller. This can support all early media response types (default), SIP 180 only, or SIP 183 only, and is configured by the IP Profile parameter, 'SBC Remote Early Media Response Type'.
- **Multiple 18x:** The device supports the interworking of different support for multiple 18x responses (including 180 Ringing, 181 Call is Being Forwarded, 182 Call Queued, and 183 Session Progress) that are forwarded to the caller. The UA can be configured as supporting only receipt of the first 18x response (i.e., the device forwards only this response to the caller), or receipt of multiple 18x responses (default). This is configured by the IP Profile parameter, 'SBC Remote Multiple 18x Support'.
- **Early Media RTP:** The device supports the interworking with remote clients that send 18x responses with early media and whose subsequent RTP is delayed, and with remote clients that do not support this and require RTP to immediately follow the 18x response. Some clients do not support 18x with early media, while others require 18x with early media (i.e., they cannot play ringback tone locally). These various interworking capabilities are configured by the IP Profile parameters, 'SBC Remote Early Media RTP', 'SBC Remote Supports RFC 3960', and 'SBC Remote Can Play Ringback'. See the flowcharts below for the device's handling of such scenarios:

Figure 90: SBC Early Media RTP 18x without SDP

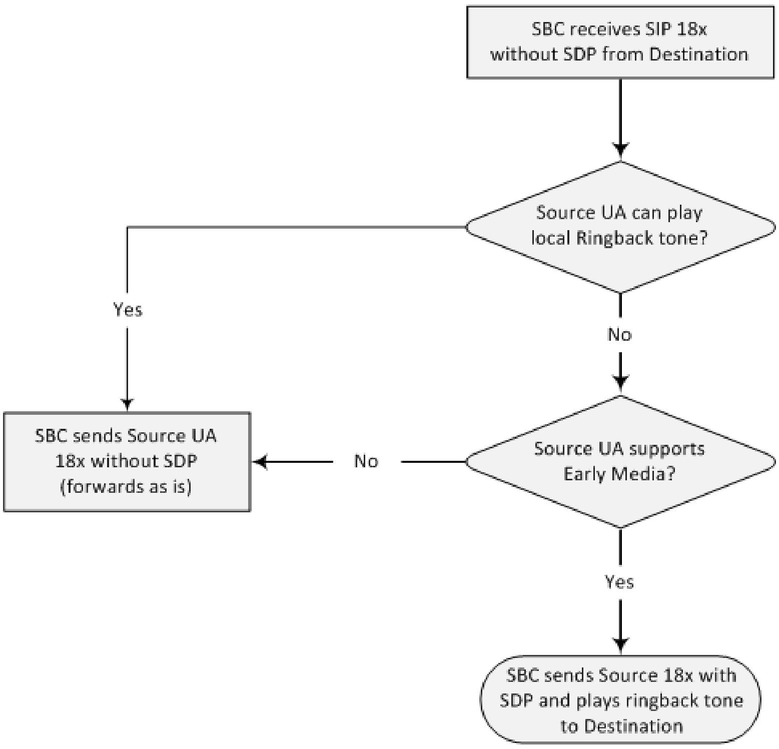
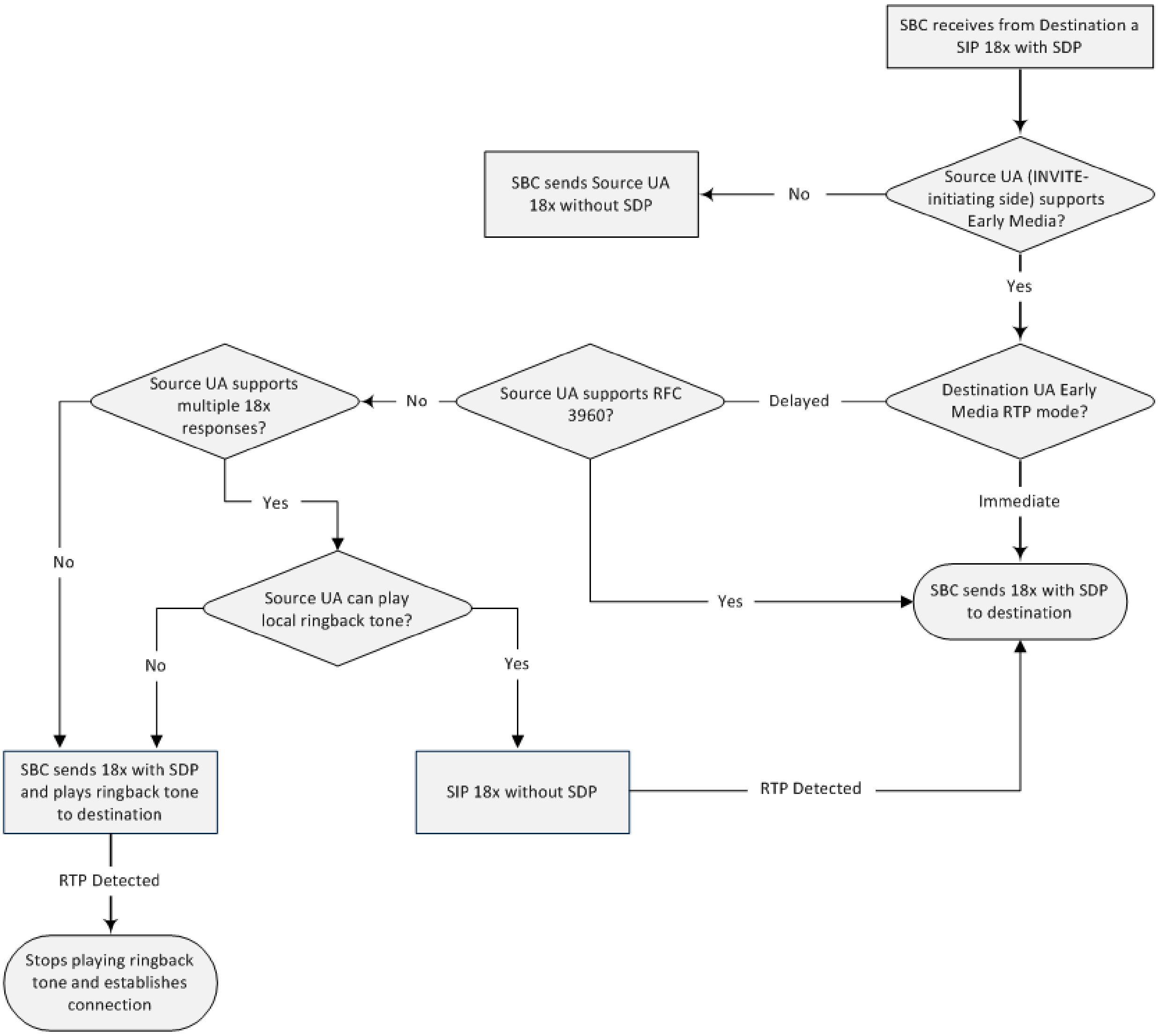

SBC receives SIP 18x without SDP from Destination
Source UA can play local Ringback tone?
Yes
No
SBC sends Source UA 18x without SDP (forwards as is)
No
Source UA supports Early Media?
Yes
SBC sends Source 18x with SDP and plays ringback tone to Destination

Figure 91: Early Media RTP - SIP 18x with SDP


SBC receives from Destination a SIP 18x with SDP
Source UA (INVITE-initiating side) supports Early Media?
No
SBC sends Source UA 18x without SDP
Yes
Destination UA Early Media RTP mode?
Delayed
Source UA supports RFC 3960?
No
Source UA supports multiple 18x responses?
Immediate
Yes
No
Source UA can play local ringback tone?
Yes
SBC sends 18x with SDP to destination
No
Yes
SBC sends 18x with SDP and plays ringback tone to destination
SIP 18x without SDP
RTP Detected
RTP Detected
Stops playing ringback tone and establishes connection

21.7.7 Interworking SIP re-INVITE Messages

The device supports interworking of SIP re-INVITE messages. This enables communication between endpoints that generate re-INVITE requests and those that do not support the receipt of re-INVITEs. The device does not forward re-INVITE requests to IP Groups that do not support it. Instead, it sends a SIP response to the re-INVITE request, which can either be a success or a failure, depending on whether the device can bridge the media between the endpoints. The device can also handle re-INVITEs with or without an SDP body, enabling communication between endpoints that do not support re-INVITE requests without SDP, and those that require SDP. The device generates an SDP offer and adds it to the incoming re-INVITE request if it does not contain an SDP and only then forwards it to the destination endpoint. This interworking support is configured by the IP Profile parameter, 'SBC Remote Reinvite Support'.

21.7.8 Interworking SIP UPDATE Messages

The device supports interworking of the SIP UPDATED message. This enables communication between UAs that generate UPDATE requests and those that do not support the receipt of UPDATE requests. The device does not forward UPDATE requests to IP Groups that do not support it. Instead, it sends a SIP response to the UPDATE request which can either be a success or a failure, depending on whether the device can bridge the media between the endpoints. The handling of UPDATE messages is configured by the IP Profile parameter 'SBC Remote Update Support'.

21.7.9 Interworking SIP re-INVITE to UPDATE

The device enables communication between endpoints (IP Groups) that do not support re-INVITE requests but support the UPDATE method, and vice versa. The device translates the re-INVITE request to the UPDATE request, and vice versa. Note that if a re-INVITE request arrives without SDP, the device generates the SDP and adds it to the outgoing UPDATE request. To enable this feature, each IP Group needs to be configured with its unique capabilities by associating it with a relevant IP Profile. For example, an IP Group that supports UPDATE requests but not re-INVITEs would be configured as follows:

- SBCRemoteUpdateSupport = 2 (Supported)
- SBCRemoteReinviteSupport = 0 (Not Supported)

If a re-INVITE request needs to be forwarded to this IP Group, it is translated to an UPDATE request.

21.7.10 Interworking Delayed Offer

The device enables sessions between endpoints (IP Groups) that send INVITEs without SDP (i.e., delayed media) and those that do not support the receipt of INVITEs without SDP. The device creates an SDP and adds it to INVITEs that arrive without SDP. Delayed offer is also supported when early media is present.

The interworking of delayed offer is configured using the IP Profile parameter 'SBC Remote Delayed Offer Support'.

21.7.11 Interworking Call Hold

The device supports the interworking of call hold / retrieve requests between IP entities supporting different call hold capabilities:

- Interworking SDP call hold formats. This is configured by the IP Profile parameter, 'SBC Remote Hold Format'.
- Interworking the play of the held tone for IP entities that cannot play held tones locally. This is configured by the IP Profile parameter, 'SBC Play Held Tone'.
- Interworking generation of held tone where the device generates the tone to the held party instead of the call hold initiator. This is configured by the IP Profile parameter, 'SBC Reliable Held Tone Source'.

For configuring IP Profiles, see "Configuring IP Profiles" on page 201.

21.8 Call Survivability

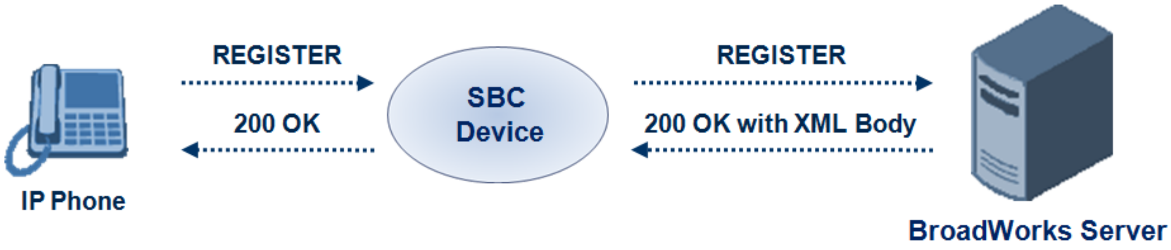
This section describes various call survivability features supported by the SBC device.

21.8.1 Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server for Survivability

This feature enables SBC user registration for interoperability with BroadSoft BroadWorks server to provide call survivability in case of connectivity failure with the BroadWorks server, for example, due to a WAN failure. This feature enables local users to dial a local extension (or any other configured alias) that identifies another local user, in survivability mode. This feature is enabled using the SBCExtensionsProvisioningMode parameter.

In normal operation, when subscribers (such as IP phones) register to the BroadWorks server through the device, the device includes the SIP Allow-Events header in the sent REGISTER message. In response, the BroadWorks server sends the device a SIP 200 OK containing an XML body with subscriber information such as extension number, phone number, and URIs (aliases). The device forwards the 200 OK to the subscriber (without the XML body).

Figure 92: Interoperability with BroadWorks Registration Process



The device saves the users in its registration database with their phone numbers and extensions, enabling future routing to these destinations during survivability mode. When in survivability mode, the device routes the call to the Contact associated with the dialed phone number or extension number in the registration database.

Below is an example of an XML body received from the BroadWorks server:

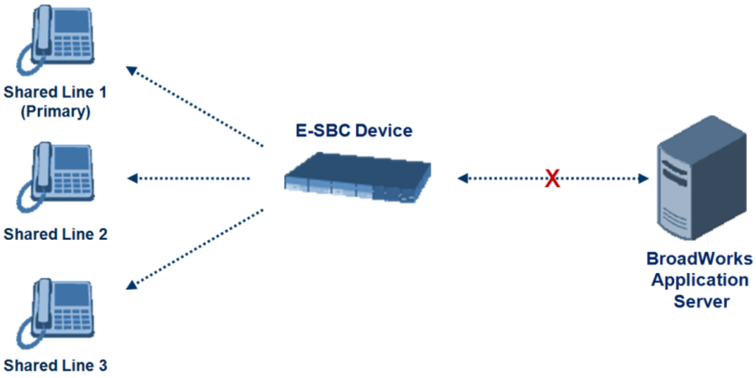
```
<?xml version="1.0" encoding="utf-8"?>
  <BroadsoftDocument version="1.0" content="subscriberData">
    <phoneNumbers>
      <phoneNumber>2403645317</phoneNumber>
      <phoneNumber>4482541321</phoneNumber>
    </phoneNumbers>
    <aliases>
      <alias>sip:bob@broadsoft.com</alias>
      <alias>sip:rhughes@broadsoft.com</alias>
    </aliases>
    <extensions>
      <extension>5317</extension>
      <extension>1321</extension>
    </extensions>
  </BroadSoftDocument>
```

21.8.2 BroadSoft's Shared Phone Line Call Appearance for SBC Survivability

The device can provide redundancy for BroadSoft's Shared Call Appearance feature. When the BroadSoft application server switch (AS) fails or does not respond, or when the network connection between the device and the BroadSoft AS is down, the device manages the Shared Call Appearance feature for the SIP clients.

This feature is supported by configuring a primary extension and associating it with secondary extensions (i.e., *shared lines*) so that incoming calls to the primary extension also ring at the secondary extensions. The call is established with the first extension to answer the call and consequently, the ringing at the other extensions stop. For example, assume primary extension number 600 is shared with secondary extensions 601 and 602. In the case of an incoming call to 600, all three phone extensions ring simultaneously, using the device's call forking feature as described in "SIP Forking Initiated by SIP Proxy Server" on page 239. Note that incoming calls specific to extensions 601 or 602 ring only at these specific extensions.

Figure 93: Call Survivability for BroadSoft's Shared Line Appearance



To configure this capability, you need to configure a shared-line, inbound manipulation rule for registration requests to change the destination number of the secondary extension numbers (e.g. 601 and 602) to the primary extension (e.g., 600). Call forking must also be enabled. The procedure below describes the main configuration required.

Notes:

- The device enables outgoing calls from all equipment that share the same line simultaneously (usually only one simultaneous call is allowed per a specific shared line).
- You can configure whether REGISTER messages from secondary lines are terminated on the device or forwarded transparently (as is), using the SBCSharedLineRegMode parameter.
- The LED indicator of a shared line may display the wrong current state.

➢ To configure the Shared Line feature:

1. In the IP Group table (see "Configuring IP Groups" on page 174), add a Server-type IP Group for the BroadWorks server.
2. In the IP Group table, add a User-type IP Group for the IP phone users and set the 'SBC Client Forking Mode' parameter to **Parallel** so that the device forks incoming calls to all contacts under the same AOR registered in the device's registration database.
3. In the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 255), add a rule for routing calls between the above configured IP Groups.
4. In the IP to IP Inbound Manipulation table (see "Configuring IP-to-IP Inbound Manipulations" on page 266), add a manipulation rule for the secondary extensions (e.g., 601 and 602) so that they also register to the device's database under the primary extension contact (e.g., 600):
   - Set the 'Manipulation Purpose' field to **Shared Line**.
   - Set the 'Source IP Group' field to the IP Group ID that you created for the users (e.g., 2).
   - Set the 'Source Username Prefix' field to represent the secondary extensions (e.g., 601 and 602).
   - Set the 'Manipulated URI' field to **Source** to manipulate the source URI.
   - Set the 'Remove From Right' field to "1" to remove the last digit of the extensions (e.g., 601 is changed to 60).

- Set the 'Suffix to Add' field to "0" to add 0 to the end of the manipulated number (e.g., 60 is changed to 600).

21.8.3 Call Survivability for Call Centers

The device supports call survivability for call centers. When a communication failure (e.g., in the network) occurs with the remote voice application server responsible for handling the call center application (such as IVR), the device routes the incoming calls received from the customer (i.e., from the TDM gateway) to the call center agents.

In normal operation, the device registers the agents in its users registration database. Calls received from the TDM gateway are forwarded by the device to the application server, which processes the calls and sends them to specific call center agents, through the device. Upon a failure with the application server, the device routes the calls from the TDM Gateway to the agents. The device routes the call to the first available user it founds. If the call is not answered by the user, the device routes it to the next available user. The SBC can handle a sequence of up to five users, after which the session is timed out and the call is dropped.

Figure 94: Normal Operation in Call Center Application

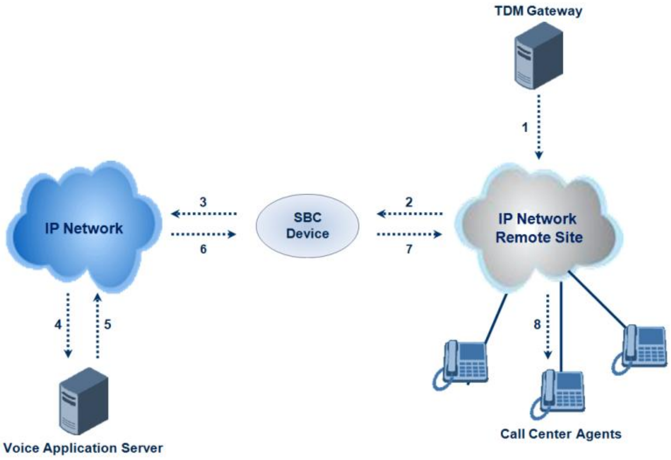
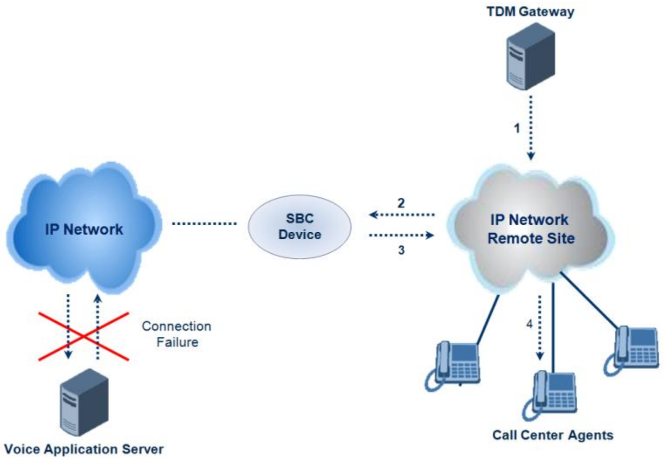


Figure 95: Call Survivability for Call Center

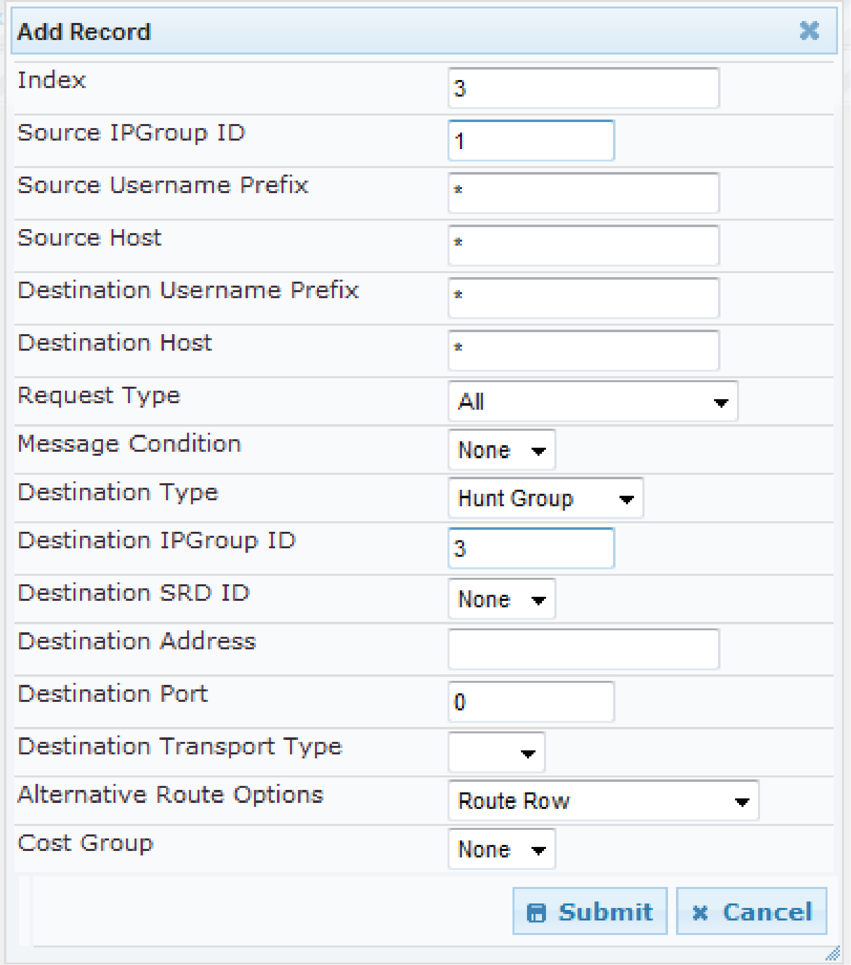


➢ To configure call survivability for a call center application:

1. In the IP Group table (see "Configuring IP Groups" on page 174), add IP Groups for the following entities:
   - TDM Gateway (Server-type IP Group). This entity forwards the customer calls through the device to the Application server.
   - Application server (Server-type IP Group). This entity processes the call and sends the call through the device to the specific call center agent located on a different network (remote).
   - Call center agents (User-type IP Group). You can configure multiple IP Groups to represent different groups of call center agents, for example, agents and managers.
2. In the Classification table (see "Configuring Classification Rules" on page 249), add rules to classify incoming calls that are received from the entities listed in Step 1, to IP Groups.
3. In the SBC IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 255), add the following IP-to-IP routing rules:
   - For normal operation:
     - Routing from TDM Gateway to Application server.
     - Routing from Application server to call center agents.
   - For call survivability mode: Routing from TDM Gateway to call center agents. This configuration is unique due to the following settings:
     - The 'Source IP Group ID' field is set to the IP Group of the TDM Gateway.
     - The 'Destination Type' field is set to **Hunt Group**, which is specifically used for call center survivability.
     - The 'Destination IP Group ID' field is set to the IP Group of the call center agents.

   The figure below displays a routing rule example, assuming IP Group "1" represents the TDM Gateway and IP Group "3" represents the call center agents:

Figure 96: Routing Rule Example for Call Center Survivability



Add Record

Index	3
Source IPGroup ID	1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Request Type	All
Message Condition	None
Destination Type	Hunt Group
Destination IPGroup ID	3
Destination SRD ID	None
Destination Address	
Destination Port	0
Destination Transport Type	
Alternative Route Options	Route Row
Cost Group	None

21.8.4 Survivability Mode Display on Aastra IP Phones

If the SBC device is deployed in an Enterprise network with Aastra IP phones and connectivity with the WAN fails, the device provides call survivability by enabling communication between IP phone users within the LAN enterprise. In such a scenario, the device can be configured to notify the IP phones that it is currently operating in Survivability mode. When this occurs, the Aastra IP phones display the message, "StandAlone Mode" on their LCD screens. This feature is enabled by setting the SBCEnableAASTRASurvivabilityNotice parameter to 1.

When this feature is enabled and the SBC device is in Survivability mode, it responds to SIP REGISTER messages from the IP phones with a SIP 200 OK containing the following XML body:

```
Content-Type: application/xml
<?xml version="1.0" encoding="utf-8"?>
<LMIDocument version="1.0">
<LocalModeStatus>
   <LocalModeActive>true</LocalModeActive>
   <LocalModeDisplay>StandAlone Mode</LocalModeDisplay>
</LocalModeStatus>
</LMIDocument>
```

21.9 Call Forking

This section describes various Call Forking features supported by the device.

21.9.1 Initiating SIP Call Forking

The SBC device supports call forking of an incoming call to multiple SBC users (destinations). Call forking is supported by the device's capability of registering multiple SIP client user phone contacts (mobile and fixed-line extensions) under the same Address of Record (AOR) in its registration database. This feature can be implemented in the following example scenarios:

- An enterprise Help Desk, where incoming customer calls are simultaneously sent to multiple customer service agent extensions.
- An employee's phone devices, where the incoming call is simultaneously sent to multiple devices (e.g., to the employee's office phone and mobile SIP phone).
- An enterprise reception desk, where an incoming call is simultaneously sent to multiple receptionists.

The device supports various modes of call forking. For example, in Parallel call forking mode, the device sends the INVITE message simultaneously to all the users registered under the same AOR, resulting in the ringing of all extensions; the first extension to pick up the call receives the call, and all other extensions stop ringing. The Call Forking feature is configured by creating a User-type IP Group and configuring the IP Group table's parameter, 'SBC Client Forking Mode' (see "Configuring IP Groups" on page 174).

The device can also fork INVITE messages received for a Request-URI of a specific contact (user), belonging to the destination IP Group User-type, registered in the database to all other users located under the same AOR as the specific contact. This is configured using the SBCSendInviteToAllContacts parameter.

21.9.2 SIP Forking Initiated by SIP Proxy Server

The device can handle SIP forking responses received from a proxy server in response to an INVITE forwarded by the device from a UA. In other words, received responses with a different SIP To header 'tag' parameter for the request forwarded by the device. This occurs in scenarios, for example, where a proxy server forks the INVITE request to several UAs, and therefore, the SBC device may receive several replies for a single request. Forked SIP responses may result in a single SDP offer with two or more SDP answers during call setup. The SBC handles this scenario by "hiding" the forked responses from the INVITE-initiating UA. This is achieved by marking the UA that responded first to the INVITE as the active UA, and only requests/responses from that UA are subsequently forwarded. All other requests/responses from other UAs are handled by the SBC (SDP offers from these users are answered with an 'inactive' media).

The SBC supports two forking modes, configured by the SBCForkingHandlingMode parameter:

- Latch On First - only the first received 18x response is forwarded to the INVITE initiating UA, and disregards any subsequently received 18x forking responses (with or without SDP).
- Sequential - all 18x responses are forwarded to the INVITE initiating UA, one at a time in a sequential manner. If 18x arrives with an offer only, only the first offer is forwarded to the INVITE initiating UA.

The SBC also supports media synchronization for call forking. If the active UA is the first one to send the final response (e.g., 200 OK) the call is established and all other final responses are acknowledged and a BYE is sent if needed. If another UA sends the first final response, then it is possible that the SDP answer that was forwarded to the INVITE-initiating UA is not relevant, and media synchronization is needed between the two UAs. Media synchronization is done by sending a re-INVITE request immediately after the call is established. The re-INVITE is sent without an offer to the INVITE-initiating UA. This causes the UA to send an offer which is forwarded to the UA that confirmed the call. The media synchronization process is enabled by the EnableSBCMediaSync parameter.

21.9.3 Call Forking-based IP-to-IP Routing Rules

You can configure call forking routing rules in the IP-to-IP Routing table. This is done by configuring multiple routing rules under a forking group. These rules send an incoming IP call to multiple destinations of any type (e.g., IP Group or IP address). The device forks the call by sending simultaneous INVITE messages to all the specified destinations. It handles the multiple SIP dialogs until one of the calls is answered and then terminates the other SIP dialogs. For more information, see "Configuring SBC IP-to-IP Routing Rules" on page 255.

21.10 Alternative Routing on Detection of Failed SIP Response

The device can detect failure of a sent SIP response (e.g., TCP timeout, and UDP ICMP). In such a scenario, the device re-sends the response to an alternative destination. This support is in addition to alternative routing if the device detects failed SIP requests.

For example, assume the device sends a SIP 200 OK in response to a received INVITE request. If the device does not receive a SIP ACK in response to this, it sends a new 200 OK to the next alternative destination. This new destination can be the next given IP address resolved from a DNS from the Contact or Record-Route header in the request related to the response.

22 SBC Configuration

This chapter describes the configuration of the SBC application.

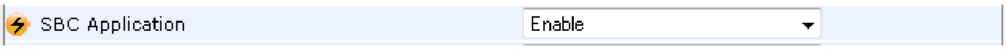
22.1 Enabling the SBC Application

Before you can start configuring the SBC, you must first enable the SBC application. Once enabled, the Web interface displays the menus and parameter fields relevant to the SBC application.

Note: The SBC feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 309.

➢ To enable the SBC application:

1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** > **Applications Enabling**).



SBC Application: Enable

2. From the 'SBC Application' drop-down list, select **Enable**.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

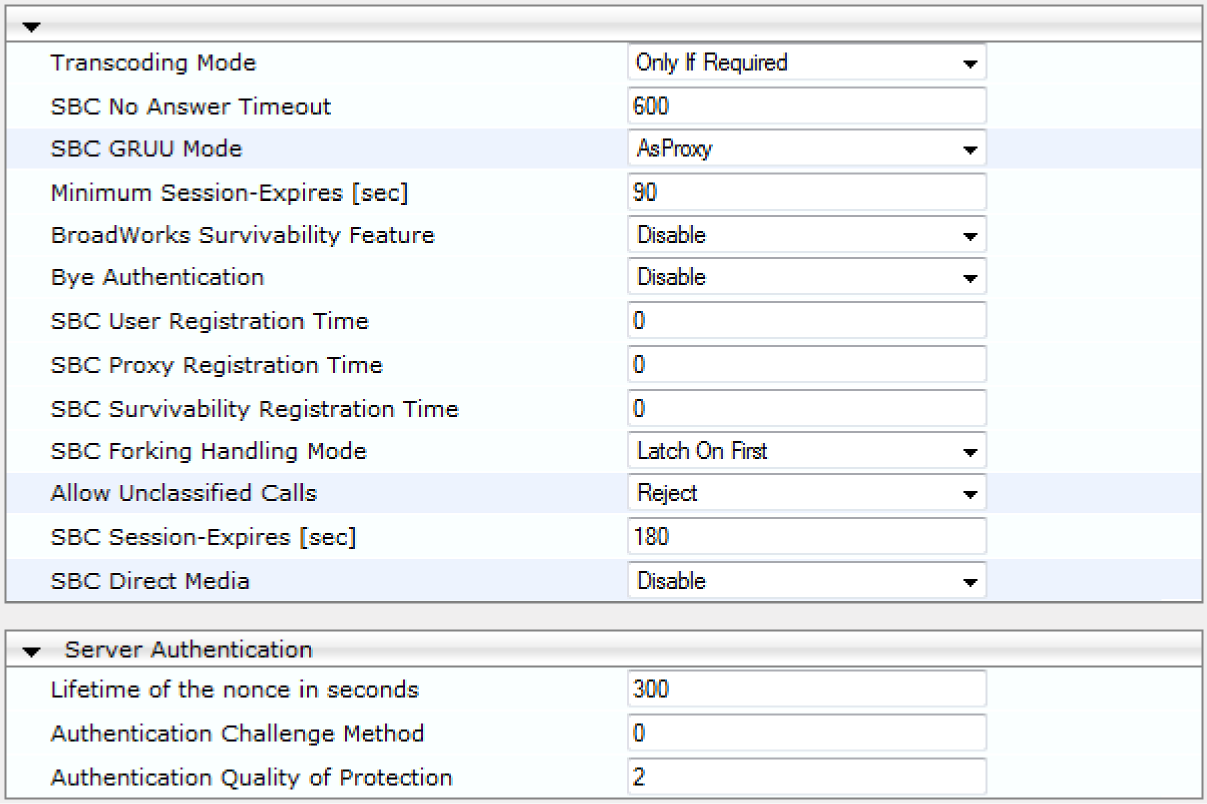
22.2 Configuring General Settings

The General Settings page allows you to configure general SBC parameters. For a description of these parameters, see "SBC Parameters" on page 444.

➢ To configure general parameters:

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **SBC** > **General Settings**).

Figure 97: General Settings Page



Transcoding Mode	Only If Required
SBC No Answer Timeout	600
SBC GRUU Mode	AsProxy
Minimum Session-Expires [sec]	90
BroadWorks Survivability Feature	Disable
Bye Authentication	Disable
SBC User Registration Time	0
SBC Proxy Registration Time	0
SBC Survivability Registration Time	0
SBC Forking Handling Mode	Latch On First
Allow Unclassified Calls	Reject
SBC Session-Expires [sec]	180
SBC Direct Media	Disable

Server Authentication

Lifetime of the nonce in seconds	300
Authentication Challenge Method	0
Authentication Quality of Protection	2

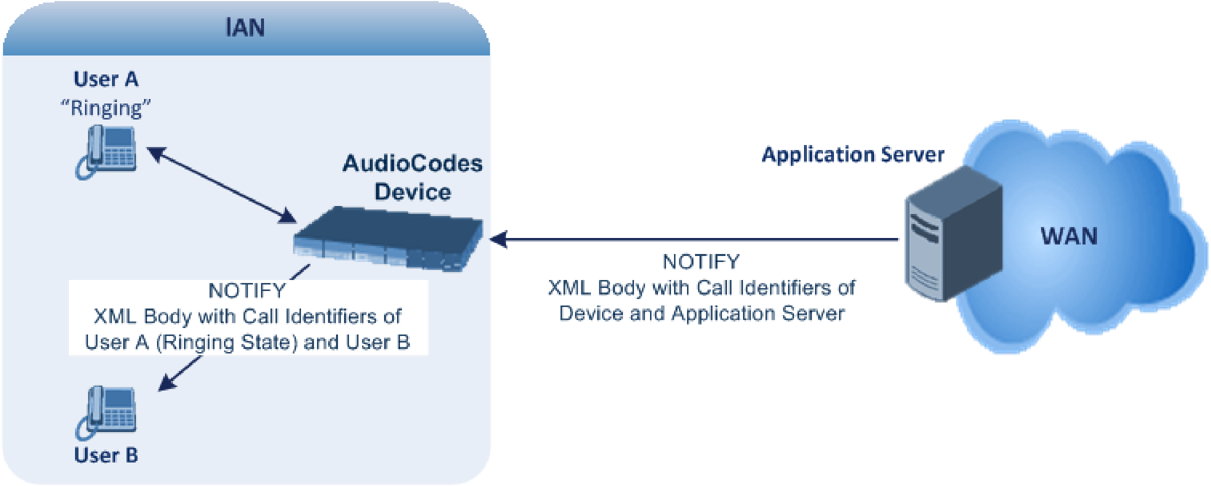
2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 295.

22.2.1 Interworking Dialog Information in SIP NOTIFY Messages

You can enable the device to interwork dialog information (XML body) received in SIP NOTIFY messages from a remote (WAN) application server. The NOTIFY message is sent by application servers to notify a SIP client, subscribed to a service and located behind the device (LAN), of the status of another SIP client in the LAN. For example, user B can subscribe to an application server for call pick-up service, whereby if user A's phone rings, the application server notifies user B. User B can then press a pre-configured key sequence to answer the call.

The NOTIFY message contains the XML body with call identifiers (call-id and tags). However, as the application server is located in the external network WAN and the SIP clients behind the device, the call dialog information sent by the application server reflects only the dialog between the device and itself; not that of the involved SIP clients. This is due to, for example, the device's topology hiding (e.g., IP address) of its LAN elements. The device resolves this by replacing the call identifiers received from the application server with the correct call identifiers (e.g., user A and user B). Thus, users subscribed to the service can receive relevant NOTIFY messages from the device and use the service.

Figure 98: Interworking NOTIFY XML Body for Application Server


IAN
User A
"Ringing"
AudioCodes Device
NOTIFY
XML Body with Call Identifiers of User A (Ringing State) and User B
User B
Application Server
WAN
NOTIFY
XML Body with Call Identifiers of Device and Application Server

To enable this feature, set the 'SBC Dialog-Info Interworking' (EnableSBCDialogInfoInterworking) parameter to **Enable**. When this feature is disabled, the device forwards the NOTIFY message as is, without modifying its XML body.

Below is an example of an XML body where the call-id, tags, and URIs have been replaced by the device:

```
<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
version="10" state="partial"
entity="sip:alice@example.com">
<dialog id="zxcvbnm3" call-id="67402270@10.132.10.150"
local-tag="1c137249965"
remote-tag="CCDORRTDRKIKWFVBRWYM" direction="initiator">
<state event="replaced">terminated</state>
</dialog>
<dialog id="sfhjsjk12" call-id="67402270@10.132.10.150"
local-tag="1c137249965"
remote-tag="CCDORRTDRKIKWFVBRWYM" direction="receiver">
<state reason="replaced">confirmed</state>
<replaces
call-id="67402270@10.132.10.150"
local-tag="1c137249965"
remote-tag="CCDORRTDRKIKWFVBRWYM"/>
<referred-by>
sip:bob-is-not-here@vm.example.net
</referred-by>
<local>
<identity display="Jason Forster">
sip:jforsters@home.net
</identity>
<target uri="sip:alice@pc33.example.com">
<param pname="+sip.rendering" pval="yes"/>
</target>
</local>
<remote>
<identity display="Cathy Jones">
sip:cjones@example.net
</identity>
<target uri="sip:line3@host3.example.net">
<param pname="actor" pval="attendant"/>
<param pname="automaton" pval="false"/>
</target>
</remote>
</dialog>
</dialog-info>
```

22.3 Configuring Admission Control

The Admission Control table lets you configure up to 200 Call Admission Control rules (CAC). CAC rules define the maximum number of concurrent calls (SIP dialogs) permitted per IP Group or SRD, and per user (identified by its registered contact) belonging to these entities. CAC rules also define a guaranteed (*reserved*) number of concurrent calls. Thus, CAC rules can be useful for implementing Service Level Agreements (SLA) policies.

CAC rules can be applied per SIP request type and SIP dialog direction (inbound and/or outbound). These relate to requests that initiate SIP dialogs and not the subsequent requests that can be of different type and direction. The SIP dialog-initiating request types can include INVITE, REGISTER, and/or SUBSCRIBE messages, or it can be configured to include the total number of all dialogs.

This feature also provides support for SIP-dialog rate control, using the "token bucket" mechanism. The token bucket is a control mechanism that dictates the rate of SIP-dialog setups based on the presence of tokens in the bucket – a logical container that holds aggregate SIP dialogs to be accepted or transmitted. Tokens in the bucket are removed ("cashed in") for the ability to setup a dialog. Thus, a flow can setup dialogs up to its peak burst rate if there are adequate tokens in the bucket and if the burst threshold is configured appropriately:

- Every SIP dialog setup request must attempt to take a token from the bucket.
- If there are no tokens, the request is dropped.
- New tokens are added to the bucket at a user-defined rate (token rate).
- If the bucket contains the maximum number of tokens, tokens to be added at that moment are dropped.

Reserved capacity is especially useful when the device operates with multiple SIP entities such as in a contact center environment handling multiple customers. For example, if the total call capacity of the device is 200 call sessions, a scenario may arise where one SIP entity may reach the maximum configured call capacity of 200 and thereby, leaving no available call resources for the other SIP entities. Thus, reserved capacity guarantees a minimum capacity for each SIP entity. If the reserved call capacity of a SIP entity is threatened by a new call for a different SIP entity, the device rejects the call to safeguard the reserved capacity.

Reserved call capacity can be configured for both an SRD and each of its associated IP Groups. In such a setup, the SRD's reserved call capacity must be greater or equal to the summation of the reserved call capacity of all these IP Groups. In other words, the SRD serves as the "parent" reserved call capacity. If the SRD's reserved call capacity is greater, the extra call capacity can be used as a shared pool between the IP Groups for unreserved calls when they exceed their reserved capacity. For example, assume that the reserved capacities for an SRD and its associated IP Groups are as follows:

- SRD reserved call capacity: 40
- IP Group ID 1 reserved call capacity: 10
- IP Group ID 2 reserved call capacity: 20

In this setup, the SRD offers a shared pool for unreserved call capacity of 10 [i.e., 40 – (10 + 20)]. If IP Group ID 1 needs to handle 15 calls, it is guaranteed 10 calls and the remaining 5 is provided from the SRD's shared pool. If the SDR's shared pool is currently empty and resources for new calls are required, the quota is taken from the device's total capacity, if available. For example, if IP Group ID 1 needs to handle 21 calls, it's guaranteed 10, the SRD's shared pool provides another 10, and the last call is provided from the device's total call capacity support (e.g., of 200).

Requests that reach the user-defined call limit (maximum concurrent calls and/or call rate) are sent to an alternative route, if configured in the IP-to-IP Routing table. If no alternative routing rule is located, the device rejects the SIP request with a SIP 486 "Busy Here" response.

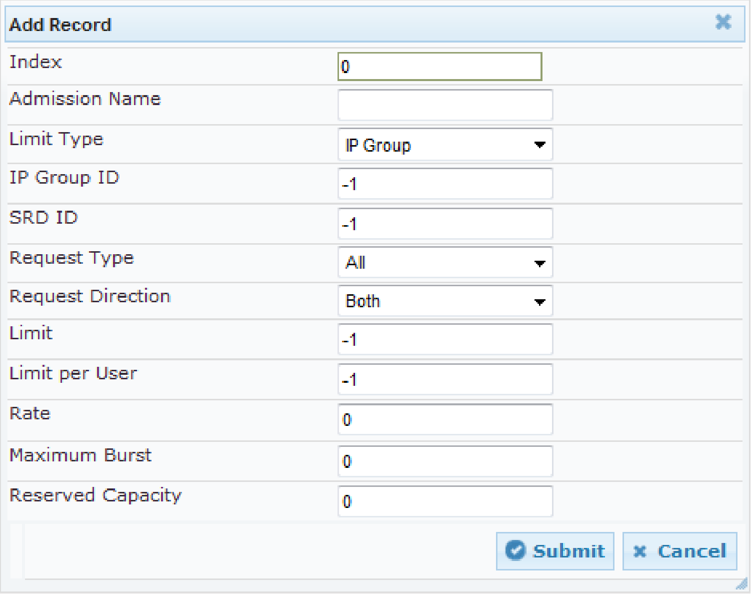
Note: The device applies the CAC rule for the incoming leg immediately after the Classification process. If the call/request is rejected at this stage, no routing is performed. The enforcement for the outgoing leg is performed within each alternative route iteration. This is accessed from two places: one during initial classification/routing, and another during alternative routing process.

The procedure below describes how to configure CAC rules in the Web interface. You can also configure CAC rules using the table ini file parameter, SBCAdmissionControl or CLI command, configure voip > sbc sbc-admission-control.

➢ To configure a CAC rule:

1. Open the Admission Control page (**Configuration** tab > **VoIP** menu > **SBC** > **Admission Control**).
2. Click **Add**; the following dialog box appears:

Figure 99: Admission Control Page - Add Record Dialog Box



Add Record

Index	0
Admission Name	
Limit Type	IP Group
IP Group ID	-1
SRD ID	-1
Request Type	All
Request Direction	Both
Limit	-1
Limit per User	-1
Rate	0
Maximum Burst	0
Reserved Capacity	0

3. Configure an Admission Control rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 68: Admission Control Table Parameter Description

Parameter	Description
Index **[SBCAdmissionControl_Index]**	Defines an index number for the new table record.
Admission Name CLI: admission-name **[SBCAdmissionControl_AdmissionControlName]**	Defines an arbitrary name to easily identify the Admission Control rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Limit Type CLI: limit-type **[SBCAdmissionControl_LimitType]**	Defines the entity to which the rule applies. ▪ **[0]** IP Group (default) ▪ **[1]** SRD
IP Group ID CLI: ip-group-id **[SBCAdmissionControl_IPGroupID]**	Defines the IP Group to which you want to apply the rule. The default value is -1 (i.e., all IP Groups). **Note:** This parameter is applicable only if 'Limit Type' is set to **IP Group**.
SRD ID CLI: srd-id **[SBCAdmissionControl_SRDID]**	Defines the SRD to which you want to apply the rule. The default value is -1 (i.e., all SRDs). **Note:** This parameter is applicable only if 'Limit Type' is set to **SRD**.

Parameter	Description
Request Type CLI: request-type **[SBCAdmissionControl_RequestType]**	Defines the SIP dialog-initiating request type to which you want to apply the rule (not the subsequent requests that can be of different type and direction). ▪ **[0]** All = (Default) Includes the total number of all dialogs. ▪ **[1]** INVITE ▪ **[2]** SUBSCRIBE ▪ **[3]** Other
Request Direction CLI: request-direction **[SBCAdmissionControl_RequestDirection]**	Defines the direction of the SIP request to which the rule applies. ▪ **[0]** Both = (Default) Rule applies to inbound and outbound SIP dialogs. ▪ **[1]** Inbound = Rule applies only to inbound SIP dialogs. ▪ **[2]** Outbound = Rule applies only to outbound SIP dialogs.
Limit CLI: limit **[SBCAdmissionControl_Limit]**	Defines the maximum number of concurrent SIP dialogs per IP Group or SRD. You can also use the following special values: ▪ **[0]** 0 = Block all these dialogs. ▪ **[-1]** -1 = (Default) Unlimited.
Limit Per User CLI: limit-per-user **[SBCAdmissionControl_LimitPerUser]**	Defines the maximum number of concurrent SIP dialogs per user belonging to the specified IP Group or SRD. You can also use the following special values: ▪ **[0]** 0 = Block all these dialogs. ▪ **[-1]** -1 = (Default) Unlimited.
Rate CLI: rate **[SBCAdmissionControl_Rate]**	Defines the rate at which tokens are added to the token bucket per second (i.e., token rate). One token is added to the bucket every 1000 divided by the value of this parameter (in milliseconds). The default is 0 (i.e., unlimited rate). **Note:** The token bucket feature is per IP Group, SRD, SIP request type, and SIP request direction.
Maximum Burst CLI: max-burst **[SBCAdmissionControl_MaxBurst]**	Defines the maximum number of tokens (SIP dialogs) that the bucket can hold. The device only accepts a SIP dialog if a token exists in the bucket. Once the SIP dialog is accepted, a token is removed from the bucket. If a SIP dialog is received by the device and the token bucket is empty, then the device rejects the SIP dialog. Alternatively, if the bucket is full, for example, 100 tokens, and 101 SIP dialogs arrive (before another token is added to the bucket, i.e., faster than that defined in the Rate field), then the device accepts the first 100 SIP dialogs and rejects the last one. Dropped requests are replied with the SIP 486 "Busy Here" response. Dropped requests are not counted in the bucket. The default is 0 (i.e., unlimited SIP dialogs). **Note:** The token bucket feature is per IP Group, SRD, SIP request type, and SIP request direction.
Reservation CLI: reservation **[SBCAdmissionControl_Reservation]**	Defines the guaranteed (minimum) call capacity. **Notes:** ▪ An IP Group ID or SRD ID must be specified when this parameter is configured and the IP Group or SRD cannot be set to all (-1). ▪ Reserved call capacity is applicable only to INVITE and SUBSCRIBE messages. ▪ Reserved call capacity must be less than the maximum capacity (limit) configured for the CAC rule. ▪ The total reserved call capacity configured for all the CAC rules must be within the device's total call capacity support.

22.4 Configuring Allowed Audio Coder Groups

The Allowed Audio Coders Group table lets you configure up to five Allowed Audio Coders Groups. An Allowed Audio Coders Group defines a list of audio media coders that can be used for a specific SIP entity. Each Allowed Audio Coders Group can be configured with up to 10 coders. The coders can include pre-defined audio coders (according to the installed Software License Key) and user-defined (string) coders for non-standard or unknown coders.

Allowed Audio Coders Groups are assigned to SIP entities, using IP Profiles (see "Configuring IP Profiles" on page 201). Coders that are not listed in the Allowed Audio Coders Group are removed from the SDP offer ('a=rtpmap' field) that is sent to the SIP entity. Only coders that are common between the coders in the SDP offer and the coders listed in the Allowed Audio Coders Group are used. Thus, Allowed Audio Coders Groups enable you to enforce the use of only specified coders. For more information, see "Restricting Coders" on page 223.

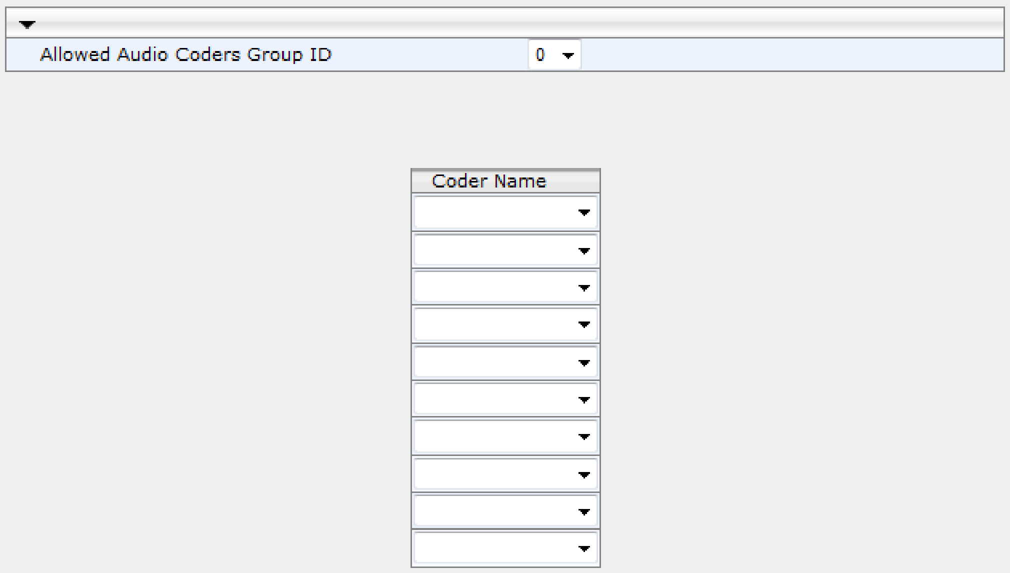
The order of appearance of the coders listed in the Allowed Audio Coders Group determines the priority (preference) of the coders in the SDP offer. The device arranges the SDP offer's coder list according to their order in the Allowed Audio Coders Group. The priority is in descending order, whereby the first coder in the list is given the highest priority and the last coder, the lowest priority. For more information, see "Prioritizing Coder List in SDP Offer" on page 224.

The procedure below describes how to configure Allowed Audio Coder Groups in the Web interface. You can also configure Allowed Audio Coder Groups using the table ini file parameter, AllowedCodersGroup or CLI command, configure voip > sbc allowed-coders-group group-0.

➢ To configure an Allowed Coders Group:

1. Open the Allowed Audio Coders Group page (**Configuration** tab > **VoIP** menu > **SBC** > **Allowed Audio Coders Group**).

Figure 100: Allowed Audio Coders Group Page



Allowed Audio Coders Group ID: 0

2. Configure an Allowed Audio Coders Group according to the parameters described in the table below.
3. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

Table 69: Allowed Audio Coders Group Table Parameter Descriptions

Parameter	Description
Allowed Coders Group ID **[AllowedCodersGroupX]**	Defines an index number for the new table record.

Parameter	Description
Coder Name CLI: name **[AllowedCodersGroupX_Name]**	Defines the audio coder. This can be a pre-defined coder or a user-defined coder. The valid value for user-defined coders is a string of up to 25 characters (case-insensitive). For example, "HD.123" (without quotes). **Note:** Each coder type (e.g., G.729) can be configured only once per Allowed Coders Group.

22.5 Configuring Allowed Video Coder Groups

The Allowed Video Coders Group table lets you configure up to four Allowed Video Coders Groups. An Allowed Video Coders Group defines a list of video coders that can be used when forwarding video streams to a specific SIP entity. Each Allowed Video Coders Group can be configured with up to 20 coders. The coders can include pre-defined video coders (according to the installed Software License Key) and user-defined (string) video coders for non-standard or unknown coders. Allowed Video Coders Groups are assigned to SIP entities, using IP Profiles (see "Configuring IP Profiles" on page 201). The video coders appear in the SDP media type "video" ('m=video' line). Coders that are not listed in the Allowed Video Coders Group are removed from the SDP offer that is sent to the SIP entity. Only coders that are common between the coders in the SDP offer and the coders listed in the Allowed Video Coders Group are used. Thus, Allowed Video Coders Groups enable you to enforce the use of only specified coders. For more information, see "Restricting Coders" on page 223.

The order of appearance of the coders listed in the Allowed Video Coders Group determines the priority (preference) of the coders in the SDP offer. The device arranges the SDP offer's coder list according to their order in the Allowed Video Coders Group. The priority is in descending order, whereby the first coder in the list is given the highest priority and the last coder, the lowest priority. For more information, see "Prioritizing Coder List in SDP Offer" on page 224.

Currently, the Allowed Video Coder Groups table can only be configured using the ini file parameter, AllowedVideoCodersGroup or CLI command, configure voip/sbc allowed-video-coders-group group-0. The table below describes this parameter.

Table 70: Allowed Video Coders Group Table Parameter Descriptions

Parameter	Description
Allowed Coders Group ID **[AllowedVideoCodersGroupX]**	Defines an index number for the new table record.
Coder Name CLI: name **[AllowedVideoCodersGroupX_Name]**	Defines the video coder. This can be a pre-configured coder or a user-defined coder. The valid value for user-defined coders is a string of up to 25 characters (case-insensitive). For example, "WOW.789" (but without quotes). **Note:** Each coder type can be configured only once per Allowed Video Coders Group.

22.6 Routing SBC

This section describes the configuration of the routing entities for the SBC application. These include the following:

- Classification rules - see "Configuring Classification Rules" on page 249
- Message Condition rules - see "Configuring Message Condition Rules" on page 254
- IP-to-IP routing rules - see "Configuring SBC IP-to-IP Routing Rules" on page 255
- Alternative routing reasons - see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 262

22.6.1 Configuring Classification Rules

The Classification table lets you configure up to 200 Classification rules. A Classification rule classifies incoming SIP dialog-initiating requests (e.g., INVITE messages) to an IP Group from where the SIP dialog request was received. The identified IP Group is then used in the manipulation and routing processes. Classification rules also enhance security by allowing you to create a SIP access list, whereby classified calls can be denied (i.e., blacklist) or allowed (i.e., whitelist).

A Classification rule includes two areas:

- Matching characteristics (**Rule**) - characteristics of incoming IP call such as source IP address.
- Operation (**Action**) - if the incoming call matches the characteristics of the rule, the device classifies the call as belonging to an IP Group.

The Classification table is used to classify incoming SIP dialog requests only if the following classification stages fail:

1. **Classification Stage 1 - Registered Users Database:** The device searches its registration database to check if the incoming SIP dialog arrived from a registered user:
   - Compares the SIP Contact header of the received SIP dialog to the Contact of the registered user.
   - Compares the URL in the SIP P-Asserted-Identity/From header to the registered address-of-record (AOR).

   If this stage fails, the device proceeds to classification based on Proxy Set.
2. **Classification Stage 2 - Proxy Set:** If the database search fails, the device performs classification based on Proxy Set. This is done only if the 'Classify By Proxy Set' parameter is enabled for the IP Group (see "Configuring IP Groups" on page 174). The device checks whether the INVITE's IP address (if host name, then according to the dynamically resolved IP address list) is defined for a Proxy Set ID (in the Proxy Set table). If a Proxy Set ID has such an IP address, the device classifies the INVITE to the IP Group that is associated with this Proxy Set. (The Proxy Set ID is assigned to the IP Group using the IP Group table's 'Proxy Set ID' parameter.)

   Note: For security, it is highly recommended to disable the Classify by Proxy Set feature so that the device can use the Classification table instead. This enables "strict" classification of incoming calls to IP Groups. In addition, in cases where multiple IP Groups are associated with the same Proxy Set ID, do **not** use the Classify by Proxy Set feature.

   If this stage fails (or Classify by Proxy Set is disabled), the device proceeds to classification based on the Classification table.
3. **Classification Stage 3 - Classification Table:** If classification based on Proxy Set fails (or disabled), the device uses the Classification table to classify the SIP dialog to an IP Group. If it locates a Classification rule whose characteristics (such as source IP address) match the incoming SIP dialog, the SIP dialog is assigned to the associated IP Group. In addition, if the Classification rule is defined as a whitelist, the SIP dialog is allowed and proceeds with the manipulation, routing and other SBC processes. If the Classification rule is defined as a blacklist, the SIP dialog is denied.

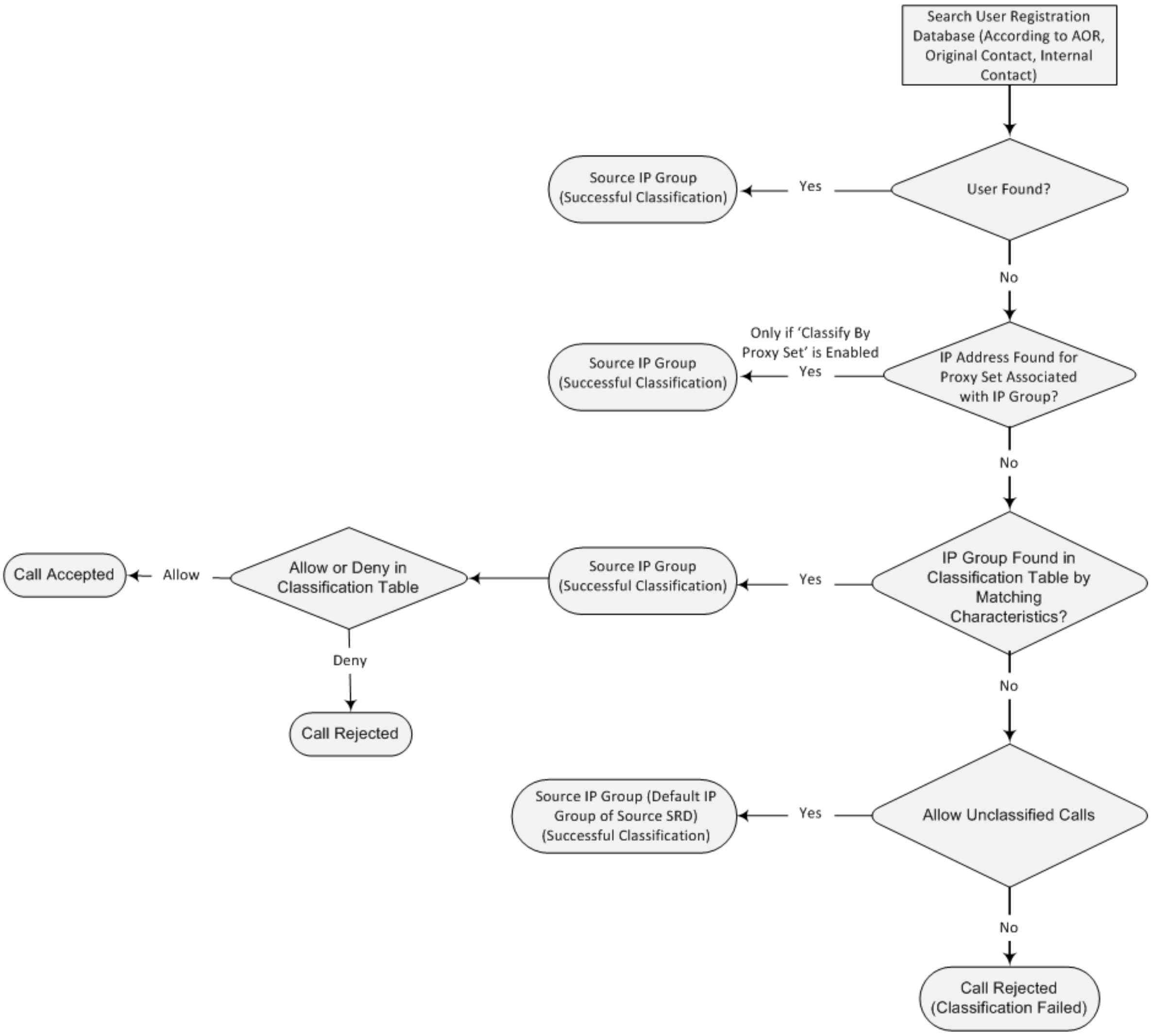
If the classification process fails, the device rejects or allows the call, depending on the setting of the 'Unclassified Calls' parameter (on the General Settings page - **Configuration** tab > **VoIP** menu > **SBC** > **General Settings**). If this parameter is set to **Allow**, the incoming SIP dialog is assigned to an IP Group as follows:

1. The device checks on which SIP listening port (e.g., 5061) the incoming SIP dialog request arrived and the SIP Interface which is configured with this port (in the SIP Interface table).
2. The device checks the SRD that is associated with this SIP Interface (in the SIP Interface table) and then classifies the SIP dialog with the first IP Group that is associated with this SRD. For example, if IP Groups 3 and 4 use the same SRD, the device classifies the call to IP Group 3.

Note: If classification for a SIP request fails and the device is configured to reject unclassified calls, the device can send a specific SIP response code per SIP interface. This is configured by the 'Classification Failure Response Type' parameter in the SIP Interface table (see "Configuring SIP Interfaces" on page 171).

The flowchart below illustrates the classification process:

Figure 101: Classification Process (Identifying IP Group or Rejecting Call)


Search User Registration Database (According to AOR, Original Contact, Internal Contact)
User Found?
Yes
Source IP Group (Successful Classification)
No
IP Address Found for Proxy Set Associated with IP Group?
Only if 'Classify By Proxy Set' is Enabled
Yes
Source IP Group (Successful Classification)
No
IP Group Found in Classification Table by Matching Characteristics?
Yes
Source IP Group (Successful Classification)
Allow or Deny in Classification Table
Allow
Call Accepted
Deny
Call Rejected
No
Allow Unclassified Calls
Yes
Source IP Group (Default IP Group of Source SRD) (Successful Classification)
No
Call Rejected (Classification Failed)

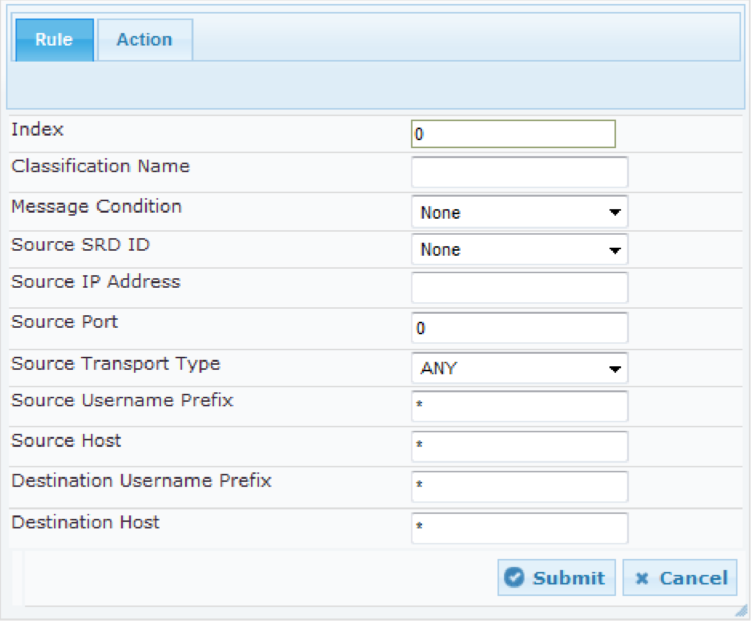
Note: The device saves incoming SIP REGISTER messages in its registration database. If the REGISTER message is received from a User-type IP Group, the device sends the message to the configured destination.

The procedure below describes how to configure Classification rules in the Web interface. You can also configure Classification rules using the table ini file parameter, Classification or CLI command, configure voip > sbc routing classification.

- ➢ To configure a Classification rule:

1. Open the Classification Table page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Classification Table**).
2. Click **Add**; the following dialog box appears:

Figure 102: Classification Table Page



Index	0
Classification Name	
Message Condition	None
Source SRD ID	None
Source IP Address	
Source Port	0
Source Transport Type	ANY
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*

3. Configure the Classification rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 71: Classification Table Parameter Descriptions

Parameter	Description
Index **[Classification_Index]**	Defines an index number for the new table record.
Classification Name CLI: classification-name **[Classification_ClassificationName]**	Defines an arbitrary name to easily identify the Classification rule. The valid value is a string of up to 20 characters. By default, no name is defined.
Matching Characteristics - Rule	
Message Condition CLI: message-condition **[Classification_MessageCondition]**	Assigns a Message Condition rule, which can be used to classify the incoming SIP dialog. To configure Condition rules, see "Configuring Message Condition Rules" on page 254.
Source SRD ID CLI: src-srd-id **[Classification_SrcSRDID]**	Defines an SRD ID of the incoming SIP dialog. To configure SRDs, see "Configuring SRDs" on page 169. By default, no SRD is defined. **Note:** The SRDs are also associated with a port number as defined by the SIP Interface used by the SRD (see "Configuring SIP Interfaces" on page 171).
Source IP Address CLI: src-ip-address **[Classification_SrcAddress]**	Defines the source IP address (in dotted-decimal notation) of the incoming SIP dialog. The IP address can be configured using the following wildcards:

Parameter	Description
	▪ "x" wildcard: represents single digits. For example, 10.8.8.xx represents all the addresses between 10.8.8.10 to 10.8.8.99. ▪ asterisk (*) wildcard: represents any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255. If this parameter is not configured or is configured as an asterisk (*), any source IP address is accepted.
Source Port CLI: src-port **[Classification_SrcPort]**	Defines the source port number of the incoming SIP dialog.
Source Transport Type CLI: src-transport-type **[Classification_SrcTransportType]**	Defines the source transport type (UDP, TCP, or TLS) of the incoming SIP dialog. ▪ **[-1]** ANY (Default) = All transport types ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS
Source Username Prefix CLI: src-user-name-prefix **[Classification_SrcUsernamePrefix]**	Defines the prefix of the source URI user part of the incoming SIP dialog. The URI is typically located in the SIP From header. However, you can configure the SIP header from where the device obtains the source URI. This is done in the IP Group table, using the 'Source URI Input' parameter. For more information on how the device obtains this URI, see "SIP Dialog Initiation Process" on page 215. The default is the asterisk (*) symbol, which represents any source username prefix. The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379. **Note:** For REGISTER requests, the source URL is obtained from the To header.
Source Host CLI: src-host **[Classification_SrcHost]**	Defines the prefix of the source URI host name. The URI is typically located in the SIP From header. However, you can configure the SIP header from where the device obtains the source URI. This is done in the IP Group table, using the 'Source URI Input' parameter. For more information on how the device obtains this URI, see "SIP Dialog Initiation Process" on page 215. The default is the asterisk (*) symbol, which represents any source host prefix. **Note:** For REGISTER requests, the source URL is obtained from the To header.
Destination Username Prefix CLI: dst-user-name-prefix **[Classification_DestUsernamePrefix]**	Defines the prefix of the destination Request-URI user part of the incoming SIP dialog. The default is the asterisk (*) symbol, which represents any destination username. The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379.
Destination Host CLI: dst-host **[Classification_DestHost]**	Defines the prefix of the destination Request-URI host name of the incoming SIP dialog request. The default is the asterisk (*) symbol, which represents any destination host prefix.
Operation Rule - Action	

Parameter	Description
Action Type CLI: action-type **[Classification_ActionType]**	Defines a whitelist or blacklist for incoming SIP dialog requests that match the characteristics of the classification rule. ▪ **[0]** Deny = Blocks incoming SIP dialogs that match the characteristics of the Classification rule (blacklist). ▪ **[1]** Allow = (Default) Allows incoming SIP dialogs that match the characteristics of the Classification rule (whitelist) and assigns it to the associated IP Group. (default)
Source IP Group ID CLI: src-ip-group-id **[Classification_SrcIPGroupID]**	Defines an IP Group to which the incoming SIP dialog request must be assigned if this SIP dialog matches the matching characteristics. The IP Group is used for SBC routing and manipulations. To configure IP Groups, see "Configuring IP Groups" on page 174. By default, no IP Group is defined. **Note:** The IP Group must be associated with the assigned SRD.

22.6.1.1 *Classification Based on URI of Selected Header Example*

The following example describes how to configure classification of incoming calls to IP Groups, based on source URI in a specific SIP header.

This example assumes the following incoming INVITE message:

```
INVITE sip:8000@10.33.4.226 SIP/2.0
Via: SIP/2.0/UDP 10.33.4.226;branch=z9hG4bKVEBTDAHSUYRTEXEDEGJY
From: <sip:100@10.33.4.226>;tag=YSQQKXXREVDPYPTNFMWG
To: <sip:8000@10.33.4.226>
Call-ID: FKPNOYRNKROIMEGBSSKS@10.33.4.226
CSeq: 1 INVITE
Contact: <sip:100@10.33.4.226>
Route: <sip:2000@10.10.10.10.10>,<sip:300@10.10.10.30>
Supported: em,100rel,timer,replaces
P-Called-Party-ID: <sip:1111@10.33.38.1>
User-Agent: Sip Message Generator V1.0.0.5
Content-Length: 0
```

1. In the Classification table, add the following classification rules:

Index	Source Username Prefix	Destination Username Prefix	Destination Host	Source IP Group ID
0	333	-	-	1
1	1111	2000	10.10.10.10	2

1. In the IP Group table, add the following IP Groups:

Index	Source URI Input	Destination URI Input
1	-	-
2	P-Called-Party-ID	Route

In this example, a match exists only for Classification Rule #1. This is because the source (1111) and destination (2000) username prefixes match those in the INVITE's P-Called-Party-ID header (i..e., "<sip:1111@10.33.38.1>") and Route header (i.e., "<sip:2000@10.10.10.10.10>"), respectively. These SIP headers were determined in IP Group ID 2.

22.6.2 Configuring Message Condition Rules

The Message Condition table lets you configure up to 40 Message Condition rules. A Message Condition defines special conditions (requisites) for incoming SIP messages. These rules can be used as additional matching criteria for the following:

- Classification rules in the Classification table (see "Configuring Classification Rules" on page 249)
- IP-to-IP routing rules in the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 255)
- IP-to-IP outbound manipulation rules in the IP to IP Outbound Manipulation table (see "Configuring IP-to-IP Outbound Manipulations" on page 268)

Message Condition rules are configured using the same syntax as that used for Conditions when configuring Message Manipulation rules in the Message Manipulations table (see "Configuring SIP Message Manipulation" on page 193). You can configure simple Message Condition rules, for example, "header.to.host contains company", meaning SIP messages whose To header has a host part containing the string "company". You can configure complex rules using the "AND" or "OR" Boolean operands and also use regular expressions (regex), for example:

- "body.sdp regex pcmu" can be used to enable routing based on the offered codec (G.711 Mu) in the incoming SDP message.
- "body.sdp regex (AVP[0-9||\s]*\s8[\s||\n])" can be used to enable routing based on payload type 8 in the incoming SDP message.

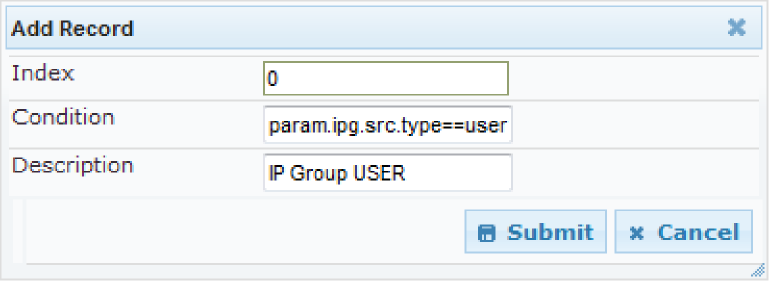
Note: For a description on SIP message manipulation syntax, refer to the *SIP Message Manipulations Quick Reference Guide.*

The procedure below describes how to configure Message Condition rules in the Web interface. You can also configure Message Condition rules using the table ini file parameter, ConditionTable or CLI command, configure voip > sbc routing condition-table.

➢ To configure a Message Condition rule:

1. Open the Message Condition Table page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Message Condition Table**).
2. Click **Add**; the following dialog box appears:

Figure 103: Condition Table Page - Add Record Dialog Box



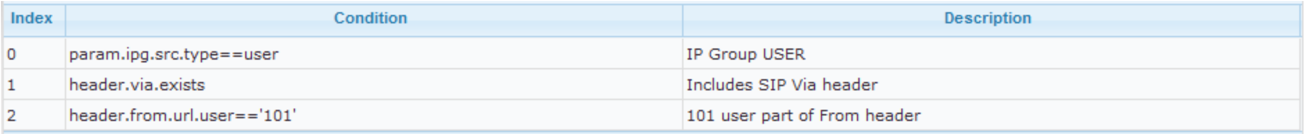
Add Record

Index	0
Condition	param.ipg.src.type==user
Description	IP Group USER

3. Configure a Message Condition rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

An example of configured Message Condition rules is shown in the figure below:

Figure 104: Condition Table Page



Index	Condition	Description
0	param.ipg.src.type==user	IP Group USER
1	header.via.exists	Includes SIP Via header
2	header.from.url.user=='101'	101 user part of From header

- **Index 0:** Incoming SIP dialog that is classified as belonging to a User-type IP Group.
- **Index 1:** Incoming SIP dialog that contains a SIP Via header.
- **Index 2:** Incoming SIP dialog with 101 as the user part in the SIP From header.

Table 72: Message Condition Table Parameter Descriptions

Parameter	Description
Index **[ConditionTable_Index]**	Defines an index number for the new table record.
Condition CLI: condition **[ConditionTable_Condition]**	Defines the Condition rule of the SIP message. The valid value is a string. **Note:** User and host parts must be enclosed in single quotes.
Description CLI: description **[ConditionTable_Description]**	Defines a brief description of the Condition rule.

22.6.3 Configuring SBC IP-to-IP Routing

The IP-to-IP Routing table lets you configure up to 1,000 SBC IP-to-IP routing rules. An IP-to-IP routing rule routes received SIP dialog messages (e.g., INVITE) to an IP destination. An IP-to-IP routing rule includes two areas:

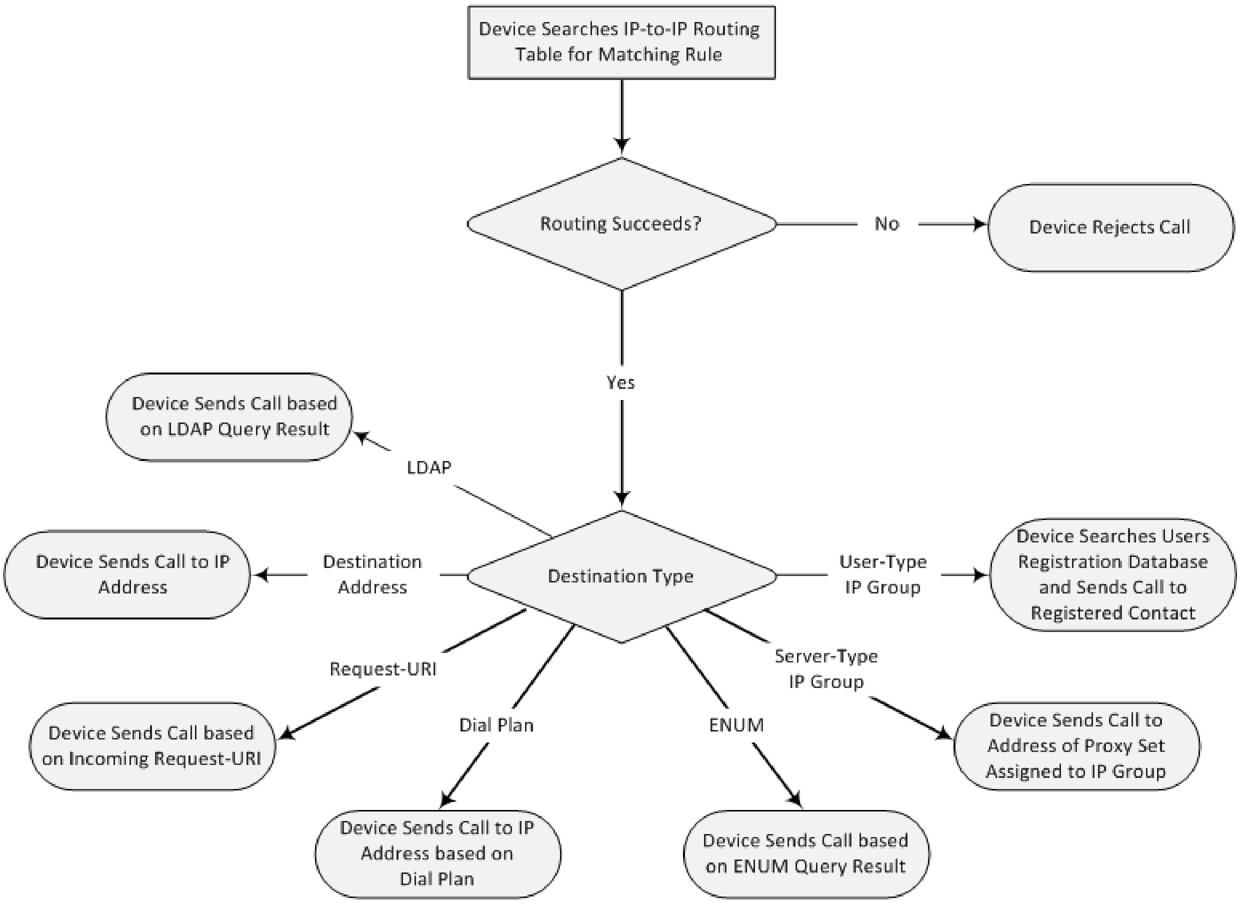
- Matching characteristics (Rule) - characteristics of incoming SIP message such as the IP Group from where the message was received.
- Operation (Action) - if the incoming SIP message matches the characteristics of the rule, the device routes the message to the configured destination

If the characteristics of an incoming call do not match the first rule in the table, the call characteristics are compared to those of the second rule, and so on until a matching rule is located. If no rule is matched, the call is rejected.

You can configure the IP-to-IP routing rule to send the call to any of the following IP destinations:

- According to registered user Contact listed in the device's database (only for User-type IP Groups).
- IP Group - the destination is the address configured for the Proxy Set associated with the IP Group (allows redundancy/load balancing).
- IP address in dotted-decimal notation or FQDN. Routing to a host name can be resolved using NAPTR/SRV/A-Record.
- Request-URI of incoming SIP dialog initiating requests.
- According to result of an ENUM query.
- Hunt Group - used for call survivability of call centers (see "Call Survivability for Call Centers" on page 236).
- IP address according to a specified Dial Plan index listed in the loaded Dial Plan file.
- According to result of LDAP query (for more information on LDAP-based routing, see "Routing Based on LDAP Active Directory Queries" on page 137).

Figure 105: IP-to-IP Routing Destination Types


Device Searches IP-to-IP Routing Table for Matching Rule
Routing Succeeds?
No
Device Rejects Call
Yes
Device Sends Call based on LDAP Query Result
LDAP
Device Sends Call to IP Address
Destination Address
Destination Type
User-Type IP Group
Device Searches Users Registration Database and Sends Call to Registered Contact
Request-URI
Server-Type IP Group
Device Sends Call based on Incoming Request-URI
Dial Plan
ENUM
Device Sends Call to Address of Proxy Set Assigned to IP Group
Device Sends Call to IP Address based on Dial Plan
Device Sends Call based on ENUM Query Result

The IP-to-IP Routing table also provides the following features:

- **Alternative routing or load balancing:** In addition to the alternative routing/load balancing provided by the Proxy Set associated with the destination IP Group, the table allows the configuration of alternative routes whereby if a route fails, the next adjacent (below) rule in the table that is configured as 'Alt Route Ignore/Consider Inputs' are used. The alternative routes rules can be set to enforce the input matching criteria or to ignore any matching criteria. Alternative routing occurs upon one of the following conditions:
    - A request sent by the device is responded with one of the following:

  - ♦ SIP response code (i.e., 4xx, 5xx, and 6xx SIP responses) configured in the SBC Alternative Routing Reasons table (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 262).
  - ♦ SIP 408 Timeout or no response (after timeout).
- The DNS resolution includes IP addresses that the device has yet to try (for the current call).

Messages are re-routed with the same SIP Call-ID and CSeq header fields (increased by 1).

- **Re-routing of SIP requests:** This table enables you to configure "re-routing" rules of requests (e.g., INVITEs) that the device sends upon receipt of SIP 3xx responses or REFER messages. These rules are configured for destinations that do not support receipt of 3xx or REFER and where the device handles the requests locally (instead of forwarding the 3xx or REFER to the destination).
- **Least cost routing (LCR):** If the LCR feature is enabled, the device searches the routing table for matching routing rules and then selects the one with the lowest call cost. The call cost of the routing rule is done by assigning it a Cost Group. For configuring Cost Groups, see "Least Cost Routing" on page 142. If two routing rules have identical costs, then the rule appearing higher up in the table (i.e., first-matched rule) is used. If a selected route is unavailable, the device uses the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched routing rules with Cost Groups, according to the settings of the LCR parameter, LCRDefaultCost (see "Enabling LCR and Configuring Default LCR" on page 144).
- **Call Forking:** The IP-to-IP Routing table can be configured to route an incoming IP call to multiple destinations (call forking). The incoming call can be routed to multiple destinations of any type such as an IP Group or IP address. The device forks the call by sending simultaneous INVITE messages to all the specified destinations. It handles the multiple SIP dialogs until one of the calls is answered and then terminates the other SIP dialogs.

  Call forking is configured by creating a Forking group. A Forking group consists of a main routing rule ('Alternative Route Options' set to **Route Row**) whose 'Group Policy' is set to **Forking**, and one or more associated routing rules ('Alternative Route Options' set to **Group Member Ignore Inputs** or **Group Member Consider Inputs**). The group members must be configured in contiguous table rows to the main routing rule. If an incoming call matches the input characteristics of the main routing rule, the device routes the call to its destination and all those of the group members.

  An alternative routing rule can also be configured for the Forking group. The alternative route is used if the call fails for the Forking group (i.e., main route and all its group members). The alternative routing rule must be configured in the table row immediately below the last member of the Forking group. The 'Alternative Route Options' of this alternative route must be set to **Alt Route Ignore Inputs** or **Alt Route Consider Inputs**. The alternative route can also be configured with its own forking group members, where if the device uses the alternative route, the call is also sent to its group members. In this case, instead of setting the alternative route's 'Group Policy' to **None**, you must set it to **Forking**. The group members of the alternative route must be configured in the rows immediately below it.

  The LCR feature can also be employed with call forking. The device calculates a maximum call cost for each Forking group and routes the call to the Forking group with the lowest cost. Thus, even if the call can successfully be routed to the main routing rule, a different routing rule can be chosen (even an alternative route, if configured) based on LCR. If routing to one Forking group fails, the device tries to route the call to the Forking group with the next lowest cost (main or alternative route), and so on. The prerequisite for this functionality is that the incoming call must successfully match the input characteristics of the main routing rule.
- **Dial Plan Prefix Tags for Representing Source / Destination Numbers:** If your deployment includes calls of many different called (source URI user name) and/or calling (destination URI user name) numbers that need to be routed to the same destination, you can employ user-defined prefix tags to represent these numbers. Thus, instead of configuring many routing rules, you need to configure only one routing rule using the prefix tag as the source and destination number matching characteristics, and a destination for the calls. For more information on prefix tags, see "Dial Plan Prefix Tags for

SBC IP-to-IP Routing" on page 303.

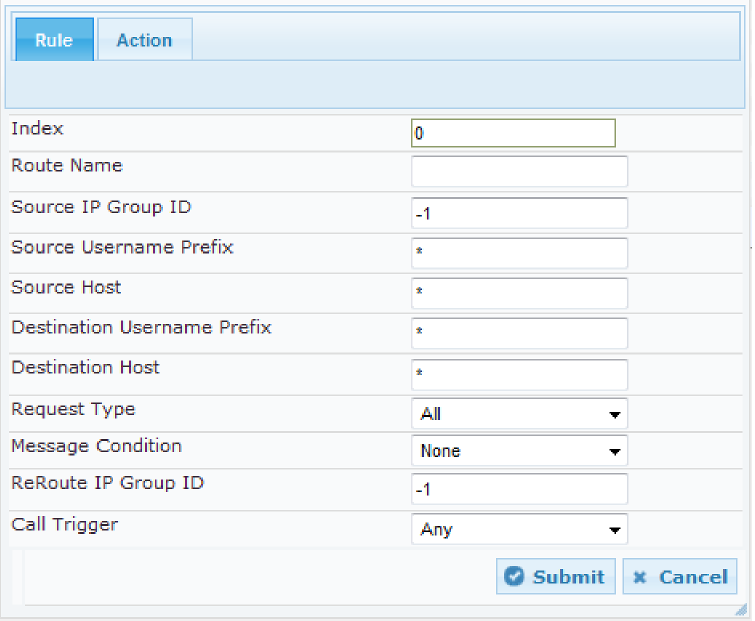
Note: Call forking is not applicable to LDAP-based IP-to-IP routing rules.

The procedure below describes how to configure IP-to-IP routing rules in the Web interface. You can also configure IP-to-IP routing rules using the table ini file parameter, IP2IPRouting or CLI command, configure voip > sbc routing ip2ip-routing.

➢ To configure an IP-to-IP routing rule:

1. Open the IP-to-IP Routing Table page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **IP to IP Routing Table**).
2. Click **Add**; the following dialog box appears:

Figure 106: IP-to-IP Routing Table - Add Record Dialog Box



Index	0
Route Name	
Source IP Group ID	-1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Request Type	All
Message Condition	None
ReRoute IP Group ID	-1
Call Trigger	Any

3. Configure an IP-to-IP routing rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 73: IP-to-IP Routing Table Parameter Descriptions

Parameter	Description
Index **[IP2IPRouting_Index]**	Defines an index number for the new table record.
Route Name CLI: route-name **[IP2IPRouting_RouteName]**	Defines an arbitrary name to easily identify the IP-to-IP routing rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Characteristics - Rule	
Source IP Group ID [IP2IPRouting_SrcIPGroupID] CLI: src-ip-group-id	Defines the IP Group from where the IP call was received. Typically, the IP Group of an incoming SIP dialog is determined (or classified) using the Classification table (see Configuring Classification Rules on page 249). The default is -1. To denote any IP Group, leave this field empty.

Parameter	Description
Source Username Prefix **[IP2IPRouting_SrcUsernamePrefix]** CLI: src-user-name-prefix	Defines the prefix of the user part of the incoming SIP dialog's source URI (usually the From URI). You can use special notations for denoting the prefix. To denote calls without a user part in the URI, use the $ sign. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379. The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty.
Source Host **[IP2IPRouting_SrcHost]** CLI: src-host	Defines the host part of the incoming SIP dialog's source URI (usually the From URI). The default is the asterisk (*) symbol (i.e., any host name). If this rule is not required, leave this field empty.
Destination Username Prefix **[IP2IPRouting_DestUsernamePrefix]** CLI: dst-user-name-prefix	Defines the prefix of the incoming SIP dialog's destination URI (usually the Request URI) user part. You can use special notations for denoting the prefix. To denote calls without a user part in the URI, use the $ sign. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379. The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty.
Destination Host **[IP2IPRouting_DestHost]** CLI: dst-host	Defines the host part of the incoming SIP dialog's destination URI (usually the Request-URI). The default is the asterisk (*) symbol (i.e., any destination host). If this rule is not required, leave this field empty.
Request Type [IP2IPRouting_RequestType] CLI: request-type	Defines the SIP dialog request type of the incoming SIP dialog. ▪ [0] All (default) ▪ [1] INVITE ▪ [2] REGISTER ▪ [3] SUBSCRIBE ▪ [4] INVITE and REGISTER ▪ [5] INVITE and SUBSCRIBE ▪ [6] OPTIONS
Message Condition **[IP2IPRouting_MessageCondition]** CLI: message-condition	Assigns a SIP message Condition rule. To configure Condition rules, see "Configuring Message Condition Rules" on page 254.
ReRoute IP Group ID **[IP2IPRouting_ReRouteIPGroupID]** CLI: re-route-ip-group-id	Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. This field is typically used for re-routing requests (e.g., INVITEs) when interworking is required for SIP 3xx redirect responses or REFER messages. For more information, see "Interworking SIP 3xx Redirect Responses" on page 227 and "Interworking SIP REFER Messages" on page 229, respectively. This parameter functions together with the 'Call Trigger' field (see below). The default is -1 (i.e., not configured).
Call Trigger **[IP2IPRouting_Trigger]** CLI: trigger	Defines the reason (i.e, trigger) for re-routing the SIP request: ▪ **[0]** Any = (Default) This routing rule is used for all scenarios (re-routes and non-re-routes). ▪ **[1]** 3xx = Re-routes the request if it was triggered as a result of a SIP 3xx response. ▪ **[2]** REFER = Re-routes the INVITE if it was triggered as a result of a REFER request. ▪ **[3]** 3xx or REFER = Applies to options [1] and [2]. ▪ **[4]** Initial only = This routing rule is used for regular requests that the device

Parameter	Description
	forwards to the destination. This rule is not used for re-routing of requests triggered by the receipt of REFER or 3xx.
Call Setup Rules Set Id CLI: call-setup-rules-set-id **[IP2IPRouting_CallSetupRulesSetId]**	Assigns a Call Setup Rule Set ID to the routing rule. The device performs the Call Setup rules of this Set ID if the incoming call matches the characteristics of this routing rule. The device routes the call to the destination according to the routing rule's configured action, only after it has performed the Call Setup rules. For configuring Call Setup rules, see "Configuring Call Setup Rules" on page 148.
Operation Routing Rule - Action	

Destination Type
[IP2IPRouting_DestType]
CLI: dst-type
	Determines the destination type to which the outgoing SIP dialog is sent.
	▪ **[0]** IP Group = (Default) The SIP dialog is sent to the IP Group's Proxy Set (SERVER-type IP Group) or registered contact from the database (if USER-type IP Group).
	▪ **[1]** Dest Address = The SIP dialog is sent to the address configured in the following fields: 'Destination SRD ID', 'Destination Address', 'Destination Port', and 'Destination Transport Type'.
	▪ **[2]** Request URI = The SIP dialog is sent to the address indicated in the incoming Request-URI. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence.
	▪ **[3]** ENUM = An ENUM query is sent to include the destination address. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence.
	▪ **[4]** Hunt Group = Used for call center survivability. For more information, see "Call Survivability for Call Centers" on page 236.
	▪ **[5]** Dial Plan = The IP destination is determined by a Dial Plan index of the loaded Dial Plan file. The syntax of the Dial Plan index in the Dial Plan file is as follows: <destination / called prefix number>,0,<IP destination>
	▪ Note that the second parameter "0" is ignored. An example of a configured Dial Plan (# 6) in the Dial Plan file is shown below:
	`[ PLAN6 ]`
	`200,0,10.33.8.52     ; called prefix 200 is routed to destination 10.33.8.52`
	`201,0,10.33.8.52`
	`300,0,itsp.com       ; called prefix 300 is routed to destination itsp.com`
	▪ Once the Dial Plan is defined, you need to assign it (0 to 7) to the routing rule as the destination in the 'Destination Address' parameter, where "0" denotes [PLAN1], "1" denotes [PLAN2], and so on.
	▪ [7] LDAP = LDAP-based routing.

Destination IP Group ID **[IP2IPRouting_DestIPGroupID]** CLI: dst-ip-group-id	Defines the IP Group ID to where you want to route the call. The SIP dialog messages are sent to the IP address defined for the Proxy Set associated with this IP Group. If you select an IP Group, it is unnecessary to configure a destination IP address (in the 'Destination Address' field). However, if both parameters are configured, then the IP Group takes precedence. If the destination IP Group is of USER type, the device searches for a match between the Request-URI (of the received SIP dialog) to an AOR registration record in the device's database. The SIP dialog is then sent to the IP address of the registered contact. The default is -1. **Notes:** ▪ This parameter is only relevant if the parameter 'Destination Type' is set to **IP Group**. However, regardless of the settings of the parameter 'Destination

Parameter	Description
	Type', the IP Group is still used - only for determining the IP Profile or outgoing SRD. If neither IP Group nor SRD are defined in this table, the destination SRD is determined according to the source SRD associated with the Source IP Group (configured in the IP Group table, see "Configuring IP Groups" on page 174). If this table does not define an IP Group but only an SRD, the first IP Group associated with this SRD (in the IP Group table) is used. ▪ If the destination IP Group ID is of SERVER type, the request is routed according to the IP Group addresses. ▪ If the destination IP Group ID is of USER type, the request is routed according to the IP Group specific database (i.e., only to registered users of the selected database). ▪ If the destination IP Group ID is ANY USER ([-2]), the request is routed according to the general database (i.e., any matching registered user).
Destination SRD ID [IP2IPRouting_DestSRDID] CLI: dst-srd-id	Defines the SRD ID. The default is None. **Note:** The destination IP Group must belong to the destination SRD if both are configured in this table.
Destination Address **[IP2IPRouting_DestAddress]** CLI: dst-address	Defines the destination to where the call is sent. This can be an IP address or a domain name (e.g., domain.com). If ENUM-based routing is used (i.e., the 'Destination Type' parameter is set to **ENUM**) this parameter defines the IP address or domain name (FQDN) of the ENUM service, for example, e164.arpa, e164.customer.net or NRENum.net. The device sends the ENUM query containing the destination phone number to an external DNS server, configured in the Interface table. The ENUM reply includes a SIP URI (user@host) which is used as the destination Request-URI in this routing table. The valid value is a string of up to 50 characters. **Notes:** ▪ This parameter is applicable only if the 'Destination Type' parameter is set to **Dest Address** [1] or **ENUM** [3]. ▪ When using domain names, enter a DNS server IP address or alternatively, define these names in the Internal DNS table (see "Configuring the Internal SRV Table" on page 106). ▪ To terminate SIP OPTIONS messages at the device (i.e., to handle them locally), set this parameter to "internal".
Destination Port **[IP2IPRouting_DestPort]** CLI: dst-port	Defines the destination port to where the call is sent.
Destination Transport Type **[IP2IPRouting_DestTransportType]** CLI: dst-transport-type	Defines the transport layer type for sending the call: ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** If this parameter is not configured, the transport type is determined by the SIPTransportType parameter.
Alternative Route Options [IP2IPRouting_AltRouteOptions] CLI: alt-route-options	Determines whether this routing rule is the main routing rule or an alternative routing rule (to the rule defined directly above it in the table). ▪ [0] Route Row (default) = Main routing rule - the device first attempts to route the call to this route if the incoming SIP dialog's input characteristics matches this rule. ▪ [1] Alt Route Ignore Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route regardless of the incoming SIP dialog's input characteristics. ▪ [2] Alt Route Consider Inputs = If the call cannot be routed to the main route

Parameter	Description
	(Route Row), the call is routed to this alternative route only if the incoming SIP dialog matches this routing rule's input characteristics. ▪ [3] Group Member Ignore Inputs = This routing rule is a member of the Forking routing rule. The incoming call is also forked to the destination of this routing rule. The matching input characteristics of the routing rule are ignored. ▪ [4] Group Member Consider Inputs = This routing rule is a member of the Forking routing rule. The incoming call is also forked to the destination of this routing rule only if the incoming call matches this rule's input characteristics. **Notes:** ▪ The alternative routing entry ([1] or [2]) must be defined in the next consecutive table entry index to the Route Row entry (i.e., directly below it). For example, if Index 4 is configured as a Route Row, Index 5 must be configured as the alternative route. ▪ The Forking Group members must be configured in a table row that is immediately below the main Forking routing rule, or below an alternative routing rule for the main rule, if configured. ▪ For IP-to-IP alternative routing, configure alternative routing reasons upon receipt of 4xx, 5xx, and 6xx SIP responses (see Configuring SIP Response Codes for Alternative Routing Reasons on page 262). However, if no response, ICMP, or a SIP 408 response is received, the device attempts to use the alternative route even if no entries are configured in the 'SBC Alternative Routing Reasons' table. ▪ Multiple alternative route entries can be configured (e.g., Index 1 is the main route - Route Row - and indices 2 through 4 are configured as alternative routes).
Group Policy CLI: group-policy **[IP2IPRouting_GroupPolicy]**	Defines whether the routing rule includes call forking. ▪ **[0]** None (default) = Call uses only this route (even if Forking Group members are configured in the rows below it). ▪ **[1]** Forking = Call uses this route and the routes of Forking Group members, if configured (in the rows below it).
Cost Group **[IP2IPRouting_CostGroup]** CLI: cost-group	Assigns a Cost Group to the routing rule for determining the cost of the call. To configure Cost Groups, see "Configuring Cost Groups" on page 146. By default, no Cost Group is defined.

22.6.4 Configuring SIP Response Codes for Alternative Routing Reasons

The SBC Alternative Routing Reasons table lets you configure up to 20 SIP response codes for call release (termination) reasons. If a call (outgoing SIP dialog-initiating methods, e.g., INVITE, OPTIONS, and SUBSCRIBE messages) is released as a result of a configured SIP code (provided in SIP 4xx, 5xx, and 6xx), the device attempts to locate an alternative route for the call in the IP-to-IP Routing table. Alternative routing rules are configured with the 'Alternative Route Options' parameter set to **Alt Route Ignore Inputs** or **Alt Route Consider Inputs** (see "Configuring SBC IP-to-IP Routing Rules" on page 255).

Typically, the device performs alternative routing when there is no response at all to an INVITE message. This is done after a user-defined number of INVITE re-transmissions, configured by the SIPMaxRtx parameter. In such a scenario, the device issues itself the SIP response code 408 (Request Timeout). Alternative routing is only done if you have configured this response code in the SBC Alternative Routing Reasons table.

You can also configure alternative routing for the following proprietary response codes that are issued by the device itself:

- **805 IP Profile Call Limit:** The device generates this response code when Call Admission Control (CAC) limits (such as maximum concurrent calls) are exceeded for an IP Group (or SRD). The CAC rules are configured in the Admission Control table (see "Configuring Admission Control" on page 244). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity. In such a scenario, an alternative route configured in the IP-to-IP Routing table can be used.
- **806 Media Limits Exceeded:** The device generates this response code when the call is terminated due to crossed thresholds of QoE metrics such as MOS, packet delay, and packet loss (configured in the Quality of Experience Profile table) and/or media bandwidth (configured in the Bandwidth profile table). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity. This is configured by 1) assigning an IP Group a QoE and/or Bandwidth profile that rejects calls if the threshold is crossed, 2) configuring 806 in the SBC Alternative Routing Reasons table and 3) configuring an alternative routing rule.

Notes:

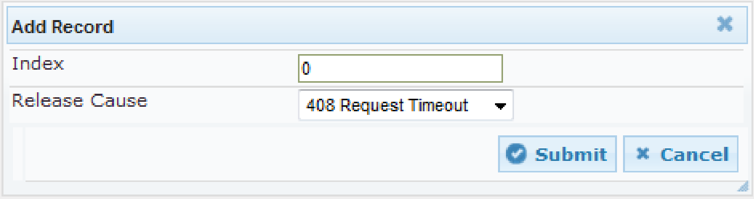
- If the device receives a SIP 408 response, an ICMP message, or no response, alternative routing is still performed even if the SBC Alternative Routing Reasons table is not configured.
- SIP requests belonging to an SRD or IP Group that have reached the call limit (maximum concurrent calls and/or call rate) as configured in the Call Admission table are sent to an alternative route if configured in the IP-to-IP Routing table for the SRD or IP Group. If no alternative routing rule is located, the device automatically rejects the SIP request with a SIP 486 (Busy Here) response.

The procedure below describes how to configure the SBC Alternative Routing Reasons table in the Web interface. You can also configure this table using the table ini file parameter, SBCAlternativeRoutingReasons or CLI command, configure voip > sbc routing sbc-alt-routing-reasons.

➢ To configure SIP reason codes for alternative IP routing:

1. Open the SBC Alternative Routing Reasons page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Alternative Routing Reasons**).
2. Click **Add**; the following dialog box appears:

Figure 107: Alternative Routing Reasons Table - Add Record



Add Record

Index	0
Release Cause	408 Request Timeout

3. Configure a SIP response code for alternative routing according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 74: SBC Alternative Routing Reasons Table Parameter Descriptions

Parameter	Description
Index **[SBCAlternativeRoutingReasons_Index]**	Defines an index number for the new table record.
Release Cause CLI: rel-cause **[SBCAlternativeRoutingReasons_ReleaseCause]**	Defines a SIP response code for triggering the device's alternative routing mechanism.

22.7 SBC Manipulations

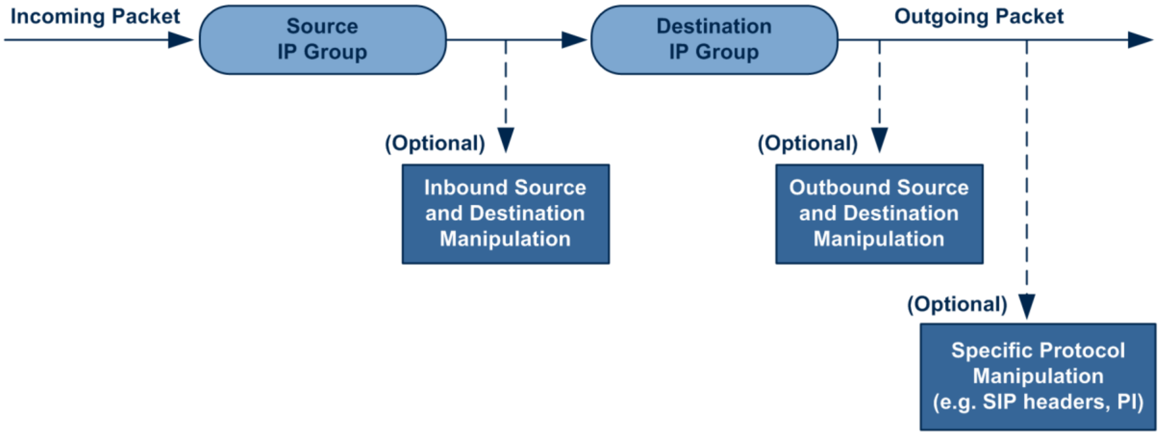
This section describes the configuration of the manipulation rules for the SBC application.

Note: For additional manipulation features, see the following:

- "Configuring SIP Message Policy Rules".
- "Configuring SIP Message Manipulation" on page 193.

The device supports SIP URI user part (source and destination) manipulations for inbound and outbound routing. These manipulations can be applied to a source IP group, source and destination host and user prefixes, and/or user-defined SIP request (e.g., INVITE, OPTIONS, SUBSCRIBE, and/or REGISTER). Since outbound manipulations are performed after routing, the outbound manipulation rule matching can also be done by destination IP Group. Manipulated destination user and host are performed on the following SIP headers: Request-URI, To, and Remote-Party-ID (if exists). Manipulated source user and host are performed on the following SIP headers: From, P-Asserted (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists).

Figure 108: SIP URI Manipulation in IP-to-IP Routing


Incoming Packet
Source IP Group
Destination IP Group
Outgoing Packet
(Optional)
(Optional)
Inbound Source and Destination Manipulation
Outbound Source and Destination Manipulation
(Optional)
Specific Protocol Manipulation (e.g. SIP headers, PI)

You can also restrict source user identity in outgoing SIP dialogs in the Outbound Manipulation table (using the column PrivacyRestrictionMode). The device identifies an incoming user as restricted if one of the following exists:

- From header user is 'anonymous'.
- P-Asserted-Identity and Privacy headers contain the value 'id'.

All restriction logic is done after the user number has been manipulated.

Host name (source and destination) manipulations are simply host name substitutions with the names defined for the source and destination IP Groups respectively (if any, in the IP Group table).

Below is an example of a call flow and consequent SIP URI manipulations:

- **Incoming INVITE from LAN:**

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
Via: SIP/2.0/UDP 10.2.2.6;branch=z9hGLLLLLan
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OlLAN;paramer1=abe
To: <sip:1000@10.2.2.3;user=phone>
Call-ID: USELLLAN@10.2.2.3
CSeq: 1 INVITE
Contact: <sip:7000@10.2.2.3>
Supported: em,100rel,timer,replaces
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK
User-Agent: Sip Message Generator V1.0.0.5
Content-Type: application/sdp
Content-Length: 155
```

```
v=0
o=SMG 791285 795617 IN IP4 10.2.2.6
s=Phone-Call
c=IN IP4 10.2.2.6
t=0 0
m=audio 6000 RTP/AVP 8
a=rtpmap:8 pcma/8000
a=sendrecv
a=ptime:20
```

- **Outgoing INVITE to WAN:**

```
INVITE sip: 9721000@ITSP;user=phone;x=y;z=a SIP/2.0
Via: SIP/2.0/UDP 212.179.1.12;branch=z9hGWwan
From: <sip:97000@IP PBX;user=phone;x=y;z=a>;tag=OWan;paramer1=abe
To: <sip: 9721000@ ITSP;user=phone>
Call-ID: USEVWWAN@212.179.1.12
CSeq: 38  INVITE
Contact: <sip:7000@212.179.1.12>
Supported: em,100rel,timer,replaces
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER
User-Agent: Sip Message Generator V1.0.0.5
Content-Type: application/sdp
Content-Length: 155
v=0
o=SMG 5  9  IN IP4 212.179.1.11
s=Phone-Call
c=IN IP4 212.179.1.11
t=0 0
m=audio 8000 RTP/AVP 8
a=rtpmap:8 pcma/8000
a=sendrecv
a=ptime:20
```

The SIP message manipulations in the example above (contributing to typical topology hiding) are as follows:

- Inbound source SIP URI user name from "7000" to "97000":

```
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OlLAN;paramer1=abe
```

to

```
From: <sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;paramer1=abe
```

- Source IP Group name (i.e., SIP URI host name) from "10.2.2.6" to "IP_PBX":

```
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OlLAN;paramer1=abe
```

to

```
From: <sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;paramer1=abe
```

- Inbound destination SIP URI user name from "1000" to 9721000":

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
To: <sip:1000@10.2.2.3;user=phone>
```

to

```
INVITE sip:9721000@ITSP;user=phone;x=y;z=a SIP/2.0
To: <sip:9721000@ITSP;user=phone>
```

- Destination IP Group name (SIP URI host name) from "10.2.2.3" to "ITSP":

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
To: <sip:1000@10.2.2.3;user=phone>
```

to

```
INVITE sip:9721000@ITSP;user=phone;x=y;z=a SIP/2.0
To: <sip:9721000@ITSP;user=phone>
```

22.7.1 Configuring IP-to-IP Inbound Manipulations

The IP to IP Inbound Manipulation table lets you configure up to 200 IP-to-IP Inbound Manipulation rules. An IP-to-IP Inbound Manipulation rule defines a manipulation sequence for the source or destination SIP URI user part of inbound SIP dialog requests. You can apply these manipulations to different SIP dialog message types (e.g., INVITE or REGISTER) and SIP headers as follows:

- Manipulated destination URI user part are done on the following SIP headers: Request-URI, To, and Remote-Party-ID (if exists)
- Manipulated source URI user part are done on the following SIP headers: From, P-Asserted-Identity (if exists), P-Preferred-Identity (if exists), and Remote-Party-ID (if exists)

An IP-to-IP Inbound Manipulation rule includes two areas:

- Matching characteristics (Rule) - characteristics of incoming SIP dialog such as source host name.
- Operation (Action) - if the incoming call matches the characteristics of the rule, the device manipulates the source or destination SIP URI user part of the SIP dialog (e.g., removes user-defined number of characters from the left of the SIP URI user part).

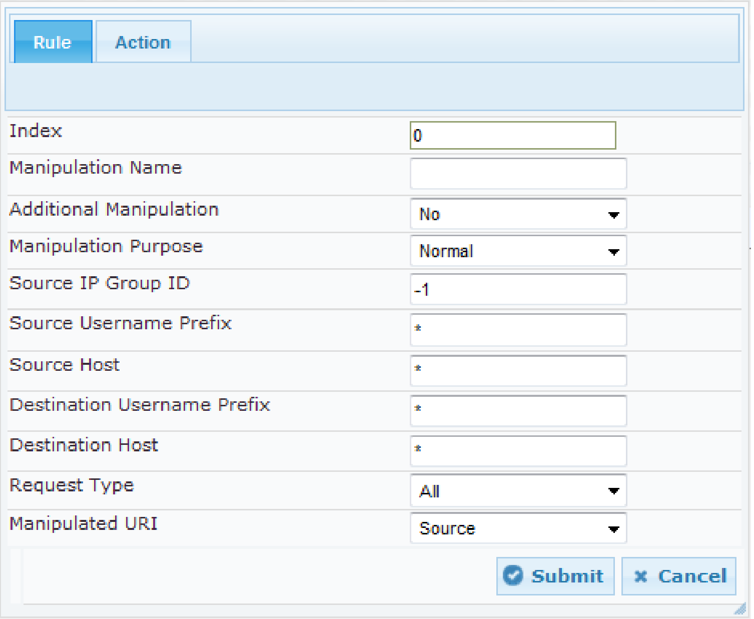
Note: The IP Group table can be used to configure a host name that overwrites the received host name. This manipulation can be done for source and destination IP Groups (see "Configuring IP Groups" on page 174).

The procedure below describes how to configure IP-to-IP Inbound Manipulation rules in the Web interface. You can also configure these rules using the table ini file parameter, IPInboundManipulation or CLI command, configure voip > sbc manipulations ip-inbound-manipulation.

➤ To configure an IP-to-IP Inbound Manipulation rule:

1. Open the IP to IP Inbound Manipulation page (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP to IP Inbound**).
2. Click **Add**; the following dialog box appears:

Figure 109: IP to IP Inbound Manipulation Page - Add Dialog Box



Index	0
Manipulation Name	
Additional Manipulation	No
Manipulation Purpose	Normal
Source IP Group ID	-1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Request Type	All
Manipulated URI	Source

3. Configure the IP-to-IP inbound manipulation rule according to the parameters described in the table below.

4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 75: IP to IP Inbound Manipulation Parameter Descriptions

Parameter	Description
Index **[IPInboundManipulation_Index]**	Defines an index number for the new table record.
Manipulation Name CLI: manipulation-name **[IPInboundManipulation_ManipulationName]**	Defines an arbitrary name to easily identify the manipulation rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Characteristics - Rule	
Additional Manipulation CLI: is-additional-manipulation **[IPInboundManipulation_IsAdditionalManipulation]**	Determines whether additional SIP URI user part manipulation is done for the table entry rule listed directly above it. ▪ **[0]** No = (Default) Regular manipulation rule (not done in addition to the rule above it). ▪ **[1]** Yes = If the above row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule. **Note:** Additional manipulation can only be done on a different SIP URI, source or destination, to the rule configured in the row above as configured by the 'Manipulated URI' parameter (see below).
Manipulation Purpose CLI: purpose **[IPInboundManipulation_ManipulationPurpose]**	Defines the purpose of the manipulation: ▪ **[0]** Normal = (Default) Inbound manipulations affect the routing input and source and/or destination number. ▪ **[1]** Routing input only = Inbound manipulations affect the routing input only, retaining the original source and destination number. ▪ **[2]** Shared Line = Used for the Shared-Line Appearance feature. This manipulation is for registration requests to change the destination number of the secondary extension numbers to the primary extension. For more information, see "BroadSoft's Shared Phone Line Call Appearance for SBC Survivability" on page 234.
Source IP Group ID CLI: src-ip-group-id **[IPInboundManipulation_SrcIpGroup]**	Defines the IP Group from where the incoming INVITE is received. The default is -1 (i.e., any IP Group).
Source Username Prefix CLI: src-user-name-prefix **[IPInboundManipulation_SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name (usually in the From header). The default is the asterisk (*) symbol (i.e., any source username prefix). **Note:** The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379.
Source Host CLI: src-host **[IPInboundManipulation_SrcHost]**	Defines the source SIP URI host name - full name (usually in the From header). The default is the asterisk (*) symbol (i.e., any host name).
Destination Username Prefix CLI: dst-user-name-prefix **[IPInboundManipulation_DestUsernamePrefix]**	Defines the prefix of the destination SIP URI user name, usually located in the Request-URI. The default is the asterisk (*) symbol (i.e., any destination username prefix). **Note:** The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379.
Destination Host CLI: dst-host	Defines the destination SIP URI host name - full name, usually located

Parameter	Description
[IPInboundManipulation_DestHost]	in the Request URI. The default is the asterisk (*) symbol (i.e., any destination host name).
Request Type CLI: request-type **[IPInboundManipulation_RequestType]**	Defines the SIP request type to which the manipulation rule is applied. ▪ **[0]** All = (Default) All SIP messages. ▪ **[1]** INVITE = All SIP messages except REGISTER and SUBSCRIBE. ▪ **[2]** REGISTER = Only REGISTER messages. ▪ **[3]** SUBSCRIBE = Only SUBSCRIBE messages. ▪ **[4]** INVITE and REGISTER = All SIP messages except SUBSCRIBE. ▪ **[5]** INVITE and SUBSCRIBE = All SIP messages except REGISTER.
Manipulated URI CLI: manipulated-uri **[IPInboundManipulation_ManipulatedURI]**	Determines whether the source or destination SIP URI user part is manipulated. ▪ **[0]** Source = (Default) Manipulation is done on the source SIP URI user part. ▪ **[1]** Destination = Manipulation is done on the destination SIP URI user part.
Operation Rule - Action	
Remove From Left CLI: remove-from-left **[IPInboundManipulation_RemoveFromLeft]**	Defines the number of digits to remove from the left of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".
Remove From Right CLI: remove-from-right **[IPInboundManipulation_RemoveFromRight]**	Defines the number of digits to remove from the right of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "j". **Note:** If both 'Remove From Right' and 'Leave From Right' parameters are configured, the 'Remove From Right' setting is applied first.
Leave From Right CLI: leave-from-right **[IPInboundManipulation_LeaveFromRight]**	Defines the number of characters that you want retained from the right of the user name. **Note:** If both 'Remove From Right' and 'Leave From Right' parameters are configured, the 'Remove From Right' setting is applied first.
Prefix to Add CLI: prefix-to-add **[IPInboundManipulation_Prefix2Add]**	Defines the number or string that you want added to the front of the user name. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn".
Suffix to Add CLI: suffix-to-add **[IPInboundManipulation_Suffix2Add]**	Defines the number or string that you want added to the end of the user name. For example, if you enter '01' and the user name is "john", the new user name is "john01".

22.7.2 Configuring IP-to-IP Outbound Manipulations

The IP to IP Outbound Manipulation table lets you configure up to 200 IP-to-IP Outbound Manipulation rules. An IP-to-IP Outbound Manipulation rule defines a manipulation action for the SIP Request-URI user part (source or destination) or calling name of outbound SIP dialog requests. The IP-to-IP Outbound Manipulation rules can be applied to any SIP request type (e.g., INVITE). Manipulated destination URI user part are done on the SIP headers - Request URI, To, and Remote-Party-ID (if exists). Manipulated source URI user part are done on the SIP headers - From, P-Asserted (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists).

An IP-to-IP Outbound Manipulation rule includes two areas:

- Matching characteristics (Rule) - characteristics of incoming SIP dialog such as source host name. As the device performs outbound manipulations only after the routing process, the IP-to-IP Outbound Manipulation rule can also use destination IP Groups as matching characteristics.
- Operation (Action) - if the incoming call matches the characteristics of the rule, the device manipulates the source or destination SIP URI user part or calling name of the SIP dialog (e.g., removes user-defined number of characters from the left of the SIP URI user part).

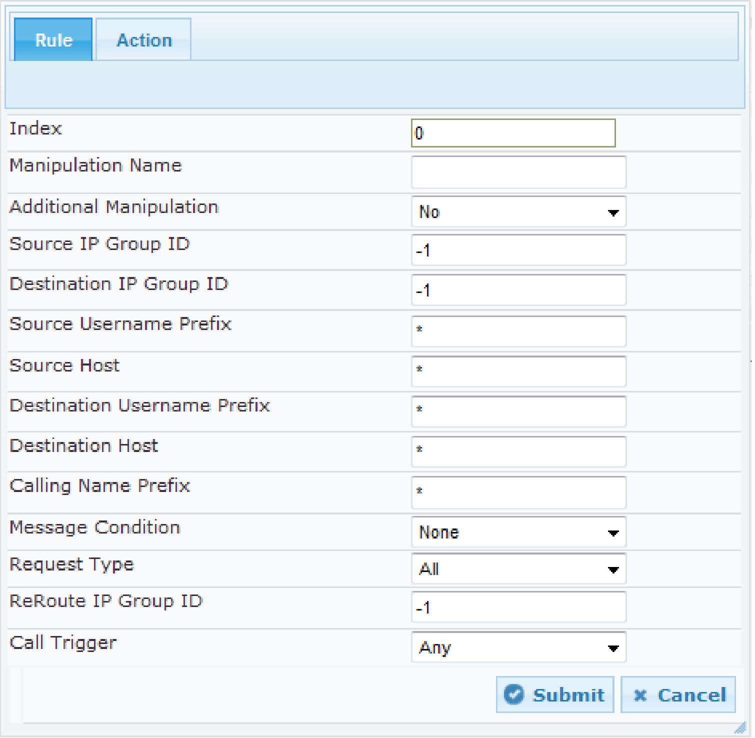
Note: SIP URI host name (source and destination) manipulations can also be configured in the IP Group table. These manipulations are simply host name substitutions with the names configured for the source and destination IP Groups, respectively.

The procedure below describes how to configure IP-to-IP Outbound Manipulation rules in the Web interface. You can also configure these rules using the table ini file parameter, IPOutboundManipulation or CLI command, configure voip > sbc manipulations ip-outbound-manipulation.

➤ To configure IP-to-IP outbound manipulation rules:

1. Open the IP to IP Outbound Manipulation page (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP to IP Outbound**).
2. Click **Add**; the following dialog box appears:

Figure 110: IP to IP Outbound Manipulation Page - Add Dialog Box



Index	0
Manipulation Name	
Additional Manipulation	No
Source IP Group ID	-1
Destination IP Group ID	-1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Calling Name Prefix	*
Message Condition	None
Request Type	All
ReRoute IP Group ID	-1
Call Trigger	Any

3. Configure an IP-to-IP outbound manipulation rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 76: IP to IP Outbound Manipulation Table Parameter Description

Parameter	Description
Index	Defines an index number for the new table record.

Parameter	Description
[IPOutboundManipulation_Index]	
Manipulation Name CLI: manipulation-name **[IPOutboundManipulation_ManipulationName]**	Defines an arbitrary name to easily identify the manipulation name. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Characteristics - Rule	
Additional Manipulation CLI: is-additional-manipulation **[IPOutboundManipulation_IsAdditionalManipulation]**	Determines whether additional manipulation is done for the table entry rule listed directly above it. ▪ **[0]** No = (Default) Regular manipulation rule - not done in addition to the rule above it. ▪ **[1]** Yes = If the previous table row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule. **Note:** Additional manipulation can only be done on a different item (source URI, destination URI, or calling name) to the rule configured in the row above (configured by the 'Manipulated URI' parameter).
Source IP Group ID CLI: src-ip-group-id **[IPOutboundManipulation_SrcIPGroupID]**	Defines the IP Group from where the INVITE is received. The default values is -1 (i.e., any IP Group).
Destination IP Group ID CLI: dst-ip-group-id **[IPOutboundManipulation_DestIPGroupID]**	Defines the IP Group to where the INVITE is to be sent. The default values is -1 (i.e., any IP Group).
Source Username Prefix CLI: src-user-name-prefix **[IPOutboundManipulation_SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name, typically used in the SIP From header. The default value is the asterisk (*) symbol (i.e., any source username prefix). The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379.
Source Host CLI: src-host **[IPOutboundManipulation_SrcHost]**	Defines the source SIP URI host name - full name, typically in the From header. The default value is the asterisk (*) symbol (i.e., any source host name).
Destination Username Prefix CLI: dst-user-name-prefix **[IPOutboundManipulation_DestUsernamePrefix]**	Defines the prefix of the destination SIP URI user name, typically in the Request-URI. The default value is the asterisk (*) symbol (i.e., any destination username prefix). The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 379.
Destination Host CLI: dst-host **[IPOutboundManipulation_DestHost]**	Defines the destination SIP URI host name - full name, typically in the Request-URI. The default value is the asterisk (*) symbol (i.e., any destination host name).
Calling Name Prefix CLI: calling-name-prefix **[IPOutboundManipulation_CallingNamePrefix]**	Defines the prefix of the calling name (caller ID). The calling name appears in the SIP From header. The valid value is a string of up to 37 characters. By default, no prefix is defined.
Message Condition CLI: message-condition **[IPOutboundManipulation_MessageC**	Assigns a Message Condition rule as a matching characteristic. Message Condition rules define required SIP message formats. For configuring Message Condition rules, see "Configuring Message Condition Rules" on

Parameter	Description
ondition]	page 254.
Request Type CLI: request-type **[IPOutboundManipulation_RequestType]**	Defines the SIP request type to which the manipulation rule is applied. ▪ **[0]** All = (Default) all SIP messages. ▪ **[1]** INVITE = All SIP messages except REGISTER and SUBSCRIBE. ▪ **[2]** REGISTER = Only SIP REGISTER messages. ▪ **[3]** SUBSCRIBE = Only SIP SUBSCRIBE messages. ▪ **[4]** INVITE and REGISTER = All SIP messages except SUBSCRIBE. ▪ **[5]** INVITE and SUBSCRIBE = All SIP messages except REGISTER.
ReRoute IP Group ID CLI: re-route-ip-group-id **[IPOutboundManipulation_ReRouteIPGroupID]**	Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. This parameter is typically used for re-routing requests (e.g., INVITEs) when interworking is required for SIP 3xx redirect responses or REFER messages. The default is -1 (i.e., not configured). **Notes:** ▪ This parameter functions together with the 'Call Trigger' parameter (see below). ▪ For more information on interworking of SIP 3xx redirect responses or REFER messages, see "Interworking SIP 3xx Redirect Responses" on page 227 and "Interworking SIP REFER Messages" on page 229, respectively.
Call Trigger CLI: trigger **[IPOutboundManipulation_Trigger]**	Defines the reason (i.e, trigger) for the re-routing of the SIP request: ▪ **[0]** Any = (Default) Re-routed for all scenarios (re-routes and non-re-routes). ▪ **[1]** 3xx = Re-routed if it triggered as a result of a SIP 3xx response. ▪ **[2]** REFER = Re-routed if it triggered as a result of a REFER request. ▪ **[3]** 3xx or REFER = Applies to options [1] and [2]. ▪ **[4]** Initial only = Regular requests that the device forwards to a destination. In other words, re-routing of requests triggered by the receipt of REFER or 3xx does not apply.
Operation Manipulation Rule - Action	
Manipulated Item CLI: manipulated-uri **[IPOutboundManipulation_IsAdditionalManipulation]**	Defines the element in the SIP message that you want manipulated. ▪ **[0]** Source URI = (Default) Manipulates the source SIP Request-URI user part. ▪ **[1]** Destination URI = Manipulates the destination SIP Request-URI user part. ▪ **[2]** Calling Name = Manipulates the calling name in the SIP message.
Remove From Left CLI: remove-from-left **[IPOutboundManipulation_RemoveFromLeft]**	Defines the number of digits to remove from the left of the manipulated item prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".
Remove From Right CLI: remove-from-right **[IPOutboundManipulation_RemoveFromRight]**	Defines the number of digits to remove from the right of the manipulated item prefix. For example, if you enter 3 and the user name is "john", the new user name is "j".
Leave From Right CLI: leave-from-right **[IPOutboundManipulation_LeaveFromRight]**	Defines the number of digits to keep from the right of the manipulated item.

Parameter	Description
Prefix to Add CLI: prefix-to-add **[IPOutboundManipulation_Prefix2Add]**	Defines the number or string to add in the front of the manipulated item. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn". If you set the 'Manipulated Item' parameter to **Source URI** or **Destination URI**, you can configure this parameter to a string of up 49 characters. If you set the 'Manipulated Item' parameter to **Calling Name**, you can configure this parameter to a string of up 36 characters.
Suffix to Add CLI: suffix-to-add **[IPOutboundManipulation_Suffix2Add]**	Defines the number or string to add at the end of the manipulated item. For example, if you enter '01' and the user name is "john", the new user name is "john01". If you set the 'Manipulated Item' parameter to **Source URI** or **Destination URI**, you can configure this parameter to a string of up 49 characters. If you set the 'Manipulated Item' parameter to **Calling Name**, you can configure this parameter to a string of up 36 characters.
Privacy Restriction Mode CLI: privacy-restriction-mode **[IPOutboundManipulation_PrivacyRestrictionMode]**	Determines user privacy handling (i.e., restricting source user identity in outgoing SIP dialogs). ▪ **[0]** Transparent = (Default) No intervention in SIP privacy. ▪ **[1]** Don't change privacy = The user identity remains the same as in the incoming SIP dialog. If a restricted number exists, the restricted presentation is normalized as follows: ✓ From URL header: anonymous@anonymous.invalid. ✓ If a P-Asserted-Identity header exists (either in the incoming SIP dialog or added by the device), a Privacy header is added with the value "id". ▪ **[2]** Restrict = The user identity is restricted (the restricted presentation is as mentioned above). ▪ **[3]** Remove Restriction = The device attempts to reveal the user identity by setting user values in the From header and removing the privacy "id" value if the Privacy header exists. If the From header user is anonymous, the value is taken from the P-Preferred-Identity, P-Asserted-Identity, or Remote-Party-ID header (if exists). The device identifies an incoming user as restricted if one of the following exists: ▪ From header user is anonymous. ▪ P-Asserted-Identity and Privacy headers contain the value "id". **Note:** All restriction logic is performed after the user number has been manipulated.

Reader's Notes

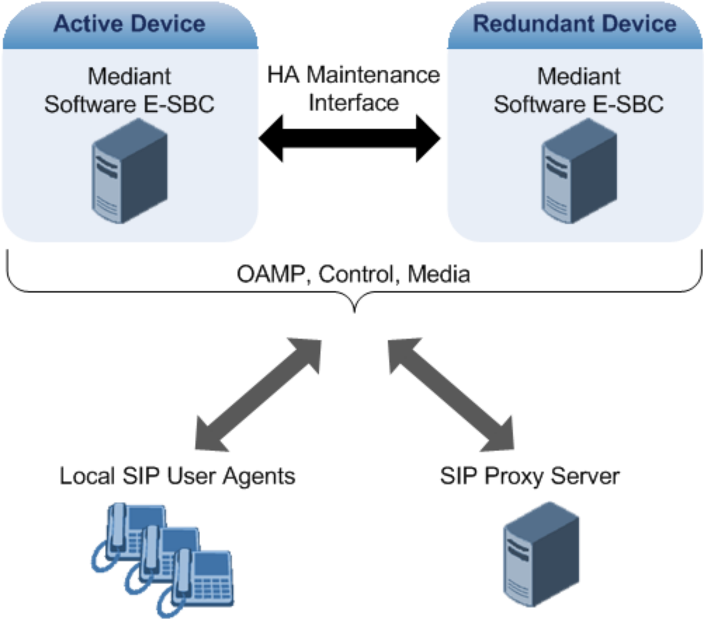
Part VI

High Availability System

23 HA Overview

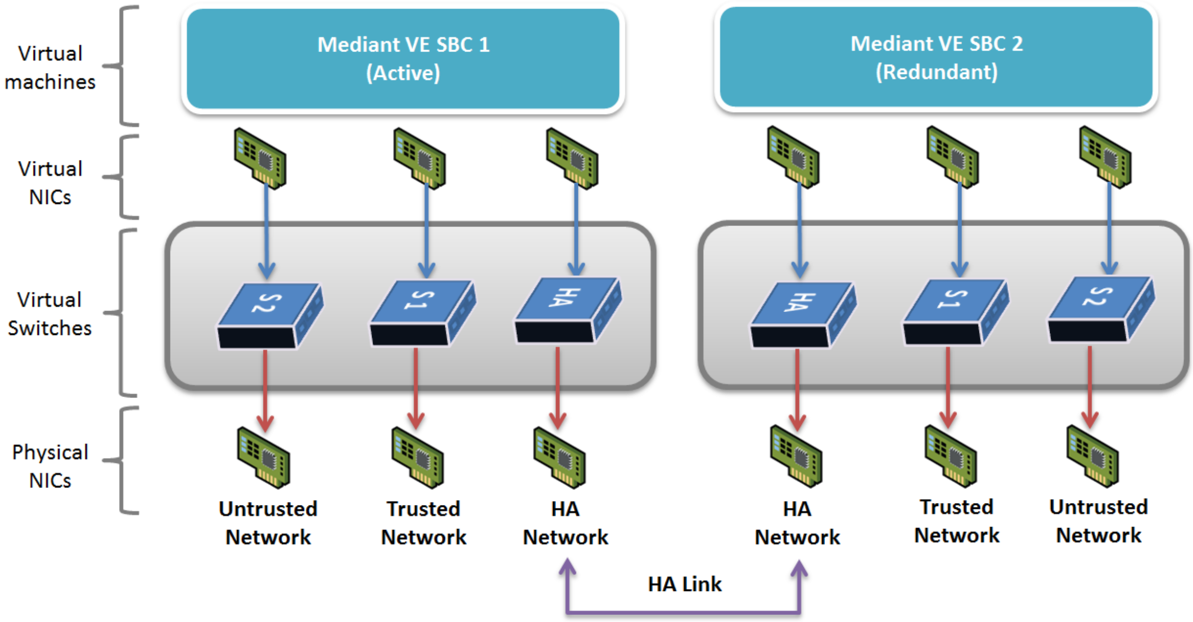
The device's High Availability (HA) feature provides 1+1 system redundancy using two Mediant Software E-SBC devices. If failure occurs in the active device, a switchover occurs to the redundant device which takes over the call handling process. Thus the continuity of call services is ensured. All active calls (signaling and media) are maintained upon switchover.

The figure below illustrates the Active-Redundant HA devices under normal operation. Communication between the two devices is through a Maintenance interface, having a unique IP address for each device. The devices have identical software and configuration including network interfaces (i.e., OAMP, Control, and Media), and have identical local-port cabling of these interfaces.



The figure below shows two Virtual Machines -- Mediant VE SBCs -- running on different servers to work in an HA configuration:

Figure 111: Mediant VE SBC HA - Virtual Network Setup



23.1 Connectivity and Synchronization between Devices

In HA mode, the Ethernet connectivity between the two devices is through a special LAN interface on each device, referred to as the *Maintenance* interface. Each device has its own Maintenance interface with a unique address, and each device knows the Maintenance address of the other. The Maintenance interface can use a dedicated Ethernet port group or share the same Ethernet port group with the other network interface types (i.e., OAMP, Media, and Control).

When only one of the devices is operational it is in HA stand-alone state. This means that the device has no connectivity to the second device. When the second device is powered up, it recognizes the active device through the Maintenance network and acquires the HA redundant state. It then begins synchronizing for HA with the active device through the Maintenance network. During synchronization, the active device sends the redundant device its current configuration settings, including auxiliary files. The active device also sends its software file (.cmp) if the redundant device is running a different software version. Once loaded to the redundant device, the redundant device reboots to apply the new configuration and/or software.

Thus, under normal operation, one of the devices is in active state while the other is in redundant state, where both devices share the same configuration and software. Any subsequent configuration update or software upgrade on the active device is also done on the redundant device.

In the active device, all logical interfaces (i.e., Media, Control, OAMP, and Maintenance) are active. In the redundant device, only the Maintenance interface is active, which is used for connectivity to the active device. Therefore, management is done only through the active device. Upon a failure in the active device, the redundant device becomes active and activates all its logical interfaces exactly as was used on the active device.

23.2 Device Switchover upon Failure

When a failure occurs in the active device, a switchover occurs to the redundant device making it the new active device. Whether a switchover is later done back to the repaired failed device, depends on whether you have enabled the Revertive mode:

- **Revertive mode enabled:** The Revertive mode specifies one of the device's as the "preferred" device between the two devices. This is done by assigning a priority level to each device (1 to 10, where 1 is the lowest). Whenever the device with higher priority recovers from a failure, it first becomes the redundant device but then initiates a switchover to become the active device once again; otherwise, after recovery, it becomes the redundant device and remains as redundant. If you change the priority level of the redundant device to one that is higher than the active device and then reset the redundant device, a switchover occurs to the redundant device making it the active device and the "preferred" device. If both devices are configured with the same priority level, then Revertive mode is irrelevant.
- **Revertive mode disabled:** A switchover is done only upon failure of the currently active device.

Failure detection by the devices is done by the constant keep-alive messages they send between themselves to verify connectivity. Upon detection of a failure in one of the devices, the following occurs:

- **Failure in active device:** The redundant device initiates a switchover. The failed device resets and the previously redundant device becomes the active device in stand-alone mode. If at a later stage this newly active device detects that the failed device has been repaired, the system returns to HA mode. If Revertive mode is enabled and the originally active device was configured with a higher priority, a switchover occurs to this device; otherwise, if it was configured with a lower priority (or Revertive mode was disabled), the repaired device is initialized as the redundant device.

- **Failure in redundant device:** The active device moves itself into stand-alone mode until the redundant device is returned to operation. If the failure in the redundant device is repaired after reset, it's initialized as the redundant device once again and the system returns to HA mode.

Connectivity failure triggering a switchover can include, for example, one of the following:

- **Loss of physical (link) connectivity:** If one or more physical network groups (i.e., Ethernet port pair) used for one or more network interfaces of the active device disconnects (i.e., no link) and these physical network groups are connected OK on the redundant device, then a switchover occurs to the redundant device.
- **Loss of network (logical) connectivity:** No network connectivity, verified by keep-alive packets between the devices. This applies only to the Maintenance interface.

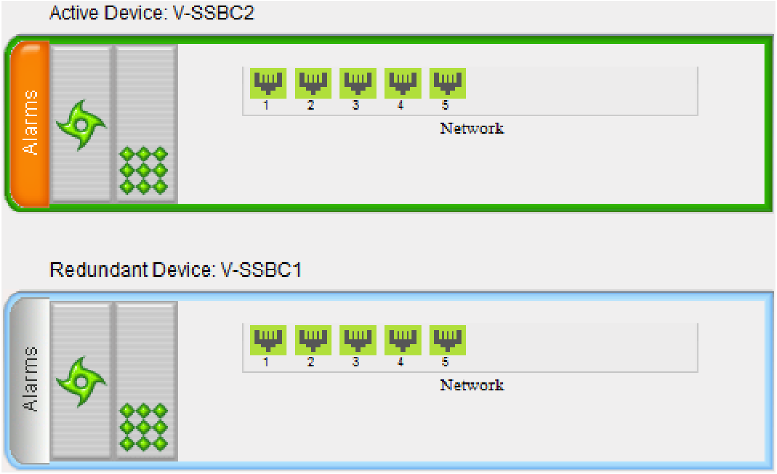
Notes:

- Switchover triggered by loss of physical connectivity in one or more Ethernet port-group is not done if the active device has been set to a Revertive priority level of 10. In such a scenario, the device remains active despite the loss of connectivity in one or more of its Ethernet port groups.
- After HA switchover, the active device updates other hosts in the network about the new mapping of its Layer-2 hardware address to the global IP address, by sending a broadcast gratuitous Address Resolution Protocol (ARP) message.

23.3 HA Status on the Home Page

The Home page of the device's Web interface displays the status of the HA system. The Home page provides a graphical display of both active and redundant devices.

Figure 112: Home Page Display of HA System



- Active device:
  - Color border: The active device is surrounded by a green border.
  - Title: The default title of the device is Active Device: "Device 1".
- Redundant device:
  - Color border: The redundant device is surrounded by a blue border.
  - Title: The default title of the device is Redundant Device: "Device 2".

The title of each device can be configured as described below:

➢ To define a name for the device:

1. Open the HA Settings page (**Configuration** tab > **System** menu > **HA Settings**).
2. In the 'HA Device Name' field, enter a name for the active device.
3. Click **Submit**.

> **Note:** Once the devices are running in HA mode, you can change the name of the redundant device, through the active device only, in the 'Redundant HA Device Name' field.

The Home page also displays the HA operational status of the device to which you are currently logged in. This is displayed in the 'High Availability' field under the General Information pane:

- "Not Operational": HA is not configured or the installed Software License Key does not include the HA feature
- "Synchronizing": Redundant device is synchronizing with Active device
- "Operational": The device is in HA mode
- "Stand Alone": HA is configured but the Redundant device is missing and HA is currently unavailable
- "Not Available": HA is not configured correctly (error)

24 HA Configuration

This section describes the configuration of the HA system.

24.1 Initial HA Configuration

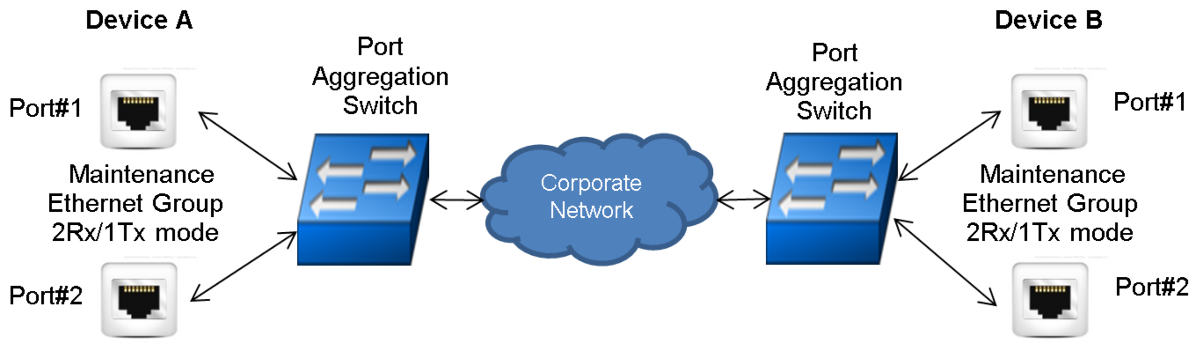
By default, HA is disabled on the device. When a device is loaded with valid HA configuration and it is the first device to be loaded, it becomes the active device. The second device that is loaded with HA configuration becomes the redundant (standby) device.

24.1.1 Network Topology Types and Tx/Rx Ethernet Port Group Settings

The initial configuration of HA depends on how you want to deploy your HA system in the network. The Maintenance Interface, used for the HA link between Active and Redundant units, should be configured on a dedicated Ethernet Port Group, separate from the other interfaces. The required transmit (TX) / receive (Rx) mode for the port pair in the Ethernet Port Group used by the Maintenance interface is as follows:

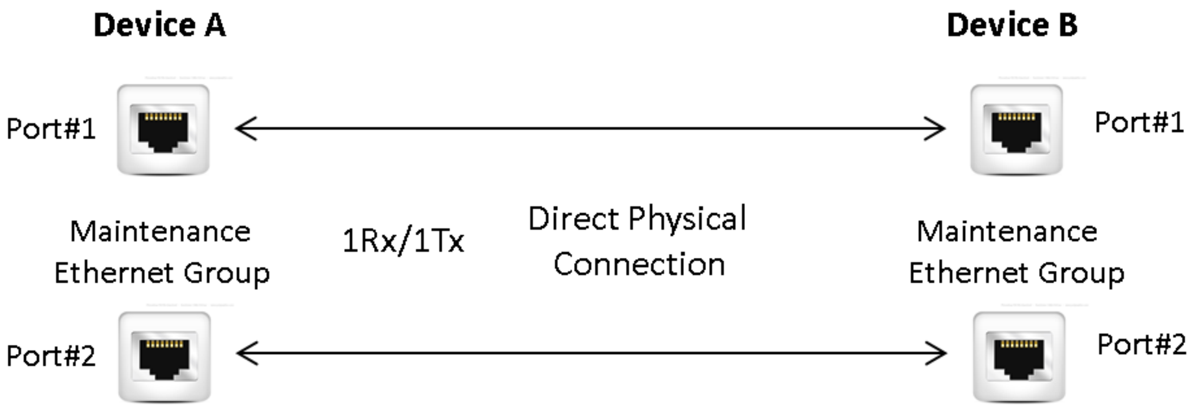
- For Geographical HA (both units are located far from each other), **2Rx/1Tx** port mode connected to a port aggregation switch is the recommended option:

Figure 113: Rx/Tx Mode for Geographical HA



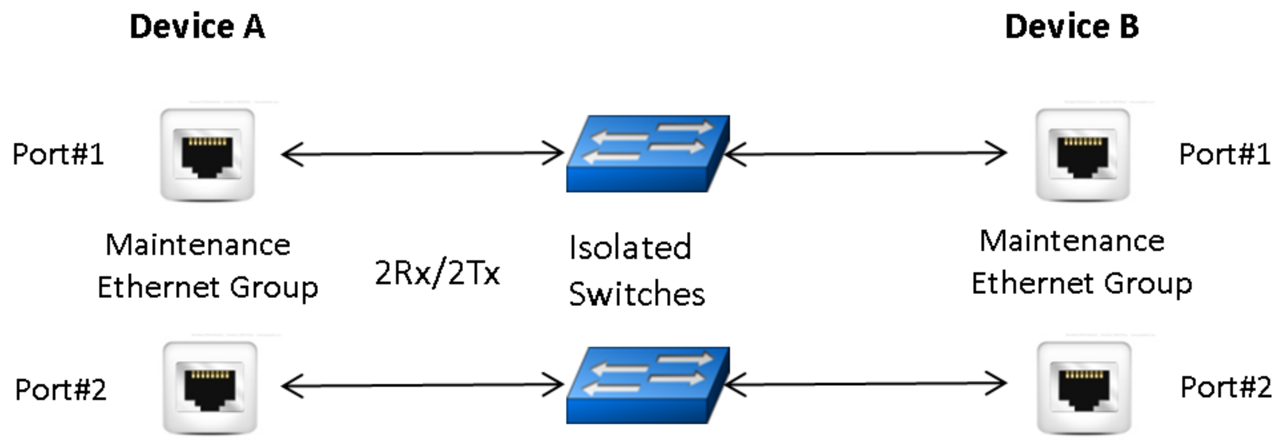
- If the Maintenance ports of both devices are connected directly to each other without intermediation of switches, configure the mode to **1RX/1TX**:

Figure 114: Rx/Tx Mode for Direct Connection


Device A
Device B
Port#1
Port#1
Maintenance Ethernet Group
1Rx/1Tx
Direct Physical Connection
Maintenance Ethernet Group
Port#2
Port#2

- If the two devices are connected through two (or more) isolated LAN switches (i.e., packets from one switch cannot traverse the second switch), configure the mode to **2RX/2TX**:

Figure 115: Redundancy Mode for Two Isolated Switches



Notes:

- When two LAN switches are used, the LAN switches must be in the same subnet (i.e., broadcast domain).
- To configure Tx/Rx modes of the Ethernet ports, see "Network Topology Types and Tx/Rx Ethernet Port Group Settings" on page 279.

24.1.2 Configuring the HA Devices

This section describes how to initially configure the two devices comprising the HA system. This configuration is done in the following chronological order:

1. Configuring the first device for HA - see "Step 1: Configure the First Device" on page 280
2. Configuring the second device for HA - see "Step 2: Configure the Second Device" on page 282
3. Activating HA on the devices - see "Step 3: Initialize HA on the Devices" on page 283

Notes:

- The HA feature is available only if both devices are installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 309.
- The physical connections of the first and second devices to the network (i.e., Maintenance interface and OAMP, Control and Media interfaces) **must be identical**. This also means that the two devices must also use the same Ethernet Port Groups and the port numbers belonging to these Ethernet Port Groups. For example, if the first device uses Ethernet Port Group 1 (with ports 1 and 2), the second device must also use Ethernet Port Group 1 (with ports 1 and 2).
- Before configuring HA, determine the required network topology, as described in "Network Topology Types and Tx/Rx Ethernet Port Group Settings" on page 279.
- The Maintenance network should be able to perform a fast switchover in case of link failure and thus, Spanning Tree Protocol (STP) should not be used in this network; the Ethernet connectivity of the Maintenance interface between the two devices should be constantly reliable without any disturbances.

24.1.2.1 Step 1: Configure the First Device

The first stage is to configure the first device for HA, as described in the procedure below:

Note: During this stage, ensure that the second device is powered off or disconnected from the network.

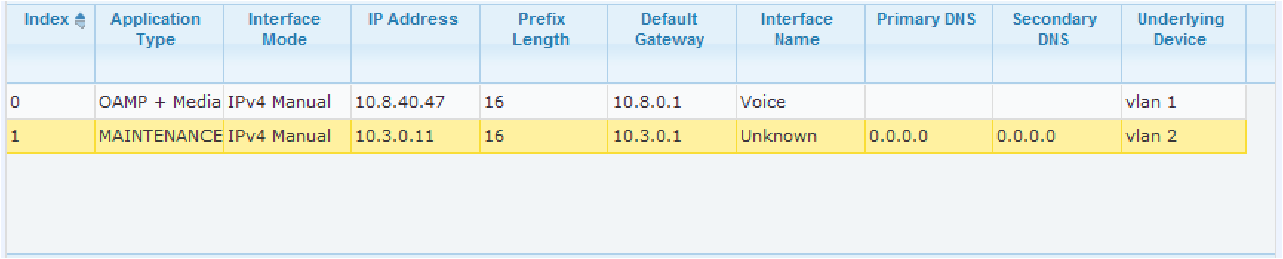
➢ To configure the first device for HA:

1. Configure the network interfaces, including the default OAMP interface:
   a. If you are already connected to the SBC via keyboard and monitor, change the OAMP parameters to suite your networking scheme, using CLI (refer to the Installation Manual).
   b. Connect to the SBC's Web interface with the newly assigned OAMP IP address.
   c. Open the Interface table (**Configuration** tab > **VoIP** menu > **Network** > **IP Interfaces Table**).
   d. Configure the Control and Media network interfaces, as required.
   e. Add the HA Maintenance interface (i.e., the **MAINTENANCE** Application Type).

Note: Make sure that the MAINTENANCE interface uses an Ethernet Port Group that is not used by any other network interface. The Ethernet Port Group is associated with the Ethernet Device assigned to the interface in the 'Underlying Interface' field.

The Interface table below shows an example where the Maintenance interface is assigned to Ethernet Device "vlan 2" (which is associated with Ethernet Port Group "GROUP_2") in the 'Underlying Device' field, while the other interface is assigned to "vlan 1" (associated with "GROUP_1"):

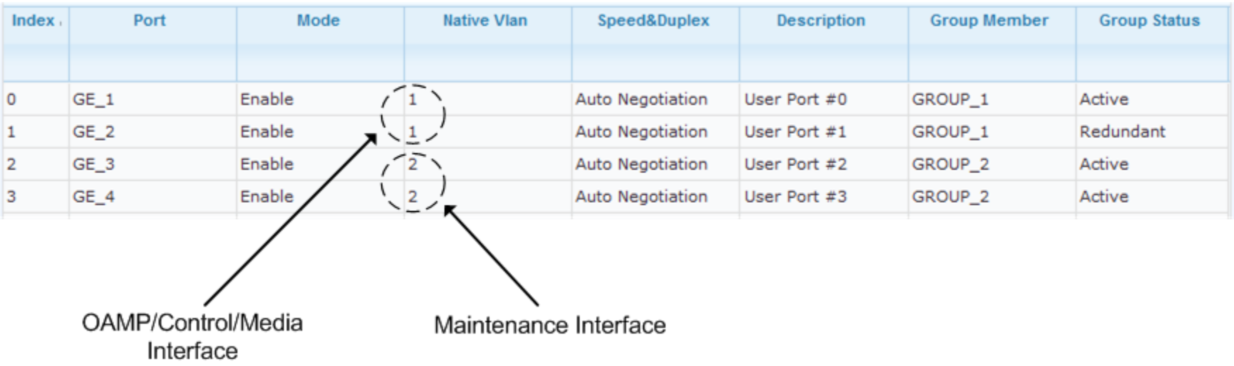
Figure 116: Configured MAINTENANCE Interface in Interface Table



Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Interface Name	Primary DNS	Secondary DNS	Underlying Device
0	OAMP + Media	IPv4 Manual	10.8.40.47	16	10.8.0.1	Voice			vlan 1
1	MAINTENANCE	IPv4 Manual	10.3.0.11	16	10.3.0.1	Unknown	0.0.0.0	0.0.0.0	vlan 2

2. If the connection is through a switch, the packets of both interfaces should generally be untagged. In such a scenario, set the Native VLAN ID of each Ethernet Port Group so that it is the same as the VLAN ID set for each interface assigned to that Ethernet Port Group. The Native VLAN ID is configured in the Physical Ports Settings page (see "Configuring Physical Ethernet Ports" on page 87). The figure below shows an example whereby the Native VLAN IDs of the Ethernet Port Groups are set to the same VLAN IDs of the interfaces using these Ethernet Port Groups:

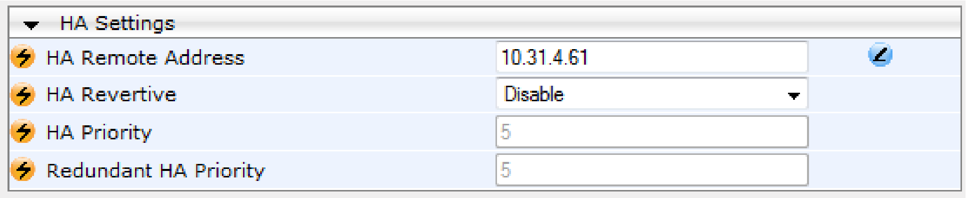
Figure 117: Native VLAN for Ethernet Port Groups of Maintenance and Other Interfaces



Index	Port	Mode	Native Vlan	Speed&Duplex	Description	Group Member	Group Status
0	GE_1	Enable	1	Auto Negotiation	User Port #0	GROUP_1	Active
1	GE_2	Enable	1	Auto Negotiation	User Port #1	GROUP_1	Redundant
2	GE_3	Enable	2	Auto Negotiation	User Port #2	GROUP_2	Active
3	GE_4	Enable	2	Auto Negotiation	User Port #3	GROUP_2	Active

3. Set the Ethernet port Tx / Rx mode of the Ethernet Port Group used by the Maintenance interface. This is configured in the Ethernet Group Settings page (see "Configuring Tx/Rx for Ethernet Port-Pair Groups" on page 88). The port mode depends on the type of Maintenance connection between the devices, as described in "Network Topology Types and Tx/Rx Ethernet Port Group Settings" on page 279.
4. Configure the HA parameters in the HA Settings page (**Configuration** tab > **System** menu > **HA Settings**):

Figure 118: HA Settings Page



HA Settings

HA Remote Address	10.31.4.61
HA Revertive	Disable
HA Priority	5
Redundant HA Priority	5

   a. In the 'HA Remote Address' field, enter the Maintenance IP address of the **second** device.
   b. (Optional) Enable the Revertive mode by setting the 'HA Revertive' parameter to **Enable** and then setting the priority level of this device in the 'HA Priority' field.
5. Burn the configuration to flash **without** a reset.
6. Power down the device.
7. Continue to "Step 2: Configure the Second Device" on page 282 for configuring the second device.

24.1.2.2 *Step 2: Configure the Second Device*

Once you have configured the first device for HA, you can configure the second device for HA. As the configuration of the second device is similar to the first device, the procedure below briefly describes each procedural step. For detailed configuration such as the path to the Web configuration pages, refer to the section on configuring the first device ("Step 1: Configure the First Device" on page 280).

Note: During this stage, ensure that the first device is powered off or disconnected from the network.

➢ To configure the second device for HA:

1. Connect to the device in the same way as you did with the first device.
2. Configure the **same** OAMP, Media, and Control interfaces as you configured for the first device.
3. Configure a Maintenance interface for this device. The IP address must be different to that configured for the Maintenance interface of the first device. However, the Maintenance interfaces of the devices must be in the same subnet.
4. Configure the **same** Native VLAN IDs of the Ethernet Port Groups and VLAN IDs of the network

interfaces as you configured for the first device.

5. Configure the **same** Ethernet port Tx / Rx mode of the Ethernet Port Group used by the Maintenance interface as you configured for the first device.
6. Configure the HA parameters in the HA Settings page (**Configuration** tab > **System** menu > **HA Settings**):
   a. In the 'HA Remote Address' field, enter the Maintenance IP address of the **first** device.
   b. (Optional) Enable the Revertive mode by setting the 'HA Revertive' field to **Enable** and then setting the priority level of this second device in the 'HA Priority' field.
7. Burn the configuration to flash **without** a reset.
8. Power down the device.
9. Continue to "Step 3: Initialize HA on the Devices" on page 283 for completing the HA configuration.

24.1.2.3 Step 3: Initialize HA on the Devices

Once you have configured both devices for HA as described in the previous sections, follow the procedure below to complete and initialize HA so that the devices become operational in HA. This last stage applies to both devices.

➢ To initialize the devices for HA:

1. Cable the devices to the network.

> **Note:** You must connect both ports (two) in the Ethernet Port Group of the Maintenance interface to the network (i.e., two network cables are used). This provides 1+1 Maintenance port redundancy.

2. Power up the devices; the redundant device synchronizes with the active device and updates its configuration according to the active device. The synchronization status is indicated as follows:
   - Active device: The Web interface's Home page displays the HA status as "Synchronizing".

   When synchronization completes successfully, the redundant device resets to apply the received configuration and software.

   When both devices become operational in HA, the HA status is indicated as follows:
   - Both devices: The Web interface's Home page displays the HA status as "Operational".
3. Access the active device with its OAMP IP address and configure the device as required. For information on configuration done after HA is operational, see "Configuration while HA is Operational" on page 283.

24.2 Configuration while HA is Operational

When the devices are operating in HA state, subsequent configuration is as follows:

- All configuration, including HA is done on the active device **only**.
- Non-HA configuration on the active device is automatically updated on the redundant device (through the Maintenance interface).
- HA-related configuration on the active device is automatically updated on the redundant device:
  - Maintenance interface:
    - Modified Maintenance interface address of the active device: this address is set as the new 'HA Remote Address' value on the redundant device.
    - Modified 'HA Remote Address' value on the active device: this address is set as the new Maintenance interface address on the redundant device. This requires a device reset.
    - Modifications on all other Maintenance interface parameters (e.g., Default Gateway and

VLAN ID): updated to the Maintenance interface on the redundant device.

- 'HA Revertive' mode (this requires a device reset).
- 'HA Priority' parameter is set for the active device.
- Modified 'Redundant HA Priority' value is set for the redundant device. This requires a device reset.

Note: If the HA system is already in Revertive mode and you want to change the priority of the device, to ensure that system service is maintained and traffic is not disrupted, it is recommended to set the higher priority to the redundant device and then reset it. After it synchronizes with the active device, it initiates a switchover and becomes the new active device (the former active device resets and becomes the new redundant device).

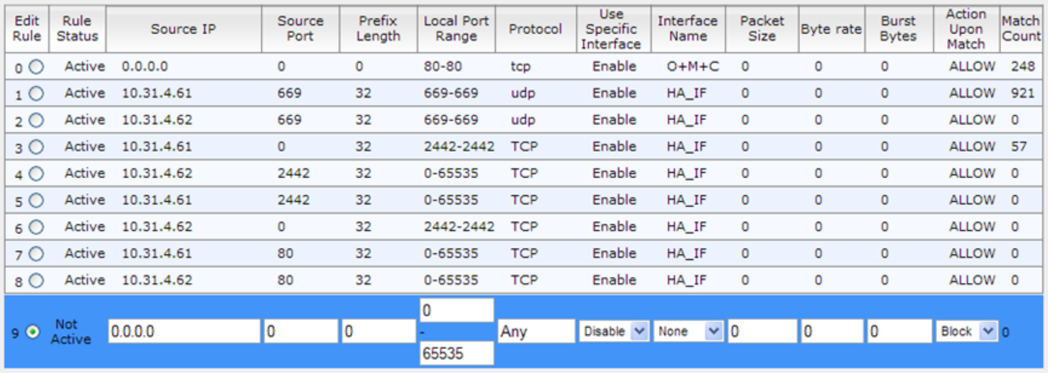
24.3 Configuring Firewall Allowed Rules

If you add firewall rules in the Firewall Settings page (see "Configuring Firewall Settings" on page 116) that block specified traffic, you also need to add rules that ensure traffic related to the HA feature is allowed. These allowed HA rules include the following:

- Keep-alive packets between the HA devices (e.g., rules #1 and #2 in the figure below).
- HA control and data packets between the HA devices (e.g., rules #3 and #4 in the figure below).
- HA control and data packets between the HA devices after switchover (e.g., rules #5 and #6 in the figure below). These rules are the same as rules #3 and #4 respectively, but are required as the TCP source and destination port IDs are not symmetric.
- HTTP protocol for file transferring (e.g., Rule #7 in the figure below).
- HTTP protocol for file transferring after switchover (e.g., Rule #8 - same as Rule #7 - in the figure below).

The figure below displays an example of the required firewall rules. In this example, 10.31.4.61 is the Maintenance interface of the redundant device and 10.31.4.62 is the Maintenance interface of the active device. "HA_IF" is the name of the Maintenance interface.

Figure 119: Allowed Firewall Rules for HA



Edit Rule	Rule Status	Source IP	Source Port	Prefix Length	Local Port Range	Protocol	Use Specific Interface	Interface Name	Packet Size	Byte rate	Burst Bytes	Action Upon Match	Match Count
0 ○	Active	0.0.0.0	0	0	80-80	tcp	Enable	O+M+C	0	0	0	ALLOW	248
1 ○	Active	10.31.4.61	669	32	669-669	udp	Enable	HA_IF	0	0	0	ALLOW	921
2 ○	Active	10.31.4.62	669	32	669-669	udp	Enable	HA_IF	0	0	0	ALLOW	0
3 ○	Active	10.31.4.61	0	32	2442-2442	TCP	Enable	HA_IF	0	0	0	ALLOW	57
4 ○	Active	10.31.4.62	2442	32	0-65535	TCP	Enable	HA_IF	0	0	0	ALLOW	0
5 ○	Active	10.31.4.61	2442	32	0-65535	TCP	Enable	HA_IF	0	0	0	ALLOW	0
6 ○	Active	10.31.4.62	0	32	2442-2442	TCP	Enable	HA_IF	0	0	0	ALLOW	0
7 ○	Active	10.31.4.61	80	32	0-65535	TCP	Enable	HA_IF	0	0	0	ALLOW	0
8 ○	Active	10.31.4.62	80	32	0-65535	TCP	Enable	HA_IF	0	0	0	ALLOW	0
9 ●	Not Active	0.0.0.0	0	0	0 - 65535	Any	Disable	None	0	0	0	Block	0

24.4 Monitoring IP Entity and HA Switchover upon Ping Failure

The device can monitor a specified network entity, using pings. If the device does not receive a ping response from the entity, a switchover to the redundant device occurs. The switchover happens only if a ping was initially successful and then a subsequent ping failed. This feature can be used, for example, to check connectivity with a nearby router (first hop) that the device uses to reach other destinations.

The network entity is defined by IP address. The IP interface from where the ping is sent can be selected from one of the device's configured network interfaces (in the Interface table).

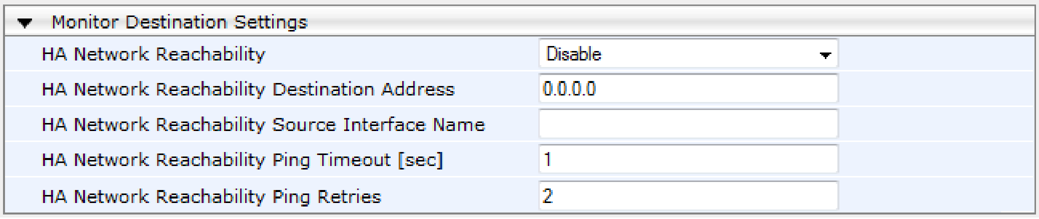
Notes:

- The ping feature is not functional under the following conditions:
  - √ HA is disabled (i.e., active device is in standalone mode).
  - √ HA Priority is used (to prevent endless loops of switchovers).
  - √ Number of Ethernet Groups in the redundant device that are in "up" state are less than on the active device (to prevent endless loops of switchovers).
- For a detailed description of the HA ping parameters, see "HA Ping Parameters" on page 404.

➤ To configure monitoring of IP entity using pings:

1. Open the HA Settings page (**Configuration** tab > **System** menu > **HA Settings**).

Figure 120: HA Settings Page - Monitor Destination Settings



▼ Monitor Destination Settings

HA Network Reachability	Disable
HA Network Reachability Destination Address	0.0.0.0
HA Network Reachability Source Interface Name	
HA Network Reachability Ping Timeout [sec]	1
HA Network Reachability Ping Retries	2

2. Under the Monitor Destination Settings group, do the following:
   - Set the 'HA Network Reachability' field to **Enable**.
   - In the 'HA Network Reachability Destination Address' field, set the address of the IP entity that you want to monitor.
   - In the 'HA Network Reachability Source Iinterface Name' field, set the device's IP network interface from where you want to ping the destination entity.
   - In the 'HA Network Reachability Ping Timeout' field, set the timeout for which the ping request waits for a response.
   - In the 'HA Network Reachability Ping Retries' field, set the number of ping requests that the device sends after no ping response is received from the destination, before the destination is declared unavailable.
3. Click **Submit**.

If this feature is operational, the status of the connectivity to the pinged destination is displayed in the read-only 'Monitor Destination Status' field:

- "Enabled": Ping is sent as configured.
- "Disabled by configuration and HA state": HA and ping are not configured.
- "Disabled by HA state": same as above.
- "Disabled by configuration": same as above.
- "Disabled by invalid configuration": invalid configuration, for example, invalid interface name or destination address (destination address must be different than a local address and from the redundant device's Maintenance address).
- "Disabled by HA priority in use": when HA priority is used, ping mechanism is disabled.
- "Disabled by Eth groups error": when the number of Ethernet Groups in the redundant device becomes less than in the active device, the ping mechanism is disabled.
- "Failed to be activated": Internal error (failed activating the ping mechanism).

25 HA Maintenance

This section describes HA maintenance procedures.

25.1 Maintenance of Redundant Device

The only interface that is operational on the redundant device is the Maintenance interface. For maintenance, there are several protocols available for this interface (unlike the active device which uses the logical OAMP / management interface for these protocols):

- **Syslog:** To receive Syslog messages from the redundant device, ensure that there is a valid VLAN and route configured from the maintenance network to where the Syslog server is located on the network.
- **Telnet:** A Telnet server is always available on the redundant device (even if disabled by configuration).

25.2 Replacing a Failed Device

If you need to replace a non-functional device with a new one, the new device must be configured exactly as the second device, as described in "Configuring the HA Devices" on page 280.

25.3 Forcing a Switchover

If required, you can force a switchover between active and redundant SBCs. For more information, see "High Availability Maintenance" on page 295.

25.4 Software Upgrade

The following types of software upgrades are available on the HA system:

- **Software Upgrade with Device Reset:** Both active and redundant devices burn and reboot with the new software version. This method is quick and simple, but it disrupts traffic (i.e., traffic affecting).
- **Hitless Software Upgrade:** This method maintains service (i.e., not traffic affecting) and is as follows:
  a. The redundant device burns and resets with the new software version.
  b. A switchover is done between the active and redundant devices, whereby the redundant device becomes the active one.
  c. The previously active device burns and resets with the new software version.
  d. The previously active device switches over to become the active device.

For more information on upgrading the software, see "Software Upgrade Wizard" on page 312.

25.5 Rescue Options

The device features a System Snapshots mechanism that provides the capability of returning the system to a previous state. The mechanism may be used as a rescue option if a system malfunction occurs.

Note: For Mediant VE SBC, in addition to the functionality described in this chapter, you can use the snapshots functionality provided by the virtual machine hypervisor.

25.5.1 Taking a Snapshot

Taking a System Snapshot captures a complete state of the device, including:

- Installed software
- Current configuration

- Auxiliary files
- Software License Key

The first 'factory' snapshot is automatically taken when initial installation is performed. Additional snapshots (up to 10) may be taken. The device can be returned to a snapshot, as described below.

➢ To take a snapshot in the CLI:

1. Connect to the CLI interface.
2. At the prompt, type the following and then press Enter:

```
> enable
```

3. At the prompt, type the password and then press Enter:

```
Password: Admin
```

4. At the prompt, type the following to save the current configuration (burn) before creating a snapshot:

```
# write
```

5. **Type the following commands to take a snapshot:**

```
# configure system
# startup-n-recovery
(startup-n-recovery)# create-system-snapshot <name>
```

25.5.2 Viewing Available Snapshots

Currently available system snapshots can be viewed by using the **show-system-snapshots** command. The ‘default’ snapshot is indicated by an asterisk.

```
(startup-n-recovery)# show-system-snapshots
first-install-2010-01-01 03-18-29
pre-production-6.70.037.010-2010-01-08 00-39-58
*production-6.70.037.010-2010-01-08_00-41-30
```

25.5.3 Changing the Default Snapshot

The ‘default’ snapshot indicates a restore point that is used by Automatic Recovery in the case of software malfunction (see "Automatic Recovery" on page 290) and/or Manual Recovery (see "Manual Recovery" on page 287). The last user-created snapshot is automatically set as ‘default’ though it can be changed using the following command:

```
(startup-n-recovery)# set-default-snapshot pre-production-6.70.037.010-2010-01-
08_00-40-27
```

25.5.4 Deleting a Snapshot

To delete a snapshot, use the following command:

```
(startup-n-recovery)# delete-system-snapshot pre-production-6.70.037.010-2010-01-
08_00-39-58
```

25.5.5 Manual Recovery

You can perform a Manual recovery. When the device reboots, a GRUB menu is displayed that lets you select one of the following rescue options:

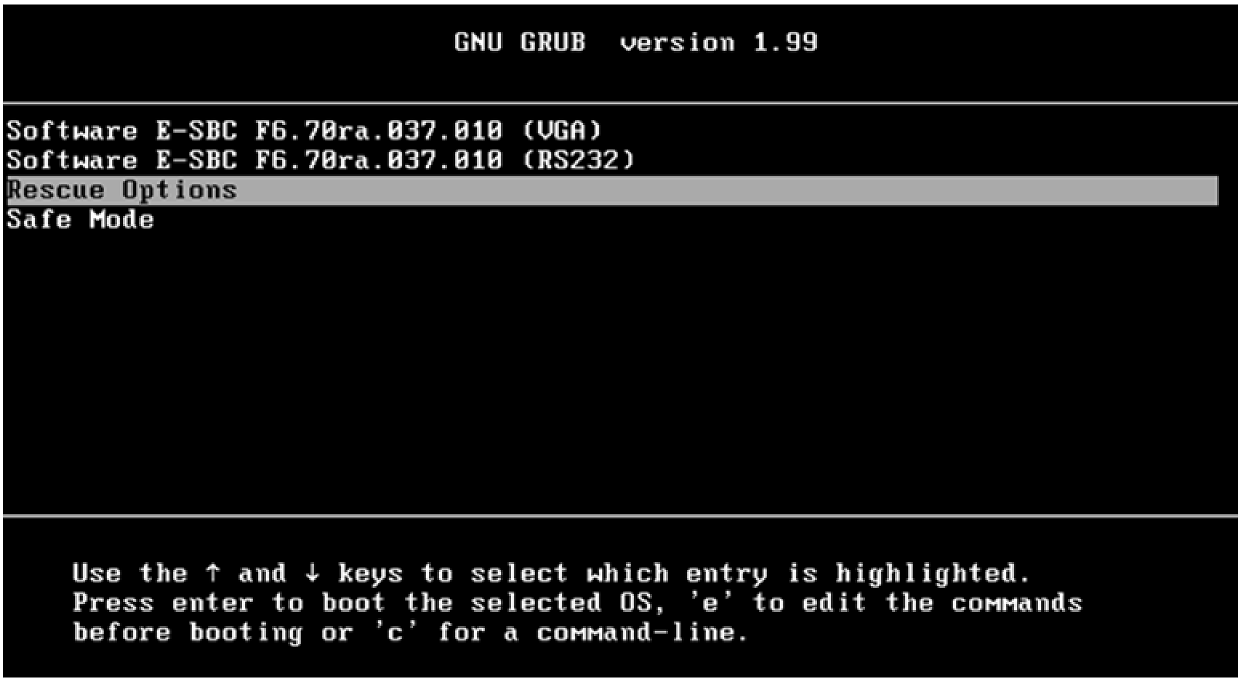
- Return to default snapshot
- Fix current installation
- Browse available system snapshots
- Return to factory snapshot (after install from CD)

25.5.5.1 *Returning to the Default Snapshot*

➢ To return to the default snapshot:

1. Reboot the server.
2. In the GRUB menu that's displayed for 5 seconds during the server start-up, press the Down ↓ key, select **Rescue option**, and then press Enter.

Figure 121: Main GRUB Menu



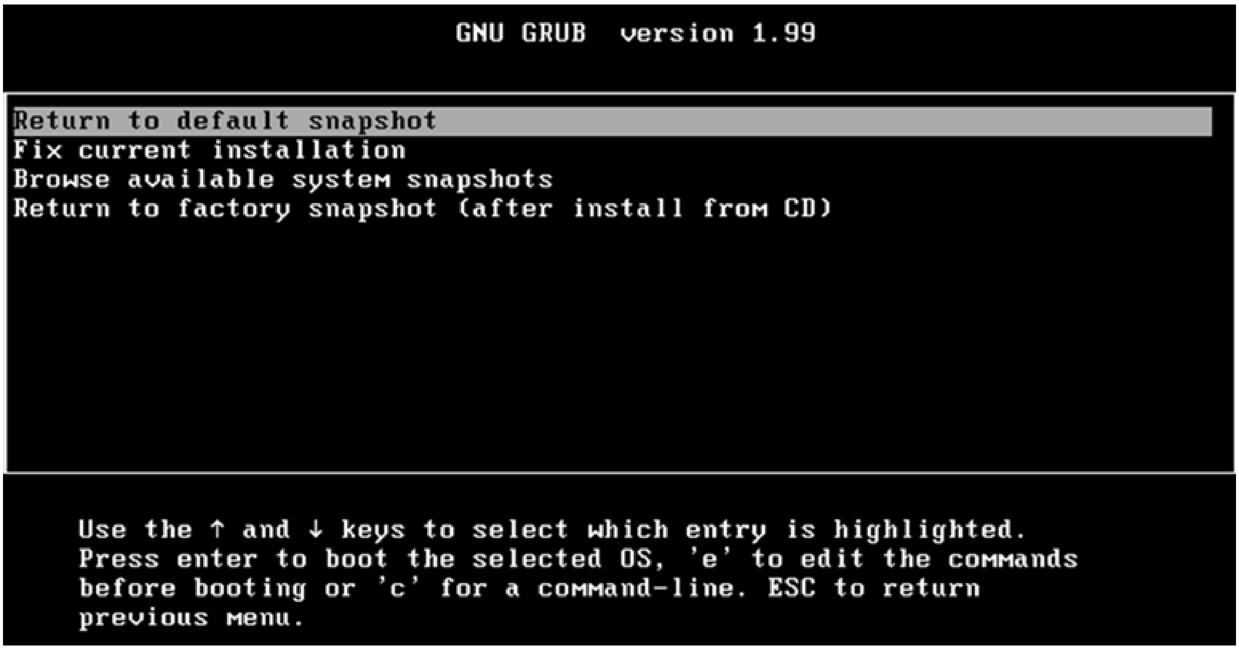
```
GNU GRUB  version 1.99

Software E-SBC F6.70ra.037.010 (VGA)
Software E-SBC F6.70ra.037.010 (RS232)
Rescue Options
Safe Mode

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the commands
    before booting or 'c' for a command-line.
```

3. In the Rescue Options menu, select **Return to default snapshot**, and then press Enter.

Figure 122: Rescue Options Menu



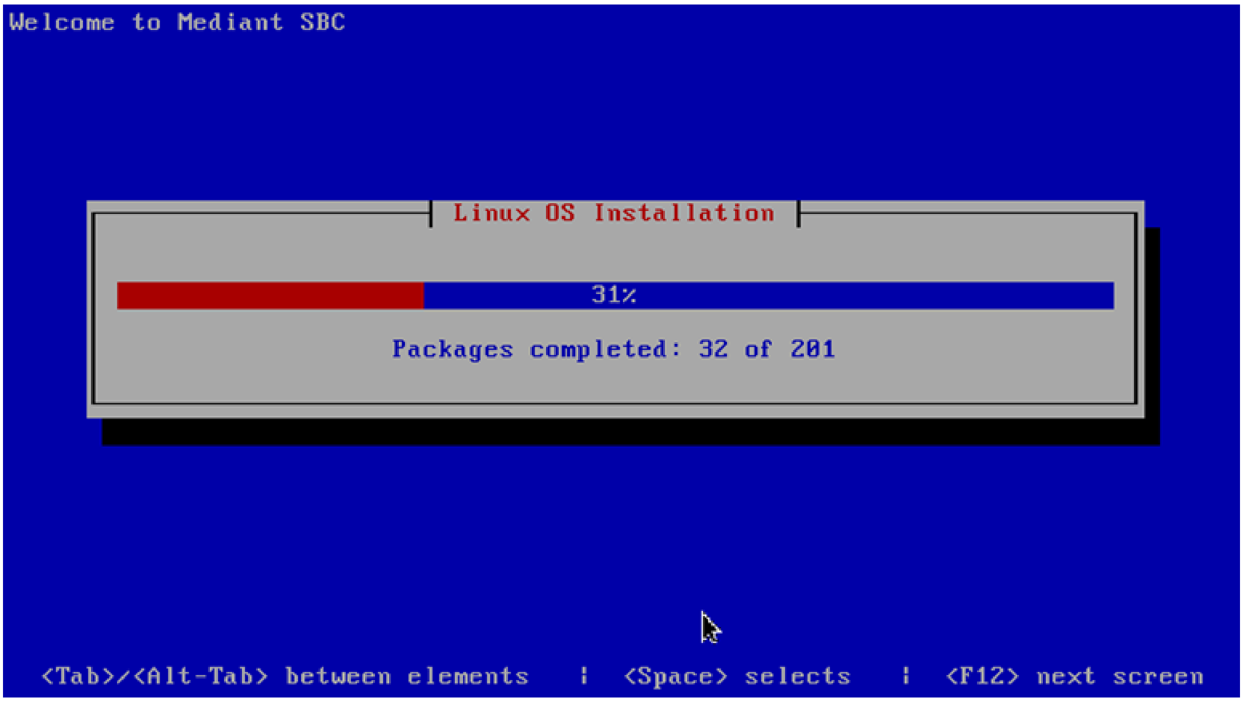
```
GNU GRUB  version 1.99

Return to default snapshot
Fix current installation
Browse available system snapshots
Return to factory snapshot (after install from CD)

    Use the ↑ and ↓ keys to select which entry is highlighted.
    Press enter to boot the selected OS, 'e' to edit the commands
    before booting or 'c' for a command-line. ESC to return
    previous menu.
```

The system returns to the default snapshot, restoring the software version and the full configuration. The process can take up to 10 minutes to complete.

Figure 123: System Returning to Snapshot State



```
Welcome to Mediant SBC

Linux OS Installation

31%

Packages completed: 32 of 201

<Tab>/<Alt-Tab> between elements   |  <Space> selects   |  <F12> next screen
```

25.5.5.2 Fixing the Current Installation

➢ To fix the current installation:

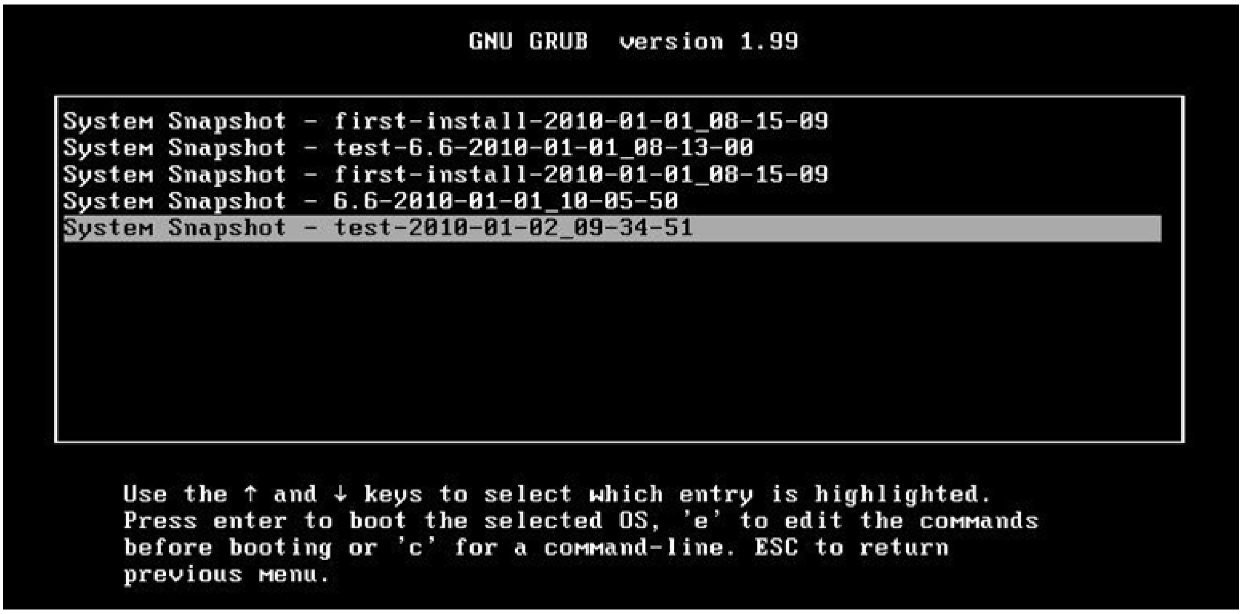
- In the GRUB menu, select **Fix current installation,** and then press Enter; the system is repaired while the currently installed software version and its configuration are preserved. The process can take up to 10 minutes to complete.

25.5.5.3 Returning to an Arbitrary Snapshot

➢ To return to an arbitrary (non-default) system snapshot:

1. In the GRUB menu, select **Browse available system snapshots**, and then press Enter; you're prompted to select a snapshot.

Figure 124: Selecting a Snapshot



```
GNU GRUB  version 1.99

System Snapshot - first-install-2010-01-01_08-15-09
System Snapshot - test-6.6-2010-01-01_08-13-00
System Snapshot - first-install-2010-01-01_08-15-09
System Snapshot - 6.6-2010-01-01_10-05-50
System Snapshot - test-2010-01-02_09-34-51

Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the commands
before booting or 'c' for a command-line. ESC to return
previous menu.
```

2. Select a snapshot, and then press Enter; the system returns to the selected snapshot, restores the software version and the full configuration. The process may take up to 10 minutes to complete.

25.5.5.4 Returning to a Factory Snapshot

➢ To return to a factory snapshot (after install from CD):

■ In the GRUB menu, select **Return to factory snapshot (after install from CD)**, and then press Enter; the system returns to the first snapshot automatically taken when initial installation from CD was performed. The process can take up to 10 minutes to complete.

25.5.6 Automatic Recovery

Thedevice activates Automatic Recovery when it encounters a severe software malfunction that prevents it from successfully booting for three subsequent attempts. Automatic Recovery returns the system to the 'default' snapshot and may take up to 10 minutes to complete.

Part VII

Maintenance

26 Basic Maintenance

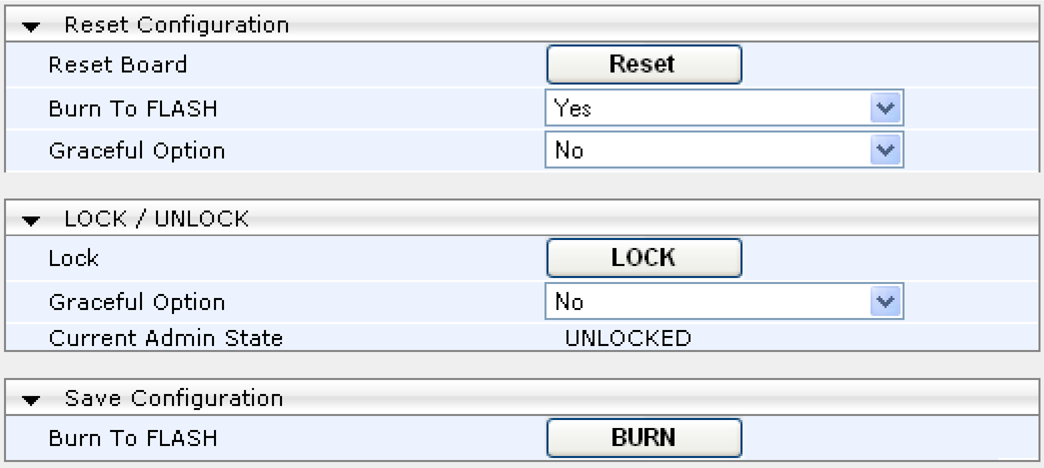
The Maintenance Actions page allows you to perform the following:

- Reset the device - see "Resetting the Device" on page 292
- Lock and unlock the device - see "Locking and Unlocking the Device" on page 294
- Save configuration to the device's flash memory - see "Saving Configuration" on page 295

➢ To access the Maintenance Actions page, do one of the following:

- On the toolbar, click the **Device Actions** button, and then from the drop-down menu, choose **Reset**.
- On the Navigation bar, click the **Maintenance** tab, and then in the Navigation tree, select the **Maintenance** menu and choose **Maintenance Actions**.

Figure 125: Maintenance Actions Page



▾ Reset Configuration

Reset Board: Reset

Burn To FLASH: Yes

Graceful Option: No

▾ LOCK / UNLOCK

Lock: LOCK

Graceful Option: No

Current Admin State: UNLOCKED

▾ Save Configuration

Burn To FLASH: BURN

26.1 Resetting the Device

The Maintenance Actions page allows you to remotely reset the device. In addition, before resetting the device, you can choose the following options:

- Save the device's current configuration to the device's flash memory (non-volatile).
- Perform a graceful shutdown, whereby device reset starts only after a user-defined time (i.e., timeout) or after no more active traffic exists (the earliest thereof).

Notes:

- Throughout the Web interface, parameters displayed with a lightning ⚡ symbol are not applied on-the-fly and require that you reset the device for them to take effect.
- When you modify parameters that require a device reset, once you click the **Submit** button in the relevant page, the toolbar displays "Reset" (see "Toolbar Description" on page 30) to indicate that a device reset is required.
- After you reset the device, the Web GUI is displayed in Basic view (see "Displaying Navigation Tree in Basic and Full View" on page 32).

➢ To reset the device:

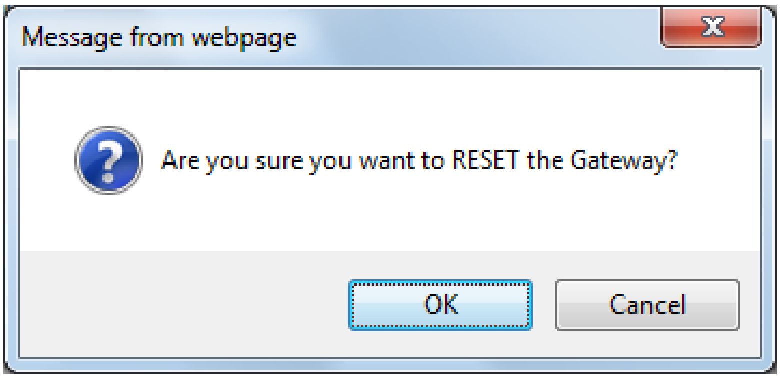
1. Open the Maintenance Actions page (see "Basic Maintenance" on page 292).
2. Under the 'Reset Configuration' group, from the 'Burn To FLASH' drop-down list, select one of the

following options:

- **Yes:** The device's current configuration is saved (*burned*) to the flash memory prior to reset (default).
- **No:** Resets the device without saving the current configuration to flash (discards all unsaved modifications).

3. Under the 'Reset Configuration' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - **Yes:** Reset starts only after the user-defined time in the 'Shutdown Timeout' field (see Step 4) expires or after no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - **No:** Reset starts regardless of traffic, and any existing traffic is terminated at once.
4. In the 'Shutdown Timeout' field (relevant only if the 'Graceful Option' in the previous step is set to **Yes**), enter the time after which the device resets. Note that if no traffic exists and the time has not yet expired, the device resets.
5. Click the **Reset** button; a confirmation message box appears, requesting you to confirm.

Figure 126: Reset Confirmation Message Box



Message from webpage

Are you sure you want to RESET the Gateway?

6. Click **OK** to confirm device reset; if the parameter 'Graceful Option' is set to **Yes** (in Step 3), the reset is delayed and a screen displaying the number of remaining calls and time is displayed. When the device begins to reset, a message appears notifying you of this.

26.2 Remotely Resetting Device using SIP NOTIFY

The device can be remotely reset upon the receipt of a SIP NOTIFY that includes an Event header set to 'check-sync;reboot=true', as shown in the example below:

```
NOTIFY sip:<user>@<dsthost> SIP/2.0
To: sip:<user>@<dsthost>
From: sip:sipsak@<srchost>
CSeq: 10 NOTIFY
Call-ID: 1234@<srchost>
Event: check-sync;reboot=true
```

➢ To enable remote reset upon receipt of SIP NOTIFY:

1. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).
2. Under the Misc Parameters group, set the 'SIP Remote Rest' parameter to **Enable**.
3. Click **Submit**.

Note: This SIP Event header value is proprietary to AudioCodes.

26.3 Locking and Unlocking the Device

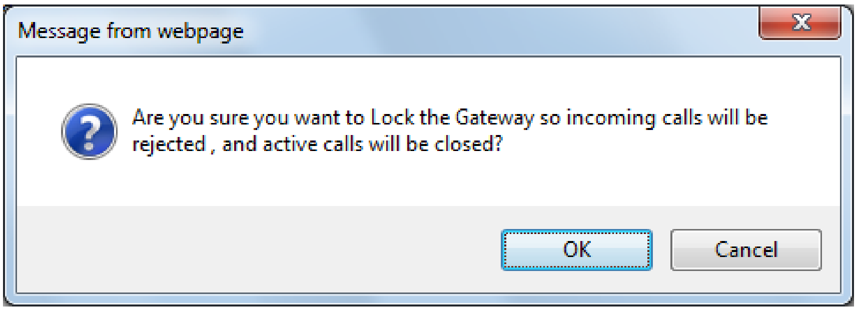
The Lock and Unlock option allows you to lock the device so that it doesn't accept any new calls and maintains only the current calls. This is useful when, for example, you are uploading new software files to the device and you don't want any traffic to interfere with the process.

➢ To lock the device:

1. Open the Maintenance Actions page (see "Basic Maintenance" on page 292).
2. Under the 'LOCK / UNLOCK' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - **Yes:** The device is 'locked' only after the user-defined time in the 'Lock Timeout' field (see Step 3) expires or no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - **No:** The device is 'locked' regardless of traffic. Any existing traffic is terminated immediately.

   Note: These options are only available if the current status of the device is in the Unlock state.
3. In the 'Lock Timeout' field (relevant only if the parameter 'Graceful Option' in the previous step is set to **Yes**), enter the time (in seconds) after which the device locks. Note that if no traffic exists and the time has not yet expired, the device locks.
4. Click the **LOCK** button; a confirmation message box appears requesting you to confirm device Lock.

Figure 127: Device Lock Confirmation Message Box



Message from webpage

Are you sure you want to Lock the Gateway so incoming calls will be rejected , and active calls will be closed?

5. Click **OK** to confirm device Lock; if 'Graceful Option' is set to **Yes**, the lock is delayed and a screen displaying the number of remaining calls and time is displayed. Otherwise, the lock process begins immediately. The Current Admin State' field displays the current state - "LOCKED" or "UNLOCKED".

➢ To unlock the device:

1. Open the Maintenance Actions page (see "Maintenance Actions" on page 292).
2. Under the 'LOCK / UNLOCK' group, click the **UNLOCK** button. Unlock starts immediately and the device accepts new incoming calls.

Note: The Home page's General Information pane displays whether the device is locked or unlocked (see "Viewing the Home Page" on page 41).

26.4 Saving Configuration

The Maintenance Actions page allows you to save (*burn*) the current parameter configuration (including loaded auxiliary files) to the device's *non-volatile* memory (i.e., flash). The parameter modifications that you make throughout the Web interface's pages are temporarily saved (to the *volatile* memory - RAM) when you click the **Submit** button on these pages. Parameter settings that are saved only to the device's RAM revert to their previous settings after a hardware/software reset (or power failure). Therefore, to ensure that your configuration changes are retained, you must save them to the device's flash memory using the burn option described below.

➢ To save the changes to the non-volatile flash memory:

1. Open the Maintenance Actions page (see "Basic Maintenance" on page 292).
2. Under the 'Save Configuration' group, click the **BURN** button; a confirmation message appears when the configuration successfully saves.

Notes:

- Saving configuration to the *non-volatile* memory may disrupt current traffic on the device. To avoid this, disable all new traffic before saving, by performing a graceful lock (see "Locking and Unlocking the Device" on page 294).
- Throughout the Web interface, parameters displayed with the lightning ⚡ symbol are not applied on-the-fly and require that you reset the device for them to take effect (see "Resetting the Device" on page 292).
- The Home page's General Information pane displays whether the device is currently "burning" the configuration (see "Viewing the Home Page" on page 41).

27 High Availability Maintenance

This section describes various maintenance procedures for the High Availability mode.

27.1 Initiating an HA Switchover

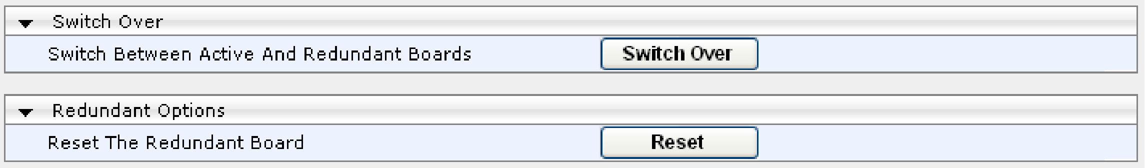
You can initiate a switchover from the Active to Redundant SBC.

> **Note:** When performing an HA switchover, the HA mode becomes temporarily unavailable.

➢ To perform a switch-over:

1. Open the High Availability Maintenance page:
   - Navigation menu tree: **Maintenance** tab > **Maintenance** menu > **High Availability Maintenance**
   - Toolbar: Click the **Device Actions** button, and then choose **Switch Over**

Figure 128: High Availability Maintenance Page



▾ Switch Over

Switch Between Active And Redundant Boards [Switch Over]

▾ Redundant Options

Reset The Redundant Board [Reset]

2. Under the 'Switch Over' group, click **Switch Over**; a confirmation box appears requesting you to confirm.
3. Click **OK**.

27.2 Resetting the Redundant Unit

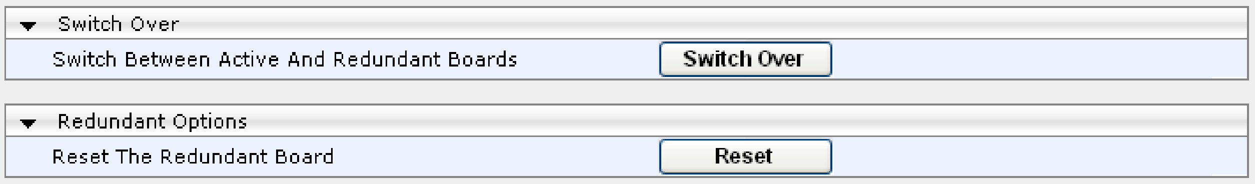
You can reset the Redundant SBC, if necessary.

> **Note:** When resetting the Redundant SBC, the HA mode becomes temporarily unavailable.

➢ To reset the Redundant SBC:

1. Open the High Availability Maintenance page:
   - Navigation menu tree: **Maintenance** tab > **Maintenance** menu > **High Availability Maintenance**
   - Toolbar: Click the **Device Actions** button, and then choose **Reset Redundant**

Figure 129: High Availability Maintenance Page



▾ Switch Over

Switch Between Active And Redundant Boards [Switch Over]

▾ Redundant Options

Reset The Redundant Board [Reset]

2. Under the 'Redundant Options' group, click **Reset**; a confirmation box appears requesting you to confirm.
3. Click **OK**.

28 Disconnecting Active Calls

You can forcibly disconnect all active (established) calls or disconnect specific calls based on their Session ID. This is done in the CLI using the following commands (from basic command mode):

- Disconnects all active calls:

```
# clear voip calls
```

- Disconnects active calls belonging to a specified Session ID:

```
# clear voip calls <Session ID>
```

Reader's Notes

29 Software Upgrade

This chapter describes various software update procedures.

29.1 Loading Auxiliary Files

Various Auxiliary files can be installed on the device. These Auxiliary files provide the device with additional configuration settings. The table below lists the different types of Auxiliary files:

Table 77: Auxiliary Files

File	Description
INI	Configures the device. The Web interface enables practically full device provisioning. However, some features may only be configured by ini file or you may wish to configure your device using the ini file. For more information on the ini file, see "INI File-Based Management" on page 69.
Call Progress Tones	Region-specific, telephone exchange-dependent file that contains the Call Progress Tones (CPT) levels and frequencies for the device. The default CPT file is U.S.A. For more information, see "Call Progress Tones File" on page 300.
Prerecorded Tones	The Prerecorded Tones (PRT) file enhances the device's capabilities of playing a wide range of telephone exchange tones that cannot be defined in the CPT file. For more information, see "Prerecorded Tones File" on page 302.
Dial Plan	Provides dialing plans, for example, for obtaining the destination IP address for outbound IP routing. For more information, see "Dial Plan File" on page 303.
User Info	The User Information file maps PBX extensions to IP numbers. This file can be used to represent PBX extensions as IP phones in the global 'IP world'. For more information, see "User Information File" on page 305.

The Auxiliary files can be loaded to the device using one of the following methods:

- Web interface.
- TFTP: This is done by specifying the name of the Auxiliary file in an *ini* file (see Auxiliary and Configuration Files Parameters) and then loading the *ini* file to the device. The Auxiliary files listed in the *ini* file are then automatically loaded through TFTP during device startup. If the *ini* file does not contain a specific auxiliary file type, the device uses the last auxiliary file of that type that was stored on its non-volatile memory.

Notes:

- You can schedule automatic loading of updated auxiliary files using HTTP/HTTPS. For more information on automatic updates, see Automatic Update Mechanism.
- When loading an *ini* file using this Web page, parameters that are excluded from the loaded *ini* file retain their current settings (*incremental*).
- Saving an auxiliary file to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device by performing a graceful lock as described in "Locking and Unlocking the Device" on page 294.
- For deleting auxiliary files, see "Viewing Device Information" on page 331.

The procedure below describes how to load Auxiliary files using the Web interface.

➢ To load auxiliary files to the device using the Web interface:

1. Open the Load Auxiliary Files page (**Maintenance** tab > **Software Update** menu > **Load Auxiliary Files**).

Note: The appearance of certain file load fields depends on the installed Software License Key.

2. Click the **Browse** button corresponding to the file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name and path of the file appear in the field next to the **Browse** button.
3. Click the **Load File** button corresponding to the file you want to load.
4. Repeat steps 2 through 3 for each file you want to load.
5. Save the loaded auxiliary files to flash memory, see "Saving Configuration" on page 295 and reset the device (if you have loaded a Call Progress Tones file), see "Resetting the Device" on page 292.

29.1.1 Call Progress Tones File

The Call Progress Tones (CPT) auxiliary file includes the definitions of the Call Progress Tones (levels and frequencies) that are detected / generated by the device.

You can use one of the supplied auxiliary files (.dat file format) or create your own file. To create your own file, it's recommended to modify the supplied *usa_tone.ini* file (in any standard text editor) to suit your specific requirements and then convert the modified *ini* file into binary format, using AudioCodes DConvert utility. For a description on converting a CPT *ini* file into a binary *dat* file, refer to the *DConvert Utility User's Guide*.

Note: Only the *dat* file format can be loaded to the device.

You can create up to 32 different Call Progress Tones, each with frequency and format attributes. The frequency attribute can be single or dual-frequency (in the range of 300 to 1980 Hz) or an Amplitude Modulated (AM). Up to 64 different frequencies are supported. Only eight AM tones, in the range of 1 to 128 kHz, can be configured (the detection range is limited to 1 to 50 kHz). Note that when a tone is composed of a single frequency, the second frequency field must be set to zero.

The format attribute can be one of the following:

- **Continuous:** A steady non-interrupted sound (e.g., a dial tone). Only the 'First Signal On time' should be specified. All other on and off periods must be set to zero. In this case, the parameter specifies the detection period. For example, if it equals 300, the tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.
- **Cadence:** A repeating sequence of on and off sounds. Up to four different sets of on/off periods can be specified.
- **Burst:** A single sound followed by silence. Only the 'First Signal On time' and 'First Signal Off time' should be specified. All other on and off periods must be set to zero. The burst tone is detected after the off time is completed.

You can specify several tones of the same type. These additional tones are used only for tone detection. Generation of a specific tone conforms to the first definition of the specific tone. For example, you can define an additional dial tone by appending the second dial tone's definition lines to the first tone definition in the *ini* file. The device reports dial tone detection if either of the two tones is detected.

The Call Progress Tones section of the *ini* file comprises the following segments:

- **[NUMBER OF CALL PROGRESS TONES]:** Contains the following key: 'Number of Call Progress Tones' defining the number of Call Progress Tones that are defined in the file.
- **[CALL PROGRESS TONE #X]:** containing the Xth tone definition, starting from 0 and not exceeding the number of Call Progress Tones less 1 defined in the first section (e.g., if 10 tones, then it is 0 to 9), using the following keys:
  - **Tone Type:** Call Progress Tone types:
    - **[1]** Dial Tone
    - **[2]** Ringback Tone
    - **[3]** Busy Tone
    - **[4]** Congestion Tone
    - **[6]** Warning Tone
    - **[7]** Reorder Tone
    - **[17]** Call Waiting Ringback Tone - heard by the calling party
    - **[18]** Comfort Tone
    - **[23]** Hold Tone
    - **[46]** Beep Tone
  - **Tone Modulation Type:** Amplitude Modulated (1) or regular (0)
  - **Tone Form:** The tone's format can be one of the following:
    - Continuous (1)
    - Cadence (2)
    - Burst (3)
  - **Low Freq [Hz]:** Frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to AM tones.
  - **High Freq [Hz:** Frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).
  - **Low Freq Level [-dBm]:** Generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).
  - **High Freq Level:** Generation level of 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).
  - **First Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For continuous tones, this parameter defines the detection period. For burst tones, it defines the tone's duration.
  - **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, this parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, this parameter is ignored.
  - **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
  - **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
  - **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
  - **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
  - **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
  - **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.

- **Carrier Freq [Hz]:** Frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** Frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).
- **Signal Level [-dBm]:** Level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** Amplitude modulation factor (valid range from 1 to 50). Recommended values from 10 to 25.

Notes:

- When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise, the continuous tone is detected instead of the cadence tone.
- The tones frequency must differ by at least 40 Hz between defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
[NUMBER OF CALL PROGRESS TONES]
Number of Call Progress Tones=1
#Dial Tone
[CALL PROGRESS TONE #0]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is required)
First Signal On Time [10msec]=300; the dial tone is detected after 3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

29.1.2 Prerecorded Tones File

The CPT file mechanism has several limitations such as a limited number of predefined tones and a limited number of frequency integrations in one tone. To overcome these limitations and provide tone generation capability that is more flexible, the Prerecorded Tones (PRT) file can be used. If a specific prerecorded tone exists in the PRT file, it takes precedence over the same tone that exists in the CPT file and is played instead of it.

Note: The PRT file is used only for generation of tones; detection of tones is performed according to the CPT file.

The PRT is a .dat file containing a set of prerecorded tones that can be played by the device. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single PRT file on the device's flash memory. The prerecorded tones are prepared offline using standard recording utilities (such as Cool Edit) and combined into a single file, using AudioCodes DConvert utility (refer to the document, *DConvert Utility User's Guide* for more information).

The raw data files must be recorded with the following characteristics:

- Coders: G.711 A-law or G.711 µ-law
- Rate: 8 kHz
- Resolution: 8-bit

- Channels: mono

Once created, the PRT file must then be loaded to the device, using the Web interface (see "Loading Auxiliary Files" on page 299).

The prerecorded tones are played repeatedly. This allows you to record only part of the tone and then play the tone for the full duration. For example, if a tone has a cadence of 2 seconds on and 4 seconds off, the recorded file should contain only these 6 seconds. The device repeatedly plays this cadence for the configured duration. Similarly, a continuous tone can be played by repeating only part of it.

29.1.3 Dial Plan File

The Dial Plan file can be used for various digit mapping features, as described in this section.

29.1.3.1 Creating a Dial Plan File

The Dial Plan file is a text-based file that can contain up to 8 Dial Plans (Dial Plan indices) and up to 8,000 rules (lines). The general syntax rules for the Dial Plan file are as follows (syntax specific to the feature is described in the respective section):

- Each Dial Plan index must begin with a Dial Plan name enclosed in square brackets "[...]" on a new line.
- Each line under the Dial Plan index defines a rule.
- Empty lines are ignored.
- Lines beginning with a semicolon ";" are ignored. The semicolon can be used for comments.

Creating a Dial Plan file is similar for all Dial Plan features. The main difference is the syntax used in the Dial Plan file and the method for selecting the Dial Plan index.

➢ To create a Dial Plan file:

1. Create a new file using a text-based editor (such as Notepad) and configure your Dial Plans, as required.
2. Save the file with the *ini* file extension name (e.g., mydialplanfile.ini).
3. Convert the *ini* file to a *dat* binary file, using AudioCodes DConvert utility. For more information, refer to *DConvert Utility User's Guide*.
4. Load the converted file to the device, as described in "Loading Auxiliary Files" on page 299.
5. Select the Dial Plan index that you want to use. This depends on the feature and is described in the respective section.

29.1.3.2 Dial Plan Prefix Tags for Routing

29.1.3.2.1 Dial Plan Prefix Tags for SBC IP-to-IP Routing

For deployments requiring many SBC IP-to-IP routing rules that exceed the maximum number of rules that can be configured in the IP-to-IP Routing table, you can employ user-defined string labels (tags) to represent the many different prefix calling (source) and called (destination) numbers. The prefix tags are used in the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 255) as source and destination URI user parts matching characteristics for the routing rule. Prefix tags are typically implemented when you have calls of many different called or calling numbers that need to be routed to the same destination. Thus, instead of configuring a routing rule for each prefix number, you need to configure only one routing rule using the prefix tag.

For example, this feature is useful in deployments that need to handle hundreds of call routing scenarios such as for a large geographical area (a state in the US). Such an area could consist of hundreds of local area codes as well as codes for international calls. The local calls and international calls would need to be routed to different SIP trunks. Thus, instead of configuring many routing rules for each call destination type, you can simply configure two routing rules, one with a unique prefix tag representing the different local area codes and the other with a prefix tag representing international calls.

> ⚠ **Note:** When using prefix tags, you need to configure manipulation rules to remove the tags before the device sends the calls to their destinations.

You configure prefix tags in the Dial Plan file, using the following syntax:

```
[ PLAN<index> ]
<prefix number>,0,<prefix tag>
```

where:

- *Index* is the Dial Plan index
- *prefix number* is the called or calling number prefix (ranges can be defined in brackets)
- *prefix tag* is the user-defined prefix tag of up to nine characters, representing the prefix number

Each prefix tag type - called or calling - must be configured in a dedicated Dial Plan index number. For example, Dial Plan 1 can be for called prefix tags and Dial Plan 2 for calling prefix tags.

The example Dial Plan file below defines the prefix tags "LOCL"and "INTL" to represent different called number prefixes for local and long distance calls:

```
[ PLAN1 ]
42520[3-5],0,LOCL
425207,0,LOCL
42529,0,LOCL
425200,0,INTL
425100,0,INTL
....
```

> ⚠ **Note:** Called and calling prefix tags can be used in the same routing rule.

The procedure below describes how to configure IP-to-IP routing using prefix tags.

➢ To configure IP-to-IP routing using prefix tags:

1. Configure a Dial Plan file with prefix tags, and then load the file to the device.
2. Add the prefix tags to the numbers of specific incoming calls using Inbound IP-to-IP Manipulation rules:
   a. Open the IP to IP Inbound Manipulation table (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP-to-IP Inbound**), and then click **Add**.
   b. Click the **Rule** tab, and then configure matching characteristics for the incoming call (e.g., set 'Source IP Group ID' to "1").
   c. From the 'Manipulated URI' drop-down list, select **Source** to add the tag to the calling URI user part, or **Destination** to add the tag to the called URI user part.
   d. Click the **Action** tab, and then enter the Dial Plan index for which you configured your prefix tag, in the 'Prefix to Add' or 'Suffix to Add' fields, using the following syntax: $DialPlan<x>, where *x* is the Dial Plan index (0 to 7). For example, if the called number is 4252000555, the device manipulates it to LOCL4252000555.
3. Add an SBC IP-to-IP routing rule using the prefix tag to represent the different source or destination URI user parts:
   a. Open the IP-to-IP Routing table (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **IP-to-IP Routing Table**), and then click **Add**.
   b. Click the **Rule** tab, and then enter the prefix tag in the 'Source Username Prefix' or 'Destination Username Prefix' fields (e.g., "LOCL", without the quotes).

c. Continue configuring the rule as required.

4. Configure a manipulation rule to remove the prefix tags before the device sends the message to the destination:
   a. Open the IP to IP Outbound Manipulation table (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP-to-IP Outbound**), and then click **Add**.
   b. Click the **Rule** tab, and then configure matching characteristics for the incoming call (e.g., set 'Source IP Group ID' to "1"), including calls with the prefix tag (in the 'Source Username Prefix' or 'Destination Username Prefix' fields, enter the prefix tag to remove).
   c. Click the **Action** tab, and then in the 'Remove from Left' or 'Remove from Right' fields (depending on whether you added the tag at the beginning or end of the URI user part, respectively), enter the number of characters making up the tag.

29.1.3.3 Obtaining IP Destination from Dial Plan File

You can use a Dial Plan index listed in a loaded Dial Plan file for determining the IP destination of SBC calls. This enables the mapping of called numbers to IP addresses (in dotted-decimal notation) or FQDNs (up to 15 characters).

➢ To configure routing to an IP destination based on Dial Plan:

1. Create the Dial Plan file. The syntax of the Dial Plan index for this feature is as follows:

```
<destination / called prefix number>,0,<IP destination>
```

Note: The second parameter "0" is not used and ignored.

An example of a configured Dial Plan (# 6) in the Dial Plan file is shown below:

```
[ PLAN6 ]
200,0,10.33.8.52     ; called prefix 200 is routed to 10.33.8.52
201,0,10.33.8.52
300,0,itsp.com       ; called prefix 300 is routed to itsp.com
```

2. Convert the file to a loadable file and then load it to the device (see "Creating a Dial Plan File" on page 303).
3. Assign the Dial Plan index to the required routing rule:
   - SBC Calls: In the SBC IP-to-IP Routing table, do the following:
     a. Set the 'Destination Type' field to Dial Plan.
     b. In the 'Destination Address' field, enter the required Dial Plan index, where "0" denotes [PLAN1] in the Dial Plan file, "1" denotes [PLAN2], and so on.

29.1.4 User Information File

This section describes the User Info table and how to configure the table.

29.1.4.1 Enabling the User Info Table

Before you can use the User Info table, you need to enable the User Info functionality as described in the procedure below.

➢ To enable the User Info table:

1. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).
2. Set the 'Enable User-Information Usage' parameter to **Enable**.
3. Save this setting to the device with a reset for the setting to take effect.

29.1.4.2 User Information File for SBC User Database

The device can use the SBC User Info table for the following:

- Registering each user to an external registrar server.
- Authenticating (for any SIP request and as a client) each user if challenged by an external server.
- Authenticating as a server incoming user requests (for SBC security).

If the device registers on behalf of users and the users do not perform registration, any SIP request destined to the user is routed to the Proxy Set associated with the user's IP Group.

The SBC User Info table can be configured using any of the following methods:

- Web interface - see "Configuring SBC User Info Table in Web Interface" on page 306
- CLI - see Configuring SBC User Info Table in CLI on page 307
- Loadable User Info file - see "Configuring SBC User Info Table in Loadable Text File" on page 308

29.1.4.2.1 Configuring SBC User Info Table in Web Interface

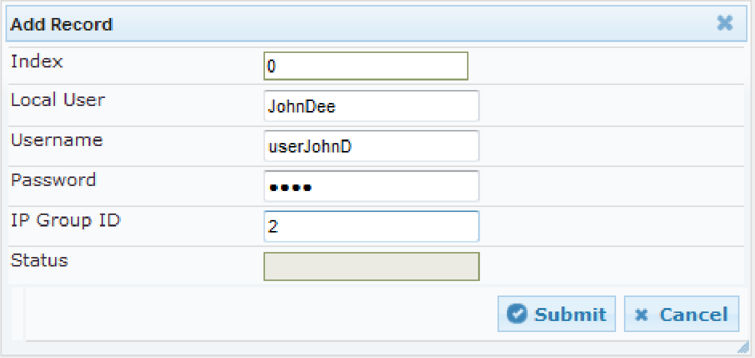
The procedure below describes how to configure the SBC User Info table in the Web interface.

> **Note:** If any User Info file is loaded to the device, all previously configured entries are removed from the table in the Web interface and replaced with the entries from the loaded User Info file.

➢ To configure the SBC User Info table in the Web interface:

1. Open the SBC User Info Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **User Information** > **SBC User Info Table**).
2. Click **Add**; the following dialog box appears:

Figure 130: SBC User Info Table Page



Add Record

Index	0
Local User	JohnDee
Username	userJohnD
Password	••••
IP Group ID	2
Status	

3. Configure the SBC User Info table parameters according to the table below.
4. Click **Submit**.
5. To save the changes to flash memory, see "Saving Configuration" **on page 295.**

To register a user, select the user's table entry, and then from the **Action** button's drop-down list , choose **Register**. To un-register a user, select the user, and then from the **Action** button's drop-down list , choose **Un-Register**.

Table 78: SBC User Info Table Parameter Descriptions

Parameter	Description
Index **[SBCUserInfoTable_Index]**	Defines an index for the new table record.

Parameter	Description
Local User **[SBCUserInfoTable_LocalUser]**	Defines the user and is used as the Request-URI user part for the AOR in the database. The valid value is a string of up to 10 characters.
Username **[SBCUserInfoTable_Username]**	Defines the username for registering the user when authentication is necessary. The valid value is a string of up to 40 characters.
Password **[SBCUserInfoTable_Password]**	Defines the password for registering the user when authentication is necessary. The valid value is a string of up to 20 characters.
IP Group ID **[SBCUserInfoTable_IPGroupID]**	Defines the IP Group ID to which the user belongs and is used as the Request-URI source host part for the AOR in the database.
Status **[SBCUserInfoTable_Status]**	(Read-only field) Displays the status of the user - "Registered" or "Not Registered".

29.1.4.2.2 Configuring SBC User Info Table in CLI

The SBC User Info table can be configured in the CLI using the following commands:

- To add and/or modify a user (example):

```
# configure voip
(config-voip)# sip-definition proxy-and-registration
(sip-def-proxy-and-reg)# user-info sbc-user-info <index, e.g., 1>
(sbc-user-info-1)# username JohnDee
(sbc-user-info-1)# <activate | exit>
```

- To delete a specific user, use the `no` command:

```
(sip-def-proxy-and-reg)# no user-info sbc-user-info <index, e.g., 1>
```

- To view all table entries:

```
(sip-def-proxy-and-reg)# user-info sbc-user-info display
---- sbc-user-info-0 ----
  local-user (JohnDee)
  username (userJohn)
  password (s3fn+fn=)
  ip-group-id (1)
  status (not-resgistered)
---- sbc-user-info-1 ----
 local-user (SuePark)
  username (userSue)
  password (t6sn+un=)
  ip-group-id (1)
  status (not-resgistered)
```

- To view a specific entry (example):

```
(sip-def-proxy-and-reg)# user-info sbc-user-info <index, e.g., 0>
(sbc-user-info-0)# display
  local-user (JohnDee)
  username (userJohn)
  password (s3fn+fn=)
  ip-group-id (1)
  status (not-resgistered)
```

- To search a user by local-user:

```
(sip-def-proxy-and-reg)# user-info find <local-user, e.g., JohnDoe>
JohnDee: Found at index 0 in SBC user info table, not registered
```

29.1.4.2.3 Configuring SBC User Info Table in Loadable Text File

The SBC User Info table can be configured as a User Info file using a text-based file (*.txt). This file can be created using any text-based program such as Notepad.

You can load the User Info file using any of the following methods:

- Web interface - see "Loading Auxiliary Files" on page 299
- *ini* file, using the UserInfoFileName parameter - see "Auxiliary and Configuration File Name Parameters" on page 388
- Automatic Update mechanism, using the UserInfoFileURL parameter - see Automatic Update Mechanism

To add SBC users to the SBC User Info file, use the following syntax:

```
[ SBC ]
FORMAT LocalUser,UserName,Password,IPGroupID
john,john user,john pass,2
sue,sue_user,sue_pass,1
```

where:

- *[SBC]* indicates that this part of the file is the SBC User Info table
- *LocalUser* is the user and is used as the Request-URI user part for the AOR in the database
- *UserName* is the user's authentication username
- *Password* is the user's authentication password
- *IPGroupID* is the IP Group ID to which the user belongs and is used as the Request-URI source host part for the AOR in the database

Note: To modify the SBC User Info table using a User Info file, you need to load to the device a new User Info file containing your modifications.

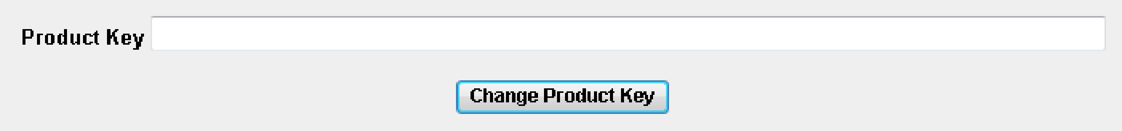
29.2 Configuring the Product Key

The Product Key is used to identify a specific purchase of your device installation for the purpose of subsequent communication with AudioCodes, for example, for support and software upgrades. The Product Key is provided at the time the product is purchased (together with the Installation Disk or download link) and should be entered into the Web interface as described below.

➢ To enter the Product Key:

1. Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** menu > **Software Upgrade Key**).

Figure 131: Product Key on Software Upgrade Key Status Page



2. In the 'Product Key' field, enter the Product Key.
3. Click the **Change Product Key** button.

You can view the Product Key on the Device Information page (see "Viewing Device Information" on page 331).

29.3 Software License Key

The device is shipped with a pre-installed Software License Key, which determines the device's supported features, capabilities, and available resources. You can upgrade or change your device's supported features by purchasing and installing a new Software License Key to match your requirements.

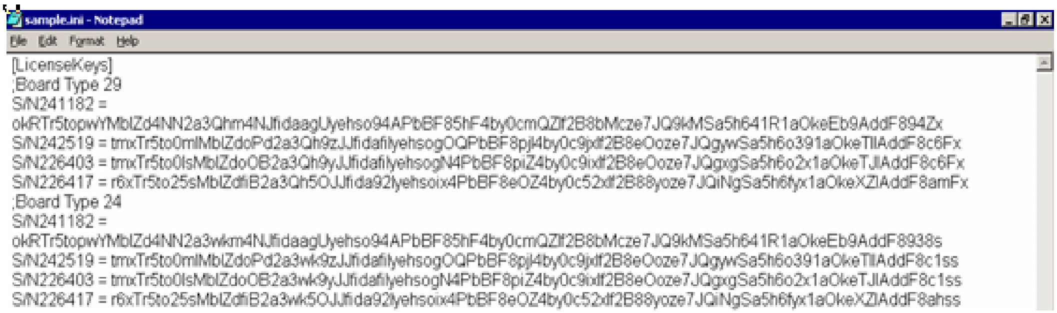
Notes:

- The device is shipped by default with a pre-installed Software License Key that enables only one call session. Once you have installed the Mediant Software E-SBC, you need to load the Software License Key file supplied in the package, to enable the call capacity and features that you ordered. If you did not receive this Software License Key file with your installation disk, contact your AudioCodes sales representative to obtain it, as described in Obtaining the Software License Key File on page 309.
- For the High Availability (HA) system, the Software License Key includes the HA feature and is installed on both devices - active and redundant. If the redundant device's Software License Key is missing or invalid, the system is moved to mismatch configuration mode (alerted by SNMP).
- The availability of certain Web pages depends on the installed Software License Key.

29.3.1 Obtaining the Software License Key File

Before you can install a new Software License Key, you need to obtain a Software License Key file for your device with the required features from your AudioCodes representative. The Software License Key is an encrypted key in string format that is associated with the device's serial number ("S/N") and supplied in a text-based file. If you need a Software License Key for more than one device, the Software License Key file can include multiple Software License Keys (see figure below). In such cases, each Software License Key in the file is associated with a unique serial number identifying the specific device. When loading such a Software License Key file, the device installs only the Software License Key that is associated with its serial number.

Figure 132: Software License Key File with Multiple S/N Lines



```
[LicenseKeys]
;Board Type 29
S/N241182 =
okRTr5topwYMblZd4NN2a3Qhm4NJfidaagUyehso94APbBF85hF4by0cmQZlf2B8bMcze7JQ9kMSa5h641R1aOkeEb9AddF894Zx
S/N242519 = tmxTr5to0mlMblZdoPd2a3Qh9zJJfidafilyehsogOQPbBF8pjl4by0c9jxlf2B8eOoze7JQgywSa5h6o391aOkeTllAddF8c6Fx
S/N226403 = tmxTr5to0lsMblZdoOB2a3Qh9yJJfidafilyehsogN4PbBF8piZ4by0c9ixlf2B8eOoze7JQgxgSa5h6o2x1aOkeTJlAddF8c6Fx
S/N226417 = r6xTr5to25sMblZdfiB2a3Qh5OJJfida92lyehsoix4PbBF8eOZ4by0c52xlf2B88yoze7JQiNgSa5h6fyx1aOkeXZlAddF8amFx
;Board Type 24
S/N241182 =
okRTr5topwYMblZd4NN2a3wkm4NJfidaagUyehso94APbBF85hF4by0cmQZlf2B8bMcze7JQ9kMSa5h641R1aOkeEb9AddF8938s
S/N242519 = tmxTr5to0mlMblZdoPd2a3wk9zJJfidafilyehsogOQPbBF8pjl4by0c9jxlf2B8eOoze7JQgywSa5h6o391aOkeTllAddF8c1ss
S/N226403 = tmxTr5to0lsMblZdoOB2a3wk9yJJfidafilyehsogN4PbBF8piZ4by0c9ixlf2B8eOoze7JQgxgSa5h6o2x1aOkeTJlAddF8c1ss
S/N226417 = r6xTr5to25sMblZdfiB2a3wk5OJJfida92lyehsoix4PbBF8eOZ4by0c52xlf2B88yoze7JQiNgSa5h6fyx1aOkeXZlAddF8ahss
```

➢ To obtain a Software License Key:

1. Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**) and make a note of the device's serial number and product key:
   - 'MAC Address' field displays the MAC address.
   - 'Serial Number' field displays the serial number.
   - 'Product Key' field displays the product key.
2. If you need a Software License Key for more than one device, repeat Step 1 for each device.
3. Send the serial number and product key to your AudioCodes representative when requesting the required Software License Key.
4. When you receive the new Software License Key file, check the file as follows:
   a. Open the file with any text-based program such as Notepad.
   b. Verify that the first line displays "[LicenseKeys]".
   c. Verify that the file contains one or more lines in the following format:

```
"S/N<serial number> = <Software License Key string>"
```

   For example: "S/N370604 = jCx6r5tovCIKaBBbhPtT53Yj..."

   d. Verify that the "S/N" value reflects the serial number of your device. If you have multiple Software License Keys, ensure that each "S/N" value corresponds to a device.

Warning: Do not modify the contents of the Software License Key file.

5. Install the Software License Key on the device, as described in "Installing the Software License Key" on page 310.

29.3.2 Installing the Software License Key

Once you have received your Software License Key file from your AudioCodes representative, you can install it on the device using one of the following management tools:

- Web interface - see "Installing Software License Key using Web Interface" on page 311
- CLI - see Installing Software License Key using CLI on page 312

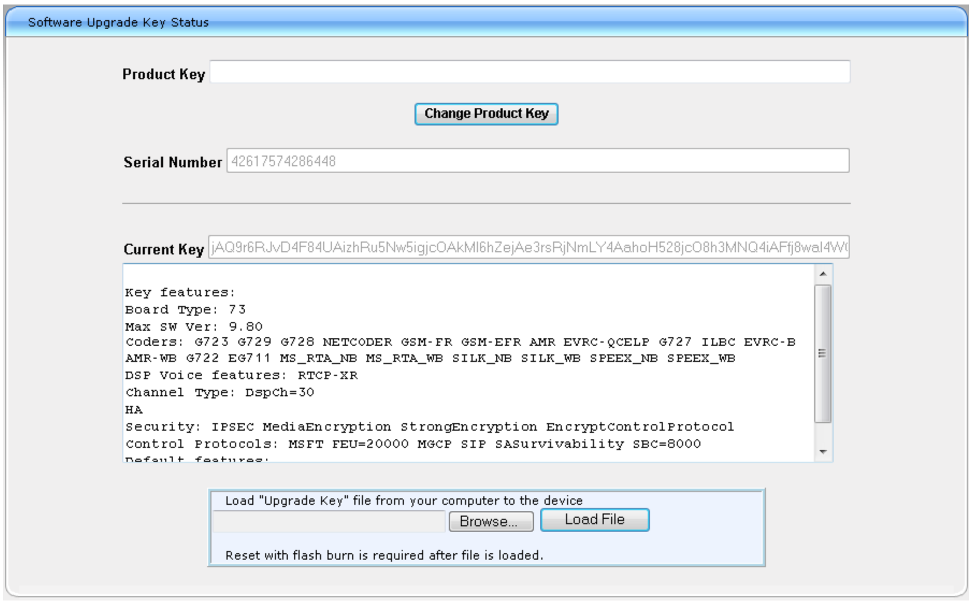
Note: When you install a new Software License Key, it is loaded to the device's non-volatile flash memory and overwrites the previously installed Software License Key.

29.3.2.1 Installing Software License Key using Web Interface

The procedure below describes how to install the Software License Key in the Web interface.

➢ To install the Software License Key in the Web interface:

1. Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** menu > **Software Upgrade Key**).



Product Key

Serial Number 42617574286448

Current Key jAQ9r6RJvD4F84UAizhRu5Nw5igjcOAkMI6hZejAe3rsRjNmLY4AahoH528jcO8h3MNQ4iAFfj8waI4W

```
Key features:
Board Type: 73
Max SW Ver: 9.80
Coders: G723 G729 G728 NETCODER GSM-FR GSM-EFR AMR EVRC-QCELP G727 ILBC EVRC-B
AMR-WB G722 EG711 MS_RTA_NB MS_RTA_WB SILK_NB SILK_WB SPEEX_NB SPEEX_WB
DSP Voice features: RTCP-XR
Channel Type: DspCh=30
HA
Security: IPSEC MediaEncryption StrongEncryption EncryptControlProtocol
Control Protocols: MSFT FEU=20000 MGCP SIP SASurvivability SBC=8000
Default features:
```

Load "Upgrade Key" file from your computer to the device

Reset with flash burn is required after file is loaded.

2. Back up the Software License Key currently installed on the device, as a precaution. If the new Software License Key does not comply with your requirements, you can re-load this backup to restore the device's original capabilities.
   a. In the 'Current Key' field, select the entire text string and copy it to any standard text file (e.g., Notepad).
   b. Save the text file with any file name and file extension (e.g., key.txt) to a folder on your computer.
3. Depending on whether you are loading a Software License Key file with a single Software License Key (i.e., one "S/N") or with multiple Software License Keys (i.e., more than one "S/N"), do one of the following:
   - **Loading a File with a Single Software License Key:**
     a. Open the Software License Key file using a text-based program such as Notepad.
     b. Copy-and-paste the string from the file to the 'Add a Software Upgrade Key' field.
     c. Click the **Add Key** button.
   - **Loading a File with Multiple Software License Keys:**
     a. In the 'Load Upgrade Key file ...' field, click the **Browse** button and navigate to the folder in which the Software License Key file is located on your computer.
     b. Click **Load File**; the new key is installed on the device.

   If the Software License Key is valid, it is burned to the device's flash memory and displayed in the 'Current Key' field.

4. Verify that the Software License Key was successfully installed, by doing one of the following:
   - In the Software Upgrade Key Status page, check that the listed features and capabilities activated by the installed Software License Key match those that were ordered.
   - Access the Syslog server and ensure that the following message appears in the Syslog server:
     "S/N___ Key Was Updated. The Board Needs to be Reloaded with ini file\n"
5. Reset the device; the new capabilities and resources enabled by the Software License Key are active.

> **Note:** If the Syslog server indicates that the Software License Key was unsuccessfully loaded (i.e., the "SN_" line is blank), do the following preliminary troubleshooting procedures:
>
> 1. Open the Software License Key file and check that the "S/N" line appears. If it does not appear, contact AudioCodes.
> 2. Verify that you have loaded the correct file. Open the file and ensure that the first line displays "[LicenseKeys]".
> 3. Verify that the content of the file has not been altered.

29.3.2.2 *Installing Software License Key using CLI*

To install the Software License Key using CLI, use the following commands:

- To install the Software License Key:

```
(config-system)# feature-key
```

- To view the Software License Key:

```
show system feature-key
```

29.4 Software Upgrade Wizard

The Web interface's Software Upgrade Wizard lets you easily upgrade the device's firmware (.cmp). The wizard also lets you load an *ini* file as well as auxiliary files such as a Call Progress Tone (CPT) file. If you want to load these additional files, you **must** also load a .cmp file during the wizard process.

The wizard can also upgrade the device when in High Availability (HA) mode. This software upgrade is referred to as Hitless Upgrade. Hitless Upgrade upgrades the device without disrupting traffic (i.e., current calls), whereby the upgrade process starts only after all current calls have ended. Hitless Upgrade does this by 1) first upgrading the redundant device, 2) performing a switchover from the active to the redundant device, 3) upgrading the previously active device, and then 4) switching over again to the previously active device.

The alternative to Hitless Upgrade is the non-Hitless method, whereby both the active and redundant devices are upgraded at the same time. Thus, this method is traffic-affecting and terminates current calls.

> **Warning:** The Software Upgrade Wizard requires the device to be reset at the end of the process, which may disrupt traffic. To avoid this, disable all traffic on the device before initiating the wizard by performing a graceful lock (see "Basic Maintenance" on page 292).

Notes:

- When upgrading from Version 6.4 to 6.8 using the Web interface, the Software Upgrade Wizard is not supported. If you require such an upgrade, contact AudioCodes support for a detailed upgrade procedure.
- You can upgrade the device to the latest software version as specified in the installed Software License Key. If you attempt to upgrade the device to a version that is later than the one specified in the Software License Key, the device remains at the current software version. For more information, contact your AudioCodes sales representative.
- When upgrading to a new software version, ensure that you have installed the new Software License Key.
- You can get the latest software files from AudioCodes Web site at http://www.audiocodes.com/downloads.
- Before upgrading the device, it is recommended that you save a copy of the device's configuration to your computer. If an upgrade failure occurs, you can then restore your configuration settings by uploading the backup file to the device. For saving and restoring configuration, see "Backing Up and Loading Configuration File" on page 316.
- When you activate the wizard, the rest of the Web interface is unavailable. After the files are successfully loaded, access to the full Web interface is restored.
- If you upgraded your .cmp and the "SW version mismatch" message appears in the Syslog or Web interface, then your Software License Key does not support the new .cmp file version. If this occurs, contact AudioCodes support for assistance.
- If you use the wizard to load an *ini* file, parameters excluded from the *ini* file are assigned default values (according to the .cmp file running on the device) thereby, overriding values previously defined for these parameters.
- You can schedule automatic loading of these files using HTTP/HTTPS.

The procedure below describes how to load files using the Web interface's Software Upgrade Wizard. To load files in the CLI, use the following commands:

- cmp file:

```
copy firmware from <URL>
```

- ini or auxiliary file:

```
copy <ini file or auxiliary file> from <URL>
```

- CLI script file:

```
copy cli-script from <URL>
```

- HA devices:
    - Hitless Software Upgrade:

```
# copy firmware from <URL and file name>
```

For example:

```
# copy firmware from https://1.1.1.1/device_SIP_F6.80A.cmp
```

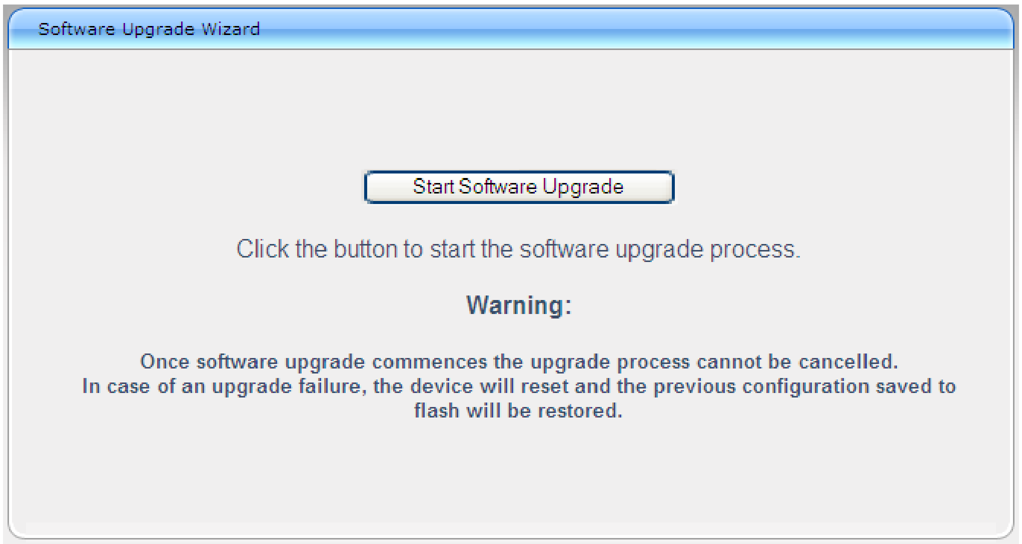
- Non-Hitless Software Upgrade:

```
# copy firmware from <URL and file name> non-hitless
```

➢ To load files using the Software Upgrade Wizard:

1. Stop all traffic on the device using the Graceful Lock feature (refer to the warning bulletin above).
2. Open the Software Upgrade wizard, by performing one of the following:
   - Select the **Maintenance** tab, click the **Software Update** menu, and then click **Software Upgrade Wizard**.
   - On the toolbar, click **Device Actions**, and then choose **Software Upgrade Wizard**.

Figure 133: Start Software Upgrade Wizard Screen



Click the button to start the software upgrade process.

Warning:

Once software upgrade commences the upgrade process cannot be cancelled.
In case of an upgrade failure, the device will reset and the previous configuration saved to flash will be restored.

3. Click **Start Software Upgrade**; the wizard starts.

> **Note:** At this stage, you can quit the Software Upgrade Wizard, by clicking **Cancel** ☒, without requiring a device reset. However, once you start loading the cmp file, the process must be completed with a device reset. If you choose to quit the process in any of the subsequent pages, the device resets.

4. Click **Browse**, navigate to the .cmp file, and then click **Load File**; a progress bar appears displaying the status of the loading process. When the .cmp file is successfully loaded to the device, a message appears notifying you of this.
5. If your device is in HA mode, select one of the following options:
   - Hitless Upgrade: (Default) Upgrade process without affecting traffic.
   - System Reset Upgrade: Both devices reset immediately after the .cmp file is loaded, regardless of traffic.

> **Note:** If you select Hitless Upgrade, you can upload only a .cmp file (auxiliary files and ini files cannot be uploaded as well).

6. If you **only** want to load a .cmp file, click **Reset** to reset the device with the newly loaded .cmp file. The device uses the existing configuration (*ini*) and auxiliary files. To load additional files, skip this step.

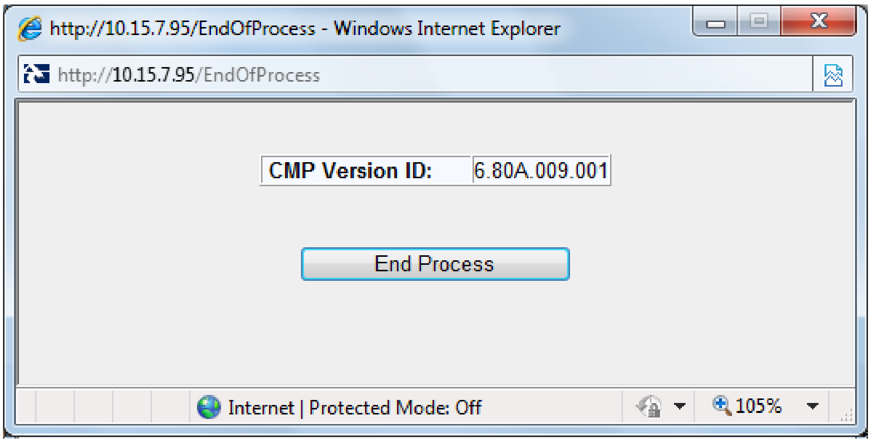
> **Note:** Device reset may take a few minutes depending on cmp file version (this may even take up to 10 minutes).

7. Click **Next** ; the wizard page for loading an *ini* file appears. You can now perform one of the following:
   - Load a new *ini* file: Click **Browse**, navigate to the *ini* file, and then click **Send File**; the *ini* file is loaded to the device and you're notified as to a successful loading.
   - Retain the existing configuration (*ini* file): Do not select an *ini* file, and make sure that the 'Use existing configuration' check box is selected (default).
   - Return the device's configuration settings to factory defaults: Do not select an *ini* file and clear the 'Use existing configuration' check box.
8. Click **Next** to progress to the relevant wizard pages for loading the desired auxiliary files. To return to the previous wizard page, click **Back** . As you navigate between wizard pages, the relevant file type corresponding to the wizard page is highlighted in the left pane.
9. When you have completed loading all the desired files, click **Next** until the last wizard page appears ("FINISH" is highlighted in the left pane).
10. Click **Reset** to complete the upgrade process; the device 'burns' the newly loaded files to flash memory and then resets.

> **Note:** Device reset may take a few minutes, depending on .cmp file version.

After the device resets, the End of Process wizard page appears, displaying the installed .cmp file and any other files (ini file and auxiliary files) that you may have installed at the same time:

Figure 134: Software Upgrade Process Completed Successfully



http://10.15.7.95/EndOfProcess - Windows Internet Explorer

http://10.15.7.95/EndOfProcess

CMP Version ID: 6.80A.009.001

End Process

Internet | Protected Mode: Off

105%

11. Click **End Process** to close the wizard; the Web Login dialog box appears.

12. Enter your login user name and password, and then click **OK**; a message box appears informing you of the new .cmp file.
13. Click **OK**; the Web interface becomes active, reflecting the upgraded device.

29.5 Backing Up and Loading Configuration File

You can save a copy/backup of the device's current configuration settings as an *ini* file to a folder on your computer, using the Configuration File page. The saved file includes only parameters that were modified and parameters with other than default values. The Configuration File page also allows you to load an *ini* file to the device. If the device has "lost" its configuration, you can restore the device's configuration by loading the previously saved *ini* file or by simply loading a newly created *ini* file.

You can also save the current configuration to a remote server

```
# copy cli-script to <URL of TFTP/HTTP/HTTPS server>
```

For example:

- Remote server:

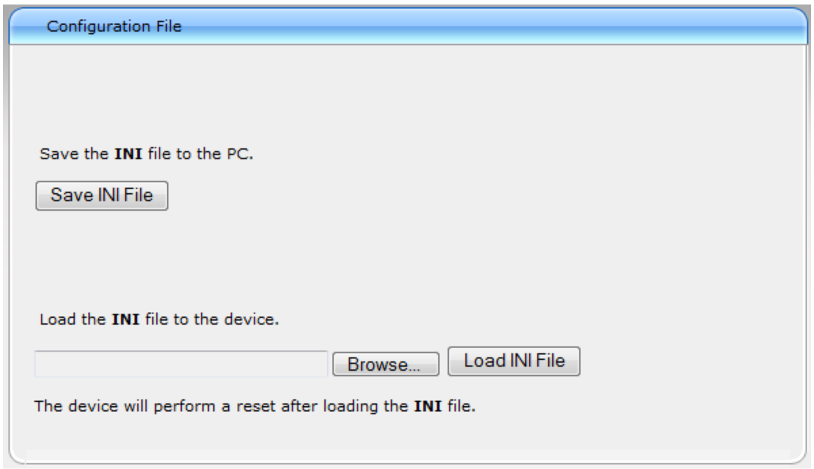
```
# copy cli-script to tftp://192.168.0.3/config-device1.txt
```

Note: When loading an *ini* file using the Configuration File page, parameters not included in the *ini* file are reset to default settings.

➢ To save or load an ini file:

1. Open the Configuration File page by doing one of the following:
   - From the Navigation tree, click the **Maintenance** tab, click the **Software Update** menu, and then click **Configuration File**.
   - On the toolbar, click **Device Actions**, and then from the drop-down menu, choose **Load Configuration File** or **Save Configuration File**.

Figure 135: Configuration File Page



Configuration File

Save the **INI** file to the PC.

Load the **INI** file to the device.

The device will perform a reset after loading the **INI** file.

2. To save the *ini* file to a folder on your computer:
   a. Click the **Save INI File** button; the File Download dialog box appears.
   b. Click the **Save** button, navigate to the folder where you want to save the file, and then click **Save**.

3. To load the *ini* file to the device:
   a. Click the **Browse** button, navigate to the folder where the file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
   b. Click the **Load INI File** button, and then at the prompt, click **OK**; the device uploads the file and then resets. Once complete, the Web Login screen appears, requesting you to enter your user name and password.

30 Automatic Update Mechanism

This chapter describes the device's Automatic Updated mechanism.

30.1 Overview of Automatic Update Mechanism

The device's Automatic Update mechanism can be used for automatic provisioning. Automatic provisioning is useful for large-scale deployment of devices. In some cases, the devices are shipped to the end customer directly from the manufacturer. In other cases, they may pass through a staging warehouse. Configuration may occur at the staging warehouse or at the end-customer premises.

The devices may be preconfigured during the manufacturing process (commonly known as private labeling). Typically, a two-stage configuration process is implemented whereby initial configuration includes only basic configuration, while the final configuration is done only when the device is deployed in the live network.

30.1.1 Files Updated by Automatic Update

The Automatic Update mechanism can be used to update the following device files:

- Software file (*cmp)*
- Auxiliary files (e.g., Call Progress Tones, SSL Cerrtificates)
- Configuration file (ini)

30.1.2 File Location for Automatic Update

The files that you want to automatically update on the device can be stored on any standard Web (HTTP/S), FTP, or NFS server. The files can be loaded periodically to the device using HTTP, HTTPS, FTP, or NFS. This mechanism can be used even when the device is installed behind NAT and firewalls.

The Automatic Update mechanism is done per file. This is configured by specifying, for each file, the file name and URL address of the provisioning server where the file is located. For a description of the parameters used to configure this, see "Automatic Update Parameters" on page 389.

Below are examples for configuring the file names and their location for Automatic Update:

- ini File:

```
IniFileURL = 'http://www.corp.com/configuration.ini'
CptFileURL = 'http://www.corp.com/call progress.dat'
AutoCmpFileUrl = 'http://www.corp.com/SIP_F6.80A.008.cmp
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# voice-configuration http://www.corp.com/configuration.ini
(automatic-update)# cal-progress-tone http://www.corp.com/call_progress.dat
(automatic-update)# auto-firmware http://www.corp.com/SIP_F6.80A.008.cmp
```

30.1.3 Triggering Automatic Update

The Automatic Update mechanism can be triggered by the following:

- Upon device startup (reset or power up).
- Upon startup, but before the device is operational, if the Secure Startup feature is enabled (see "Loading Files Securely by Disabling TFTP" on page 325).
- At a user-defined time of day (e.g., 18:00), configured by the ini file parameter AutoUpdatePredefinedTime.

- Periodically (e.g., every 60 minutes), configured by the ini file parameter AutoUpdateFrequency.
- Upon receipt of an SNMP request from a provisioning server.
- Upon receipt of a special SIP NOTIFY message (from a provisioning server) that includes an Event header with the AudioCodes proprietary value, "check-sync;reboot=false", as shown in the example below:

```
NOTIFY sip:<user>@<dsthost> SIP/2.0
To: sip:<user>@<dsthost>
From: sip:sipsak@<srchost>
CSeq: 10 NOTIFY
Call-ID: 1234@<srchost>
Event: check-sync;reboot=false
```

To enable this feature:

1. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).
2. Under the **Misc Parameters** group, set the 'SIP Remote Reset' parameter to **Enable**.
3. Click **Submit**.

30.1.4 Querying Provisioning Server for Updated Files

When the Automatic Update mechanism is triggered and HTTP/S is used, for each file and its configured URL, the device does the following:

1. The device establishes an HTTP/S connection with the URL host (provisioning server). If the connection is HTTPS, the device verifies the certificate of the provisioning server, and presents its own certificate if requested by the server.
2. The device queries the provisioning server for the requested file, by sending an HTTP Get request. This request contains the following HTTP headers:
   - **(Optional) HTTP User-Agent Header:** The HTTP User-Agent header identifies the device to the provisioning server. This header is used only if required by the provisioning server. By default, this header includes the device's product name, MAC address, and currently installed software and configuration versions. Based on its own dynamic applications for logic decision making, the provisioning server uses this information to check if it has relevant files available for the device and determines which files must be downloaded (working in conjunction with the HTTP If-Modified-Since header, described later on in this section).

     You can configure the information sent in the User-Agent header using the AupdHttpUserAgent parameter. This information can include any string value or the following supported string variable tags (case-sensitive):
     - **<NAME>** - product name, according to the installed Software License Key
     - **<MAC>** - device's MAC address
     - **<VER>** - software version currently installed on the device, e.g., "6.80.200.001"
     - **<CONF>** - configuration version, as configured in the ini file parameter, INIFileVersion or CLI command, configuration-version

     The device automatically populates these tag variables with actual values in the sent header:

```
User-Agent: Mozilla/4.0 (compatible; AudioCodes; <NAME>;<MAC>;<VER>;<CONF>).
```

For example, if you set AupdHttpUserAgent = <NAME>;<VER>, the device sends the following User-Agent header:

```
Mozilla/4.0 (compatible; AudioCodes; Mediant; 6.80.200.001)
```

> **Note:** Implementation of the HTTP User-Agent header by the provisioning server is the recommended method.

3. If the provisioning server has relevant files available for the device, the following occurs based on file type:

   - **Auxiliary Files (e.g., call progress tone / CPT):** The device downloads the Auxiliary file(s). These files are downloaded and updated only **once**, regardless of how many times the device may repeat the Automatic Update mechanism; once they are downloaded, the device discards their configured URLs. To update the Auxiliary file(s) again, you must modify their URL addresses and filenames configured by the relevant parameters (e.g., CptFileURL). For example:

Note: For security, it is recommended to use HTTPS with mutual authentication for transferring the SSL Private Key when downloading SSL certificates (Auxiliary file).

   - **Firmware (.cmp), Auto-Update, or configuration (ini) files:** The device uses the HTTP If-Modified-Since header to determine whether to download these files. This header contains the date and time (timestamp) when the device last downloaded the file. This date and time is regardless of whether the file was installed or not on the device. Below is an example of an If-Modified-Since header:

```
If-Modified-Since: Mon, 1 Dec 2013 19:43:31 GMT
```

     If the file on the provisioning server has not changed (modified) since the date and time specified in this header, the server replies with an HTTP 304 response and the file is not downloaded. If the file has been modified (i.e., server replies with a 2xx response), the device downloads the file. The device then compares the downloaded file's version with the currently installed version on its flash memory. If the downloaded file is of a later version, the device installs it after the device resets (which is only done after the device completes all file downloads); otherwise, the device does not reset and does not install the file.

     To enable this software (.cmp) file download method based on timestamp, use the ini file parameter, AutoCmpFileUrl or CLI command, auto-firmware. This parameter defines the software file name (.cmp) and the URL address where the file is located.

Note: When this method is used, there is typically no need for the provisioning server to check the device's current firmware version (using the HTTP-User-Agent header).

4. If the device receives an HTTP 301/302/303 redirect response from the provisioning server, it establishes a connection with the server at the new redirect URL and re-sends the HTTP Get request (see Step 2 above).

The Automatic Update mechanism checks the configured URLs of each file in the following order:

- ini file
- Auto Firmware*
- Firmware* (.cmp)
- Auxiliary Files

For each Automatic Update "run" (periodically or triggered via SNMP/SIP), the device attempts to download and install each configured file in the order listed above. Note that files marked with an asterisk require a device reset in order for them to be applied to the device after a successful download. Such reboots are "delayed" and performed only when the Automatic Update mechanism completes checking all the configured URLs.

Notes:

- For ini files, Cyclical Redundancy Check (CRC) is checked only if you set the AUPDCheckIfIniChanged parameter to [1] (enabled).
- For the Automatic Update mechanism, if you need to update the device's software (cmp) and configuration (ini), make sure that the new configuration is compatible with the currently ("old") installed software version; otherwise, the new configuration may not be "understood" by the software. The device first downloads and applies the new configuration, and only later does it download the new software and reset to apply the software.

30.2 Configuring Automatic Update

The procedure below describes how to configure the Automatic Update feature. It describes a scenario where the devices download a "master" configuration file with common settings from an HTTP server. This "master" file applies common configuration and instructs each device to download a specific configuration file based on the device's MAC address from an HTTP server.

Warnings: Do not use the Web interface to configure the device when the Automatic Update feature is implemented. If you do and save (burn) the new settings to the device's flash memory, the IniFileURL parameter is automatically set to 0 and Automatic Updates is consequently disabled. To enable Automatic Updates again, you would to need to re-load the ini file (using the Web interface or BootP) with the correct IniFileURL settings. As a safeguard to an unintended burn-to-flash when resetting the device, if the device is configured for Automatic Updates, the 'Burn To FLASH' field under the Reset Configuration group in the Web interface's Maintenance Actions page is automatically set to **No** by default.

Notes:

- For a description of the CLI parameters relating to Automatic Update, refer to the CLI Reference Guide.
- For a description of all the Automatic Update *ini* file parameters, see "Automatic Update Parameters" on page 389.

➢ To configure the Automatic Update feature (ini file example):

1. Setup a Web server (e.g., http://www.corp.com) and place all the required configuration files on this server.

2. For each device, preconfigure the following parameter (DHCP / DNS are assumed):

```
IniFileURL = 'http://www.corp.com/master_configuration.ini'
```

3. Create a file named *master_configuration.ini* with the following text:

```
# Common configuration for all devices
# ------------------------------------
CptFileURL = 'http://www.corp.com/call_progress.dat'
# Check for updates every 60 minutes
AutoUpdateFrequency = 60
# Additional configuration per device
# -----------------------------------
# Each device loads a file named based on its MAC address
# (e.g., config 00908F033512.ini)
IniFileURL = 'http://www.corp.com/config_<MAC>.ini'
# Reset the device after configuration is updated.
# The device resets after all files are processed.
ResetNow = 1
```

You can modify the master_configuration.ini file (or any of the config_<MAC>.ini files) at any time. The device queries for the latest version every 60 minutes and applies the new settings immediately.

4. For additional security, use HTTPS or FTPS. The device supports HTTPS (RFC 2818) and FTPS using the AUTH TLS method <draft-murray-auth-ftp-ssl-16>.

5.

6. To download configuration files from an NFS server, the NFS file system parameters should be defined in the *ini* file. The following is an example of an *ini* file for downloading files from NFS servers using NFS version 2:

```
# Define NFS servers for Automatic Update
[ NFSServers ]
FORMAT NFSServers Index = NFSServers HostOrIP,
NFSServers RootPath, NFSServers NfsVersion;
NFSServers 1 = 10.31.2.10, /usr/share, 2 ;
NFSServers 2 = 192.168.100.7, /d/shared, 2 ;
[ \NFSServers ]
CptFileUrl = 'file://10.31.2.10/usr/share/public/usa_tones.dat'
VpFileUrl = 'file://192.168.100.7/d/shared/gateways/voiceprompt.dat'
```

The following *ini* file example can be used to activate the Automatic Update mechanism.

```
# DNS is required for specifying domain names in URLs
[ InterfaceTable ]
FORMAT InterfaceTable Index = InterfaceTable ApplicationTypes,
InterfaceTable InterfaceMode, InterfaceTable IPAddress, InterfaceTable PrefixLength,
InterfaceTable Gateway, InterfaceTable VlanID, InterfaceTable InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable SecondaryDNSServerIPAddress, InterfaceTable UnderlyingInterface;
InterfaceTable 0 = 6, 10, 10.13.4.12, 16, 10.13.0.1, 1, Mng, 10.1.1.11, 0.0.0.0, ;
[ \InterfaceTable ]
# Load an extra configuration ini file using HTTP
IniFileURL = 'http://webserver.corp.com/Gateway/inifile.ini'
# Load Call Progress Tones file using HTTPS
CptFileUrl = 'https://10.31.2.17/usa_tones.dat'
# Load Voice Prompts file using FTPS with user 'root' and password 'wheel'
VPFileUrl = 'ftps://root:wheel@ftpserver.corp.com/vp.dat'
# Update every day at 03:00 AM
AutoUpdatePredefinedTime = '03:00'
# Note: The cmp file isn't updated since it's disabled by default
(AutoUpdateCmpFile).
```

Notes:

- The Automatic Update mechanism assumes that the external Web server conforms to the HTTP standard. If the Web server ignores the If-Modified-Since header or doesn't provide the current date and time during the HTTP 200 OK response, the device may reset itself repeatedly. To overcome this problem, modify the update frequency, using the AutoUpdateFrequency parameter.
- To load a different configuration file (*ini* files) per device, add the string "<MAC>" to the URL (e.g., IniFileURL = 'http://www.corp.com/config_<MAC>.ini'). This mnemonic is replaced with the device's hardware MAC address, resulting in an *ini* file name request that contains the device's MAC address (e.g., config_00908F033512.ini).
- To prevent the device from accidentally upgrading its software, by default the Automatic Update feature does not apply a downloaded cmp file even if its URL was configured (using the CmpFileURL parameter). To enable this, set the AutoUpdateCmpFile parameter to 1.
- To enable the device to automatically reset after an ini file has been loaded, set the ResetNow parameter to 1. This is important if the downloaded configuration file includes parameters that require a device reset for its settings to be applied.
- By default, parameters that are not included in the downloaded configuration file are set to default. To retain the current settings of these parameters, set the SetDefaultOnINIFileProcess parameter to 0.

30.3 Automatic Configuration Methods

This section describes available methods that can be used for automatic device configuration. The table below summarizes the automatic provisioning methods supported by the device:

Table 79: Automatic Provisioning Methods

BootP / TFTP	DHCP		Automatic Update Methods				SNMP (EMS)
	67	66	HTTP/S	TFTP	FTP	NFS	
No	No	No	Yes	Yes	Yes	Yes	Yes

30.3.1 DHCP-based Configuration Server

The DHCP server can be configured to automatically provide each device with a temporary IP address so that individual MAC addresses are not required. The DHCP server can provide additional networking parameters such as subnet mask, default gateway, primary and secondary DNS server, and two SIP server addresses. These network parameters have a time limit, after which the device must 'renew' its lease from the DHCP server.

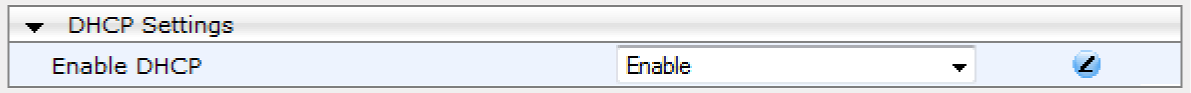
Notes:

- When using DHCP to acquire an IP address, the Interface table, VLANs and other advanced configuration options are disabled.
- For additional DHCP parameters, see "DHCP Parameters" on page 396.

➢ To enable DHCP:

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).

Figure 136: Enabling DHCP - Application Settings Page



DHCP Settings

Enable DHCP: Enable

2. From the 'Enable DHCP" drop-down list, select **Enable**.
3. Click **Submit**.
4. To activate the DHCP process, perform a cold reset by pressing the device's hardware reset button. A soft reset using the Web interface doesn't trigger the DHCP process and if done, this parameter reverts to **Disable**.

The device can use a host name in the DHCP request. The host name is set to acl_nnnnn, where *nnnnn* denotes the device's serial number. The serial number is the last six digits of the MAC address converted to decimal representation. In networks that support this feature and if the DHCP server registers this host name to a DNS server, you can access the device (through a Web browser) using the URL, http://acl_<serial number> (instead of using the device's IP address). For example, if the device's MAC address is 00908f010280, the DNS name is acl_66176.

Notes:

- If, during operation, the device's IP address is changed as a result of a DHCP renewal, the device automatically resets.
- If the DHCP server denies the use of the device's current IP address and specifies a different IP address (according to RFC 1541), the device must change its networking parameters. If this occurs while calls are in progress, they are not automatically rerouted to the new network address. Therefore, administrators are advised to configure DHCP servers to allow renewal of IP addresses.
- If the device's network cable is disconnected and then reconnected, a DHCP renewal is performed (to verify that the device is still connected to the same network). The device also includes its product name in the DHCP Option 60 Vendor Class Identifier. The DHCP server can use this product name to assign an IP address accordingly.
- After power-up, the device performs two distinct DHCP sequences. Only in the second sequence is DHCP Option 60 included. If the device is software reset (e.g., from the Web interface or SNMP), only a single DHCP sequence containing Option 60 is sent.

Below is an example configuration file for Linux DHCP server (dhcpd.conf). The devices are allocated temporary IP addresses in the range 10.31.4.53 to 10.31.4.75. TFTP is assumed to be on the same computer as the DHCP server (alternatively, the "next-server" directive may be used).

```
ddns-update-style ad-hoc;
default-lease-time 60;
max-lease-time 60;
class "gateways" {
    match if(substring(hardware, 1, 3) = 00:90:8f);
}
subnet 10.31.0.0 netmask 255.255.0.0 {
    pool {
         allow members of "audiocodes";
         range 10.31.4.53 10.31.4.75;
         filename "SIP F6.60A.217.003.cmp –fb;device.ini";
         option routers                 10.31.0.1;
         option subnet-mask             255.255.0.0;
    }
}
```

30.3.2 HTTP-based Automatic Updates

An HTTP or HTTPS server can be placed in the customer's network where configuration and software updates are available for download. This does not require additional servers at the customer premises and is NAT-safe. For example, assume the core network HTTPS server is https://www.corp.com. A master configuration ini file should be placed on the server, e.g., https://www.corp.com/gateways/master.ini. This file could point to additional ini files, auxiliary files (e.g., call progress tones), and software files (cmp), all on the same HTTP server or different HTTP servers in the network.

The main advantage of this method is that the HTTP configuration can be checked periodically when the device is deployed at the customer site. HTTP(S) is not sensitive to NAT devices, enabling configuration whenever needed without on-site intervention.

For additional security, the URL may contain a different port, and username and password.

The devices should only be preconfigured with the URL of the initial ini file, using one of the following methods:

- Methods described in "DHCP-based Configuration Server" on page 323 or above, via TFTP at a staging warehouse. The configuration URL is configured using the IniFileURL parameter.
- Private labeling.
- Manually on-site, using the RS-232 port or Web interface.

When the device is deployed at the customer site, local DHCP server provides the devices with IP addressing and DNS server information. From the URL provided in the DHCP response, the device can then contact the HTTP server at the core network and automatically download its configuration. The URL can be a simple file name or contain the device's MAC or IP address, e.g.:

- *http://corp.com/config-<MAC>.ini* - which becomes, for example, http://corp.com/config-00908f030012.ini
- *http://corp.com/<IP>/config.ini* - which becomes, for example, http://corp.com/192.168.0.7/config.ini

30.3.3 Configuration using FTP or NFS

Some networks block access to HTTP(S). The Automatic Update feature provides limited support for FTP/FTPS connectivity. Periodic polling for updates is not possible since these protocols don't support conditional fetching, i.e., updating files only if it is changed on the server.

The only difference between this method and those described in "HTTP-based Automatic Updates" on page 324 is that the protocol in the URL is "ftp" (instead of "http").

Notes:

- Unlike FTP, NFS is not NAT-safe.
- NFS v2/v3 is also supported.

30.4 Loading Files Securely by Disabling TFTP

The TFTP protocol is not considered secure and some network operators block it using a firewall. It is possible to disable TFTP completely, using the *ini* file parameter EnableSecureStartup (set to 1). This way, secure protocols such as HTTPS may be used to fetch the device configuration.

➢ To download the ini file to the device using HTTPS instead of TFTP:

1. Prepare the device's configuration file on an HTTPS server and obtain a URL to the file (e.g., https://192.168.100.53/gateways.ini).
2. Enable DHCP, if necessary.
3. Enable SSH and connect to it.
4. In the CLI, use the *ini* file parameters IniFileURL (for defining the URL of the configuration file) and EnableSecureStartup (for disabling TFTP), and then restart the device with the new configuration:

```
/conf/scp IniFileURL https://192.168.100.53/gateways.ini
/conf/scp EnableSecureStartup 1
/conf/sar bootp
```

Note: Once Secure Startup has been enabled, it can only be disabled by setting EnableSecureStartup to 0 using the CLI.

Reader's Notes

31 Restoring Factory Defaults

You can restore the device's configuration to factory defaults using one of the following methods:

- CLI (see "Restoring Defaults using CLI" on page 328)
- Loading an empty *ini* file (see "Restoring Defaults using an ini File" on page 329)

31.1 Restoring Defaults using CLI

The device can be restored to factory defaults using CLI, as described in the procedure below.

➢ To restore factory defaults using CLI:

1. Access the CLI:
   a. Connect the RS-232 serial port of the device to the communication port on your computer. For serial cabling, refer to the *Hardware Installation Manual*.
   b. Establish serial communication with the device using a serial communication program (such as HyperTerminal™) with the following communication port settings:
      - **Baud Rate:** 115,200 bps
      - **Data Bits:** 8
      - **Parity:** None
      - **Stop Bits:** 1
      - **Flow Control:** None
2. At the CLI prompt, type the username (default is "Admin" - case sensitive), and then press Enter:

```
# Username: Admin
```

3. At the prompt, type the password (default is "Admin" - case sensitive), and then press Enter:

```
# Password: Admin
```

4. At the prompt, type the following, and then press Enter:

```
# enable
```

5. At the prompt, type the password again, and then press Enter:

```
# Password: Admin
```

6. At the prompt, type the following to reset the device to default settings, and then press Enter:

```
# write factory
```

31.2 Restoring Defaults using an ini File

You can restore the device to factory default settings by loading an empty *ini* file to the device. This is done using the Web interface's Configuration File page (see ''Backing Up and Loading Configuration File'' on page 316). If the *ini* file does include content (e.g., parameters), ensure that they are on lines beginning with comment signs (i.e., semicolons ";") so that the device ignores them.

Note: The only settings that are not restored to default are the management (OAMP) LAN IP address and the Web interface's login user name and password.

Part VIII

Status, Performance Monitoring and Reporting

32 System Status

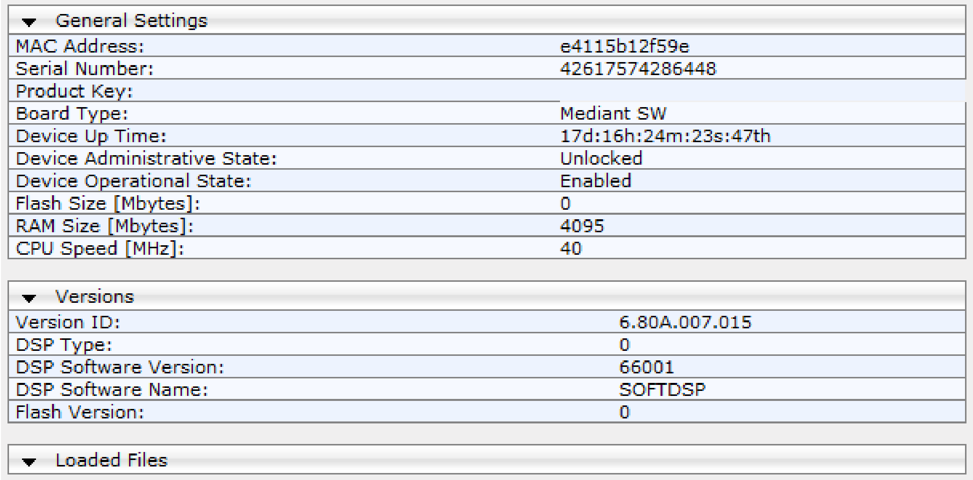
This section describes how to view various system statuses.

32.1 Viewing Device Information

The Device Information page displays hardware and software information about the device. This page also lists any Auxiliary files that have been installed on the device and allows you to remove them.

➢ To access the Device Information page:

■ Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**).



General Settings	
MAC Address:	e4115b12f59e
Serial Number:	42617574286448
Product Key:	
Board Type:	Mediant SW
Device Up Time:	17d:16h:24m:23s:47th
Device Administrative State:	Unlocked
Device Operational State:	Enabled
Flash Size [Mbytes]:	0
RAM Size [Mbytes]:	4095
CPU Speed [MHz]:	40

Versions	
Version ID:	6.80A.007.015
DSP Type:	0
DSP Software Version:	66001
DSP Software Name:	SOFTDSP
Flash Version:	0

➢ To delete a loaded file:

■ Click the **Delete** button corresponding to the file that you want to delete. Deleting a file takes effect only after device reset (see "Resetting the Device" on page 292).

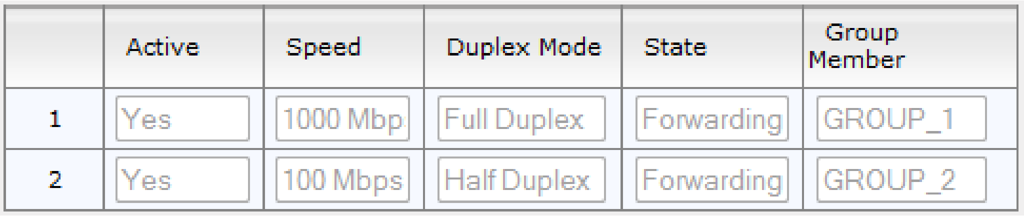
32.2 Viewing Ethernet Port Information

The Ethernet Port Information page displays read-only information about the Ethernet Port Group connections.

➢ To view Ethernet port information:

■ Open the Ethernet Port Information page:

- Navigation menu tree: **Status & Diagnostics** tab > **System Status** menu > **Ethernet Port Info**
- On the Home page, click any Ethernet port on the graphical display of the device (see "Viewing the Home Page" on page 41)



	Active	Speed	Duplex Mode	State	Group Member
1	Yes	1000 Mbp	Full Duplex	Forwarding	GROUP_1
2	Yes	100 Mbps	Half Duplex	Forwarding	GROUP_2

Table 80: Ethernet Port Information Parameters

Parameter	Description
Active	Displays whether the port is active ("Yes") or not ("No").
Speed	Displays the speed (in Mbps) of the Ethernet port.
Duplex Mode	Displays whether the port is half- or full-duplex.
State	Displays the state of the port: ▪ "Forwarding": Active port (data is being received and sent) ▪ "Disabled": Redundancy port
Group Member	Displays the port-pair group ID to which the port belongs.

33 Carrier-Grade Alarms

This section describes how to view the following types of alarms:

- Active alarms - see "Viewing Active Alarms" on page 333
- Alarm history - see "Viewing Alarm History" on page 333

33.1 Viewing Active Alarms

The Active Alarms page displays a list of currently active alarms. You can also access this page from the Home page (see "Viewing the Home Page" on page 41).

➢ To view the list of active alarms:

- Open the Active Alarms page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Active Alarms**).

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical (red)
  - Major (orange)
  - Minor (yellow)
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

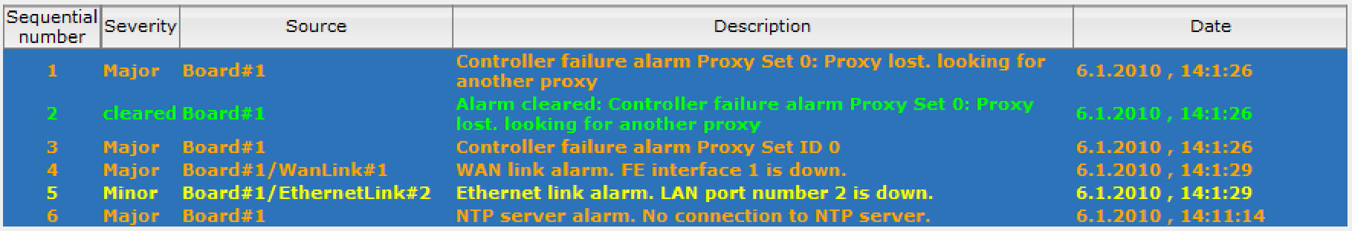
You can view the next 20 alarms (if exist), by clicking the **Go to page** button.

33.2 Viewing Alarm History

The Alarms History page displays a list of alarms that have been raised and traps that have been cleared.

➢ To view the list of history alarms:

- Open the Alarms History page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Alarms History**).



Sequential number	Severity	Source	Description	Date
1	Major	Board#1	Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
2	cleared	Board#1	Alarm cleared: Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
3	Major	Board#1	Controller failure alarm Proxy Set ID 0	6.1.2010 , 14:1:26
4	Major	Board#1/WanLink#1	WAN link alarm. FE interface 1 is down.	6.1.2010 , 14:1:29
5	Minor	Board#1/EthernetLink#2	Ethernet link alarm. LAN port number 2 is down.	6.1.2010 , 14:1:29
6	Major	Board#1	NTP server alarm. No connection to NTP server.	6.1.2010 , 14:11:14

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical (red)
  - Major (range)
  - Minor (yellow)
  - Cleared (green)
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

To view the next 20 alarms (if exist), click the **Go to page** button.

➢ To delete all the alarms in the table:

1. Click the **Delete History Table** button; a confirmation message box appears.
2. Click **OK** to confirm.

34 Performance Monitoring

This section describes how to view performance monitoring.

34.1 Viewing MOS per Media Realm

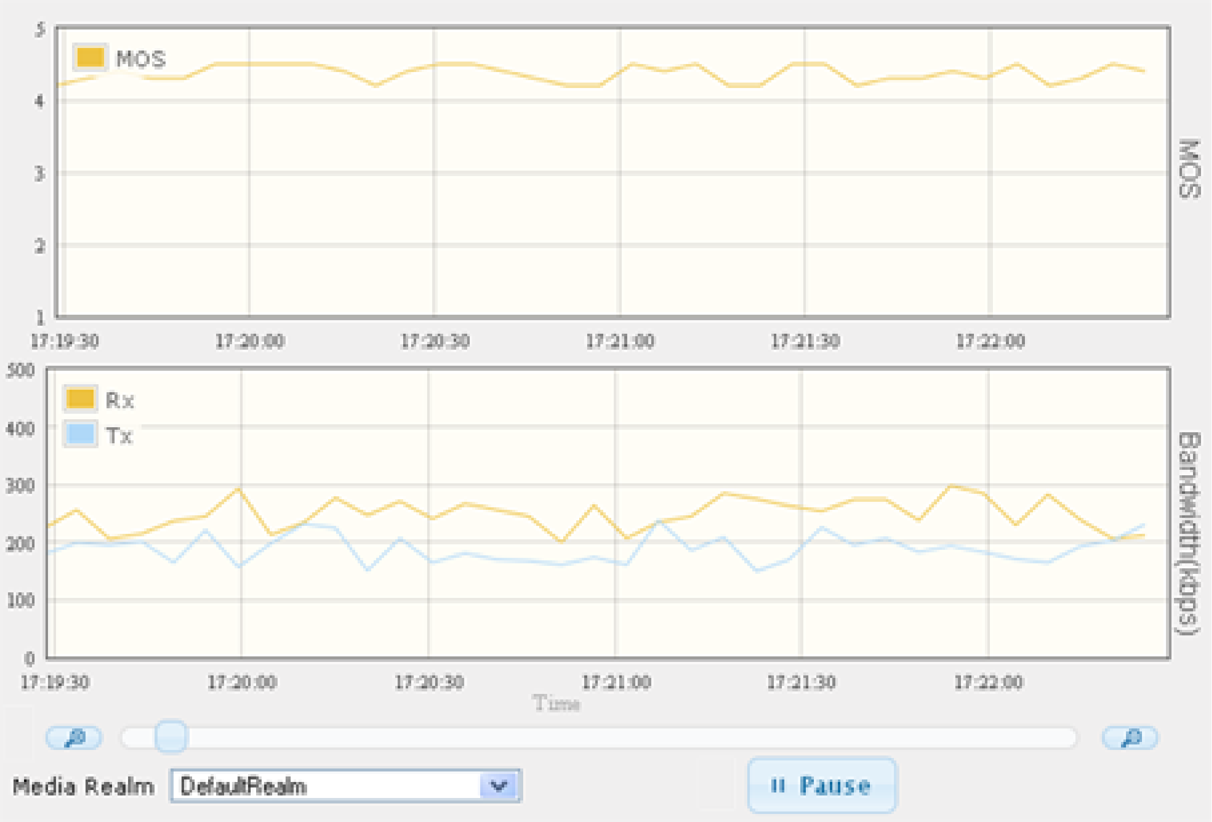
The MOS Per Media Realm page displays statistics on Media Realms (configured in ''Configuring Media Realms'' on page 164). This page provides two graphs:

- Upper graph: displays the Mean Opinion Score (MOS) quality in RTCP data per selected Media Realm.
- Lower graph: displays the bandwidth of transmitted media (in Kbps) in RTCP data per Media Realm.

➢ To view the MOS per Media Realm graph:

1. Open the MOS Per Media Realm page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **MOS Per Media Realm).**

Figure 137: MOS Per Media Realm Graph



Media Realm DefaultRealm

2. From the 'Media Realm' drop-down list, select the Media Realm for which you want to view.

Use the **Zoom In** 🔍 button to increase the displayed time resolution or the **Zoom Out** 🔍 button to decrease it. Instead of using these zoom buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

To pause the graph, click the **Pause** button; click **Play** to resume.

34.2 Viewing Quality of Experience

The Quality Of Experience page provides statistical information on calls per SRD or IP Group. The statistics can be further filtered to display incoming and/or outgoing call direction, and type of SIP dialog (INVITE, SUBSCRIBE, or all).

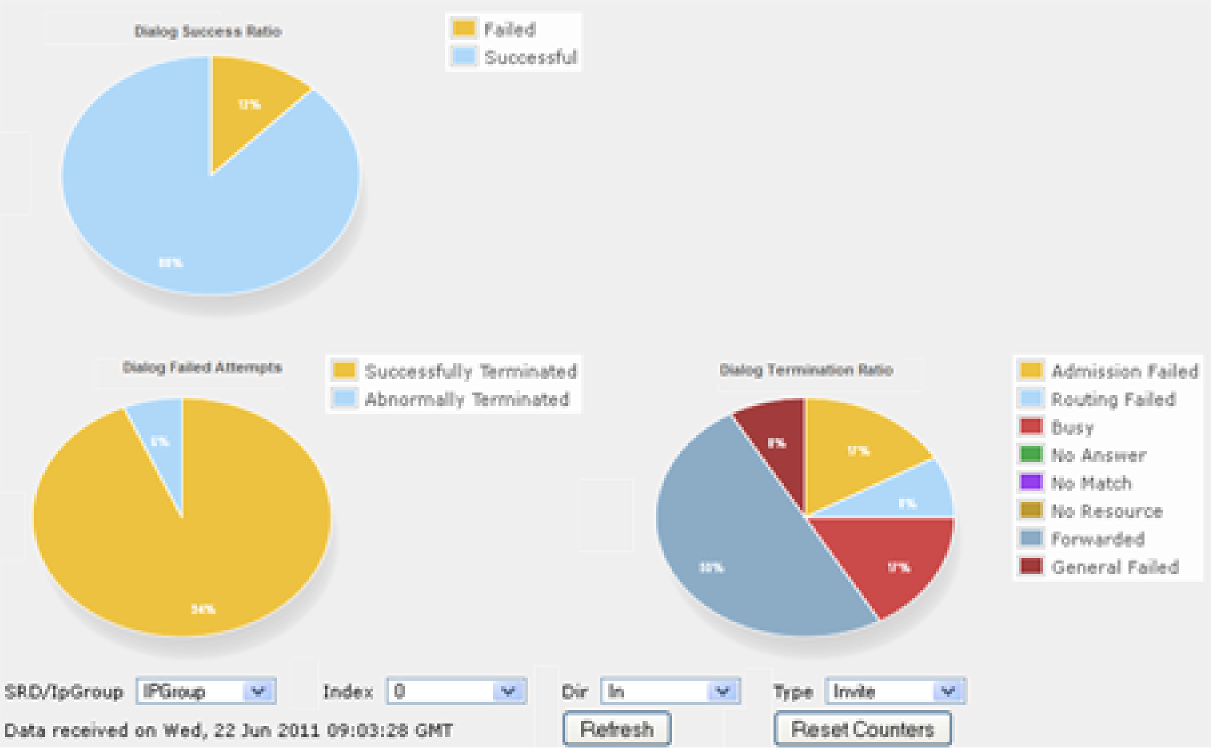
This page provides three pie charts:

- Dialog Success Ratio: displays the SIP call and subscribe (SUBSCRIBE) dialog success-failed ratio.
- Dialog Failed Attempts: displays the failed call attempts. This includes the number of calls and subscribes which were successfully and abnormally terminated.
- Dialog Termination Ratio: displays call termination by reason (e.g., due to no answer).

➢ To view Quality of Experience:

1. Open the Quality Of Experience page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **Quality Of Experience**).

Figure 138: Quality Of Experience Graph



SRD/IpGroup IPGroup
Index 0
Dir In
Type Invite
Data received on Wed, 22 Jun 2011 09:03:28 GMT

2. From the 'SRD/IpGroup' drop-down list, select whether you want to view QoE for an SRD or IP Group.
3. From the 'Index' drop-down list, select the SRD or IP Group index.
4. From the 'Dir' drop-down list, select the call direction:
   - **In** - incoming calls
   - **Out** - outgoing calls
   - **Both** - incoming and outgoing calls
5. From the 'Type' drop-down list, select the SIP message type:
   - **Invite** - INVITE
   - **Subscribe** - SUBSCRIBE
   - **Other** - all SIP messages

To refresh the charts, click **Refresh**. To reset the counters, click **Reset Counters**.

34.3 Viewing Average Call Duration

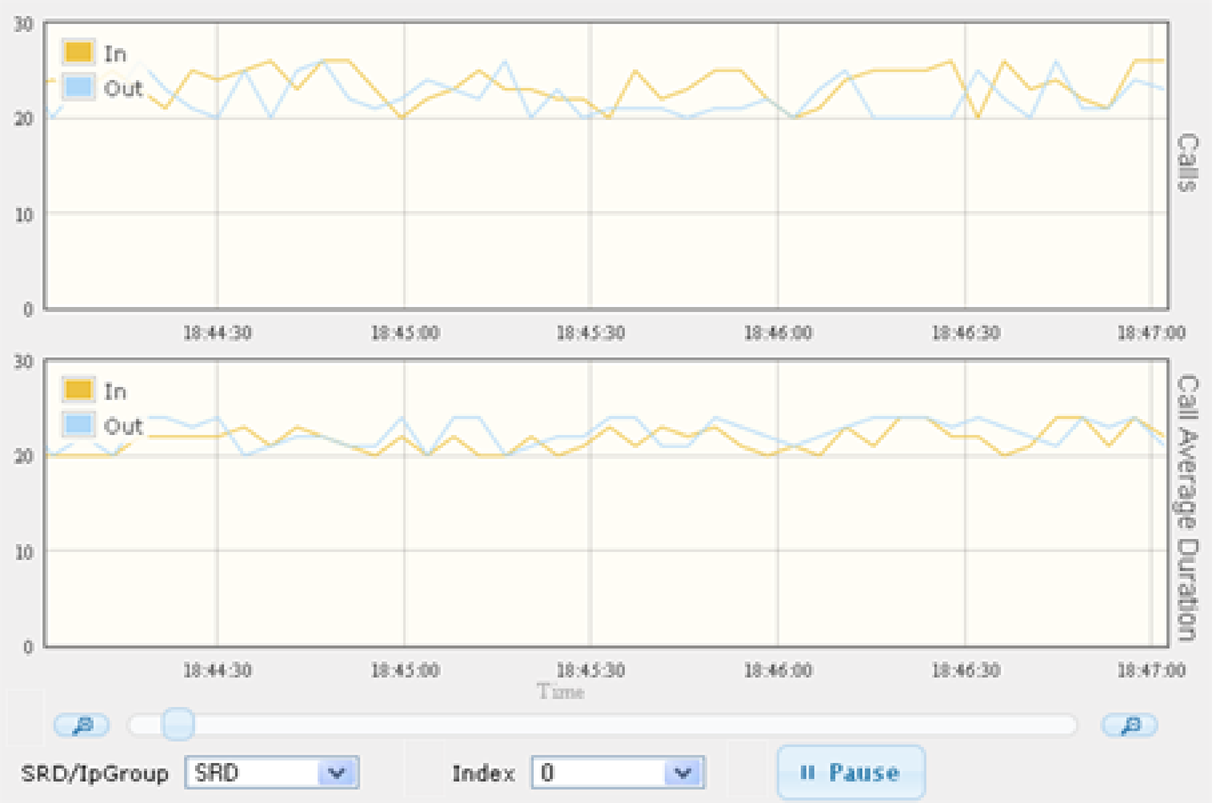
The Average Call Duration page displays information about a specific SRD or IP Group. This page includes two graphs:

- Upper graph: displays the number of calls (INVITEs).
- Lower graph: displays the average call duration.

➢ To view average call duration:

1. Open the Average Call Duration page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **Average Call Duration**).

Figure 139: Average Call Duration Graph



SRD/IpGroup SRD
Index 0

2. From the 'SRD/IpGroup' drop-down list, select whether you want to view information for an SRD or IP Group.
3. From the 'Index' drop-down list, select the SRD or IP Group index.

Use the **Zoom In** 🔍 button to increase the displayed time resolution or the **Zoom Out** 🔍 button to decrease it. Instead of using these zoom buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

To pause the graph, click the **Pause** button; click **Play** to resume.

Reader's Notes

35 VoIP Status

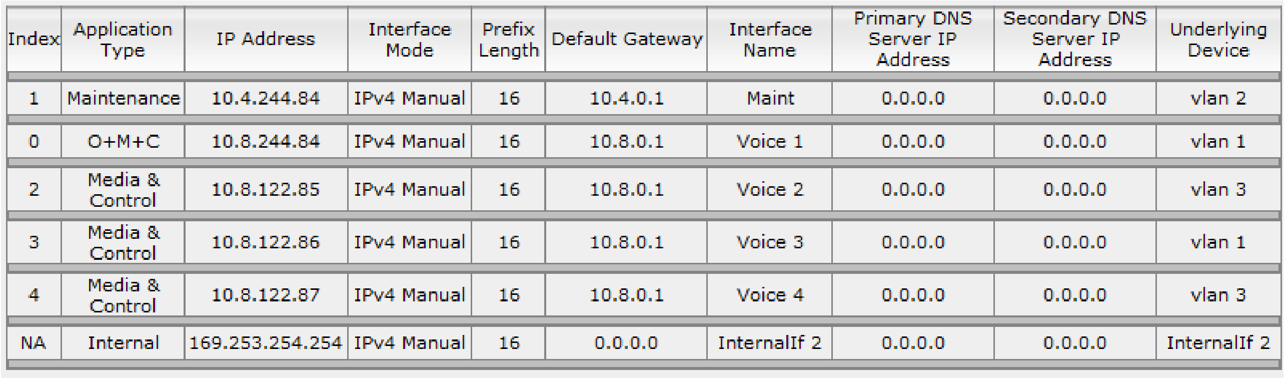
This section describes how to view VoIP status and statistics.

35.1 Viewing Active IP Interfaces

The IP Interface Status page displays the device's active IP interfaces that are listed in the Interface table (see "Configuring IP Network Interfaces" on page 91).

➢ To view active IP network interfaces:

- Open the IP Interface Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Interface Status**).



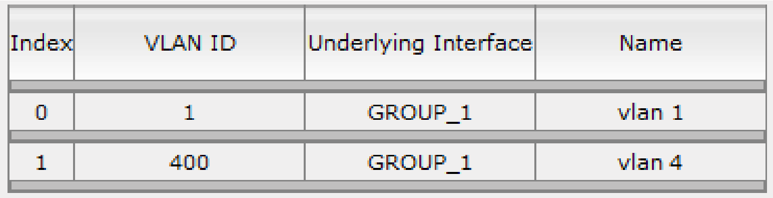
Index	Application Type	IP Address	Interface Mode	Prefix Length	Default Gateway	Interface Name	Primary DNS Server IP Address	Secondary DNS Server IP Address	Underlying Device
1	Maintenance	10.4.244.84	IPv4 Manual	16	10.4.0.1	Maint	0.0.0.0	0.0.0.0	vlan 2
0	O+M+C	10.8.244.84	IPv4 Manual	16	10.8.0.1	Voice 1	0.0.0.0	0.0.0.0	vlan 1
2	Media & Control	10.8.122.85	IPv4 Manual	16	10.8.0.1	Voice 2	0.0.0.0	0.0.0.0	vlan 3
3	Media & Control	10.8.122.86	IPv4 Manual	16	10.8.0.1	Voice 3	0.0.0.0	0.0.0.0	vlan 1
4	Media & Control	10.8.122.87	IPv4 Manual	16	10.8.0.1	Voice 4	0.0.0.0	0.0.0.0	vlan 3
NA	Internal	169.253.254.254	IPv4 Manual	16	0.0.0.0	InternalIf 2	0.0.0.0	0.0.0.0	InternalIf 2

35.2 Viewing Ethernet Device Status

The Ethernet Device Status page displays the configured Ethernet Devices that have been successfully applied to the device. For configuring Ethernet Devices, see "Configuring Underlying Ethernet Devices" on page 90.

➢ To view the configured and applied Ethernet Devices:

- Open the Ethernet Device Status page (**Status & Diagnostics** tab > **VoIP Status** menu >**Ethernet Device Status Table**).



Index	VLAN ID	Underlying Interface	Name
0	1	GROUP_1	vlan 1
1	400	GROUP_1	vlan 4

35.3 Viewing Static Routes Status

The IP Routing Status Table page displays the status of the static routes. These are routes configured in the Static Route table (see "Configuring Static IP Routing" on page 99) and routes through the Default Gateway.

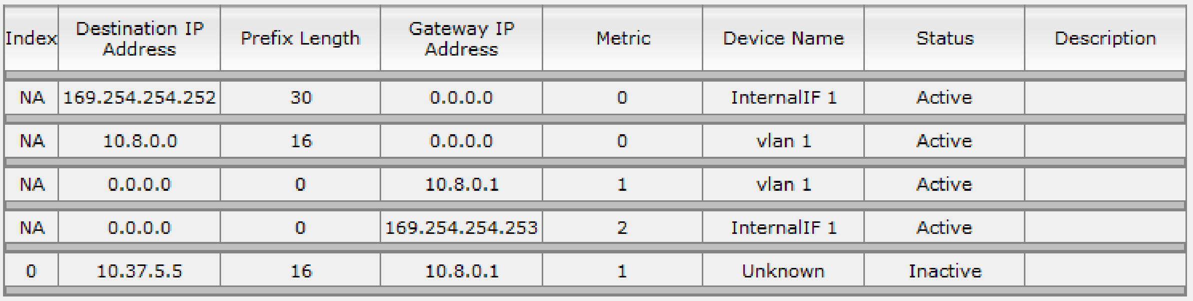
The status of the static routes can be one of the following:

- "Active": Static route is used by the device.
- "Inactive": Static route is not used. When the destination IP address is not on the same segment with the next hop, or the interface does not exist, the route state changes to "Inactive".

➢ To view the status of static IP routing:

- Open the IP Routing Status Table page (**Status & Diagnostics** tab > **VoIP Status** menu >**Static Route Status**).

Figure 140: IP Routing Status Table Page



Index	Destination IP Address	Prefix Length	Gateway IP Address	Metric	Device Name	Status	Description
NA	169.254.254.252	30	0.0.0.0	0	InternalIF 1	Active	
NA	10.8.0.0	16	0.0.0.0	0	vlan 1	Active	
NA	0.0.0.0	0	10.8.0.1	1	vlan 1	Active	
NA	0.0.0.0	0	169.254.254.253	2	InternalIF 1	Active	
0	10.37.5.5	16	10.8.0.1	1	Unknown	Inactive	

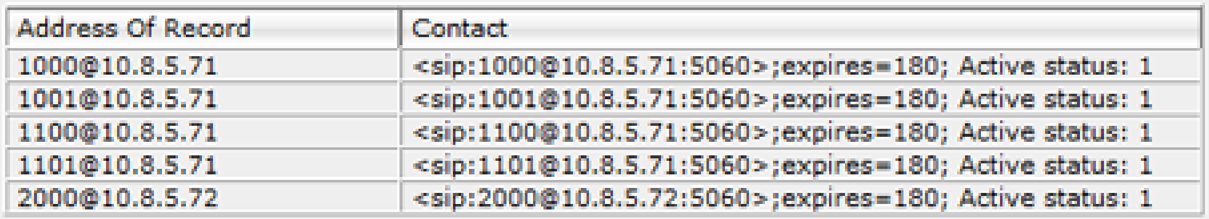
35.4 Viewing Registered Users

The SAS/SBC Registered Users page displays a list of registered SAS SBC in the device's Users Registration database.

➢ To view registered SAS SBC users:

- Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registered Users**).

Figure 141: SAS/SBC Registered Users Page



Address Of Record	Contact
1000@10.8.5.71	<sip:1000@10.8.5.71:5060>;expires=180; Active status: 1
1001@10.8.5.71	<sip:1001@10.8.5.71:5060>;expires=180; Active status: 1
1100@10.8.5.71	<sip:1100@10.8.5.71:5060>;expires=180; Active status: 1
1101@10.8.5.71	<sip:1101@10.8.5.71:5060>;expires=180; Active status: 1
2000@10.8.5.72	<sip:2000@10.8.5.72:5060>;expires=180; Active status: 1

Table 81: SAS/SBC Registered Users Parameters

Column Name	Description
Address of Record	An address-of-record (AOR) is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI (Contact) where the user might be available.
Contact	SIP URI that can be used to contact that specific instance of the User Agent for subsequent requests.

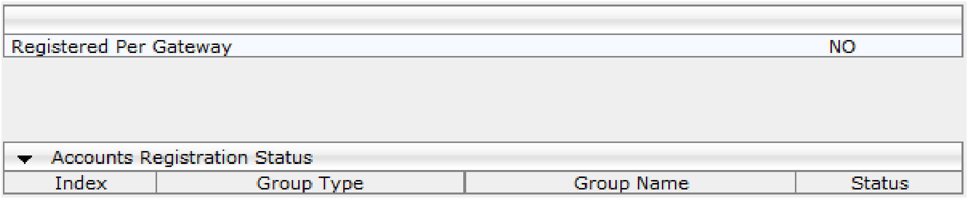
35.5 Viewing Registration Status

The Registration Status page displays the registration status of the device as a whole and SIP Accounts are registered to a SIP Registrar/Proxy server.

➢ To view the registration status:

- Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registration Status**).

Figure 142: Registration Status Page



Registered Per Gateway	NO

▼ Accounts Registration Status

Index	Group Type	Group Name	Status

- **Accounts Registration Status:** registration status based on the Accounts table (configured in "Configuring Registration Accounts" on page 187):
  - **Group Type:** type of served group - IP Group
  - **Group Name:** name of the served group, if applicable
  - **Status:** indicates whether or not the group is registered ("Registered" or "Unregistered")

Reader's Notes

36 Reporting Information to External Party

This section describes features for reporting various information to an external party.

36.1 RTP Control Protocol Extended Reports (RTCP XR)

RTP Control Protocol Extended Reports (RTCP XR) is a VoIP management control that defines a set of metrics containing information for assessing VoIP call quality and for diagnosing problems. RTCP XR (RFC 3611) extends the RTCP reports defined in RFC 3550 by providing additional VoIP metrics (Quality of Experience). RTCP XR information publishing is implemented in the device according to RFC 6035. This draft defines how a SIP User Agent (UA) publishes the detailed information to a defined collector. RTCP XR measures VoIP call quality such as packet loss, delay, signal / noise / echo levels, estimated R-factor, and mean opinion score (MOS). RTCP XR measures these parameters using metrics as listed in the table below.

Note: The RTCP XR feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 309.

RTCP XR messages containing key call-quality-related metrics are exchanged periodically (user-defined) between the device and the SIP UA. This allows an analyzer to monitor these metrics midstream, or a device to retrieve them using SNMP. The device sends RTCP XR reports to an Event State Compositor (ESC) server, using SIP PUBLISH messages. These reports can be sent at the end of each call and according to a user-defined interval between consecutive reports.

Table 82: RTCP XR Published VoIP Metrics

Group	Metric Name
General	Start Timestamp
	Stop Timestamp
	Call-ID
	Local Address (IP, Port & SSRC)
	Remote Address (IP, Port & SSRC)
Session Description	Payload Type
	Payload Description
	Sample Rate
	Frame Duration
	Frame Octets
	Frames per Packets
	Packet Loss Concealment
	Silence Suppression State
Jitter Buffer	Jitter Buffer Adaptive
	Jitter Buffer Rate
	Jitter Buffer Nominal
	Jitter Buffer Max

Group	Metric Name
	Jitter Buffer Abs Max
Packet Loss	Network Packet Loss Rate
	Jitter Buffer Discard Rate
Burst Gap Loss	Burst Loss Density
	Burst Duration
	Gap Loss Density
	Gap Duration
	Minimum Gap Threshold
Delay	Round Trip Delay
	End System Delay
	One Way Delay
	Interarrival Jitter
	Min Absolute Jitter
	Signal
	Signal Level
	Noise Level
	Residual Echo Return Noise
Quality Estimates	Listening Quality R
	RLQ Est. Algorithm
	Conversational Quality R
	RCQ Est. Algorithm
	External R In
	Ext. R In Est. Algorithm
	External R Out
	Ext. R Out Est. Algorithm
	MOS-LQ
	MOS-LQ Est. Algorithm
	MOS-CQ
	MOS-CQ Est. Algorithm
	QoE Est. Algorithm

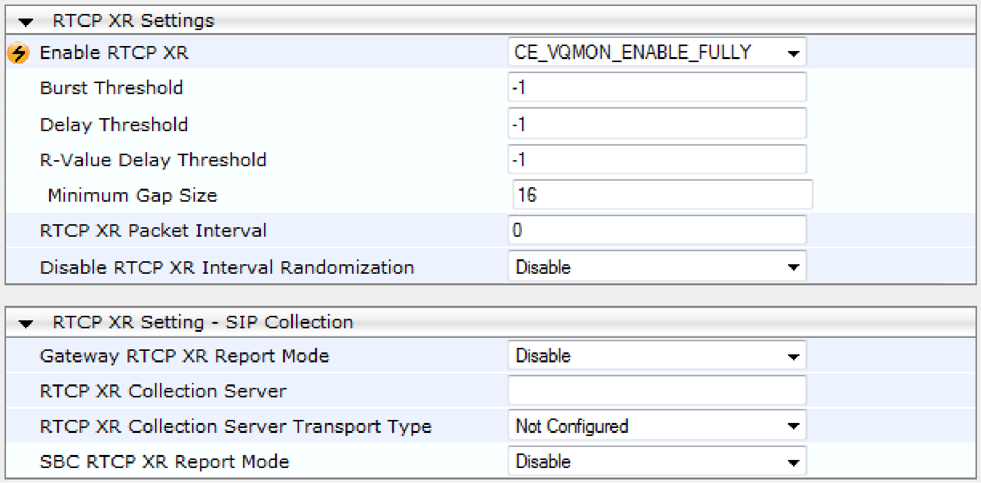
Below shows an example of a SIP PUBLISH message sent with RTCP XR and QoE information:

```
PUBLISH sip:10.8.4.61 SIP/2.0
    Via: SIP/2.0/UDP 10.8.61.16;branch=z9hG4bKac45186128
    Max-Forwards: 70
    From: <sip:10.8.61.16>;tag=1c44171734
    To: <sip:10.8.61.16>
    Call-ID: 441338942842012155836@10.8.61.16
    CSeq: 1 PUBLISH
    Contact: <sip:10.8.61.16:5060>
    Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
    Event: vq-rtcpxr
    Expires: 3600
    User-Agent: Audiocodes-Sip-Gateway-Mediant /v.6.80A.037.009
    Content-Type: application/vq-rtcpxr
    Content-Length: 710
    VQIntervalReport
    CallID=137461752128420121558350@10.8.61.16
    LocalID: <sip:12345@10.8.61.16>
    RemoteID: <sip:54321@10.8.61.18>
    OrigID: <sip:12345@10.8.61.16>
    LocalAddr: IP=10.8.61.16 Port=6110 SSRC=0xce110633
    RemoteAddr: IP=10.8.61.18 Port=6050 SSRC=0xffffffff
    LocalGroup:
    RemoteGroup:
    LocalMAC: 00:90:8f:2e:3c:67
    LocalMetrics:
    Timestamps: START=2012-04-28T15:58:36Z STOP=2012-04-28T15:58:36Z
    SessionDesc: PT=8 PD=PCMA SR=8000 FD=20 PLC=3 SSUP=Off
    JitterBuffer: JBA=3 JBR=0 JBN=0 JBM=0 JBX=300
    PacketLoss: NLR=0.00 JDR=0.00
    BurstGapLoss: BLD=0.00 BD=0 GLD=0.00 GD=0 GMIN=16
    Delay: RTD=0 ESD=0
    QualityEst:
    DialogID:137461752128420121558350@10.8.61.16;to-tag=1c252030485; from-
tag=1c1374725246
```

➢ To configure RTCP XR:

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** > **RTP/RTCP Settings**). The RTCP XR parameters are listed under the RTCP XR Settings group:

Figure 143: RTCP XR Parameters in RTP/RTCP Settings Page



RTCP XR Settings	
Enable RTCP XR	CE_VQMON_ENABLE_FULLY
Burst Threshold	-1
Delay Threshold	-1
R-Value Delay Threshold	-1
Minimum Gap Size	16
RTCP XR Packet Interval	0
Disable RTCP XR Interval Randomization	Disable
RTCP XR Setting - SIP Collection	
Gateway RTCP XR Report Mode	Disable
RTCP XR Collection Server	
RTCP XR Collection Server Transport Type	Not Configured
SBC RTCP XR Report Mode	Disable

2. Under the RTCP XR Settings group, configure the following:
   - 'Enable RTCP XR' (*VQMonEnable*) - enables voice quality monitoring and RTCP XR.
   - 'Burst Threshold' (*VQMonBurstHR*) - defines the voice quality monitoring excessive burst alert threshold.
   - 'Delay Threshold' (*VQMonDelayTHR*) - defines the voice quality monitoring excessive delay alert threshold.

- 'R-Value Delay Threshold' (*VQMonEOCRValTHR*) - defines the voice quality monitoring end of call low quality alert threshold.
- 'Minimum Gap Size' (*VQMonGMin*) - defines the voice quality monitoring minimum gap size (number of frames).
- 'RTCP XR Packet Interval' (*RTCPInterval*) - defines the time interval between adjacent RTCP reports.
- 'Disable RTCP XR Interval Randomization' (*DisableRTCPRandomize*) - determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval.

3. Under the RTCP XR Setting - SIP Collection group, configure the following:
   - 'Gateway RTCP XR Report Mode' (*RTCPXRReportMode*) - determines whether RTCP XR reports are sent to the ESC server and defines the interval at which they are sent.
   - 'RTCP XR Collection Server' (*RTCPXREscIP*) - defines the IP address of the ESC server.
   - 'RTCP XR Collection Server Transport Type' (*RTCPXRESCTransportType*) - determines the transport layer for outgoing SIP dialogs initiated by the device to the ESC server.
   - 'SBC RTCP XR Report Mode' (SBCRtcpXrReportMode) - enables the sending of RTCP XR reports of QoE metrics at the end of each call session (i.e., after a SIP BYE).
4. Click **Submit**, and then reset the device with a save ("burn") for your settings to take effect.

36.2 Generating Call Detail Records

The Call Detail Record (CDR) contains vital statistic information on calls made from the device. The device can be configured to generate and report CDRs for various stages of the call, including SIP messages and/or media. You can configure when CDRs for a call are generated, for example, only at the end of the call or only at the start and end of the call. Once generated, the device sends the CDRs to a user-defined Syslog server.

The CDR Syslog message complies with RFC 3161 and is identified by Facility 17 (local1) and Severity 6 (Informational).

For CDR in RADIUS format, see "Configuring RADIUS Accounting" on page 350.

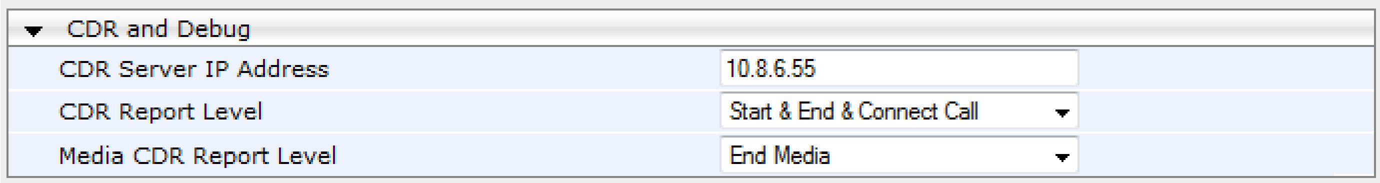
36.2.1 Configuring CDR Reporting

The procedure below describes how to configure CDR reporting.

➢ To configure CDR reporting:

1. Enable the Syslog feature for sending log messages generated by the device to a collecting log message server. For more information, see "Configuring Syslog" on page 358.
2. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**). The CDR parameters appear under the 'CDR and Debug' group, as shown below:

Figure 144: CDR Parameters in Advanced Parameters Page



CDR and Debug

CDR Server IP Address	10.8.6.55
CDR Report Level	Start & End & Connect Call
Media CDR Report Level	End Media

3. Configure the parameters as required. For a description of the parameters, see "Syslog, CDR and Debug Parameters" on page 400.
4. Click **Submit**.

Note: If the CDR server IP address is not configured, the CDRs are sent to the Syslog server, configured in "Configuring Syslog" on page 358.

36.2.2 CDR Field Description

This section describes the CDR fields that are generated by the device.

36.2.2.1 CDR Fields for SBC Signaling

The CDR fields for SBC signaling are listed in the table below. The signaling CDRs are published for each SBC leg.

Table 83: CDR Fields for SBC Signaling

CDR Field Name	Description
SBCReportType	Report Type: ▪ **CALL_START** ▪ **CALL_CONNECT** ▪ **CALL_END** ▪ **DIALOG_START** ▪ **DIALOG_END**
EPTyp	Endpoint type (**SBC**)
SIPMethod	SIP message type
SIPCallId	Unique ID of call
SessionId	Unique Session ID
Orig	Call originator: ▪ **LCL -** for local ▪ **RMT -** for remote
Caller	Caller party (source)
Called	Called party (destination)
SourceIp	Source IP address
SourcePort	Source UDP port
DestIp	Destination IP address
DestPort	Destination UDP port
TransportType	Transport type: ▪ **UDP** ▪ **TCP** ▪ **TLS**
SrcURI	Source URI
SrcURIBeforeMap	Source URI before manipulation
DstURI	Destination URI
DstURIBeforeMap	Destination URI before manipulation
Durat	Call duration

CDR Field Name	Description
TrmSd	Termination side (local or remote)
TrmReason	Termination reason
TrmReasonCategory	Termination reason category: ▪ Calls with duration 0 (i.e., not connected): ✓ **NO_ANSWER -** GWAPP_NORMAL_CALL_CLEAR, GWAPP_NO_USER_RESPONDING, GWAPP_NO_ANSWER_FROM_USER_ALERTED ✓ **BUSY -** GWAPP_USER_BUSY ✓ **NO_RESOURCES -** GWAPP_RESOUUCE_UNAVAILABLE_UNSPECIFIED, RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT, RESOURCE_BECAUSE_NO_TRANSCODING_RESOURCES_LEFT, RELEASE_BECAUSE_GW_LOCKED ✓ **NO_MATCH -** RELEASE_BECAUSE_UNMATCHED_CAPABILITIES ✓ **FORWARDED -** RELEASE_BECAUSE_FORWARD ✓ **GENERAL_FAILED -** any other reason ▪ Calls with duration: ✓ **NORMAL_CALL_CLEAR -** GWAPP_NORMAL_CALL_CLEAR ✓ **ABNORMALLY_TERMINATED -** Anything else ▪ **N/A -** Reasons not belonging to above categories
SetupTime	Call setup time
ConnectTime	Call connect time
ReleaseTime	Call release time
RedirectReason	Redirect reason
RedirectURINum	Redirection URI
RedirectURINumBeforeMap	Redirect URI number before manipulation
TxSigIPDiffServ	Signaling IP DiffServ
IPGroup	IP Group description
SrdId	SRD name
SIPInterfaceId	SIP Interface ID
ProxySetId	Proxy Set ID
IpProfileId	IP Profile name
MediaRealmId	Media Realm name
DirectMedia	Direct media or traversing SBC: ▪ yes ▪ no
SIPTrmReason	SIP call termination reason (BYE, CANCEL, or SIP error codes, e.g., 404)

CDR Field Name	Description
SipTermDesc	Description of SIP termination reason: ▪ SIP Reason header, if exists, for example: SIP ;cause=200 ;text="Call completed elsewhere". ▪ If no SIP Reason header exists, the description is taken from the reason text, if exists, of the SIP response code, for example: "417 Unknown Resource-Priority". ▪ If no reason text exists in the SIP response code, the description is taken from an internal SIP response mapping mechanism. For example, if the device receives a SIP response "422", it sends in the CDR "422 Session Interval Too Small method" as the description.

An example of an SBC signaling CDR sent by the device is shown below:

```
[S=1] |SBCReportType |EPTyp| SIPCallId| SessionId |Orig |SourceIp |SourcePort
|DestIp |DestPort |TransportType |SrcURI |SrcURIBeforeMap |DstURI |DstURIBeforeMap
|Durat |TrmSd |TrmReason |TrmReasonCategory |SetupTime |ConnectTime |ReleaseTime
|RedirectReason |RedirectURINum |RedirectURINumBeforeMap |TxSigIPDiffServ |IPGroup
(description) |SrdId (name) |SIPInterfaceId |ProxySetId |IpProfileId (name)
|MediaRealmId (name) |DirectMedia |SIPTrmReason
[S=3] |CALL END |SBC |170369730753201211288@10.132.10.245 |0 |RMT |10.132.10.245
|5060 |10.132.10.250 |5070 |UDP |103@audiocodes.com |103@audiocodes.com
|101@10.132.10.250 |101@10.132.10.250 |0 |RMT |GWAPP NORMAL CALL CLEAR |NO ANSWER
|06:13:54.950 UTC Thu Mar 02 2012 | |06:14:01.175 UTC Thu Mar 02 2012 |-1 | | |40 |2
() |0 (5070SRD) |2 |3 |0 () |0 (lanmedia) |no |CANCEL
```

36.2.2.2 CDR Fields for SBC Media

The CDR fields for SBC media are listed in the table below. The media CDRs are published for each active media stream, thereby allowing multiple media CDRs, where each media CDR has a unique call ID corresponding to the signaling CDR.

Table 84: CDR Fields for SBC Media

CDR Field Name	Description
MediaReportType	Report type (media start, update, or end)
SIPCallId	Unique call ID
Cid	Channel CID
MediaType	Media type (audio, video, or text)
Coder	Coder name
PacketInterval	Coder packet interval
LocalRtpIp	Local RTP IP address
LocalRtpPort	Local RTP port
RemoteRtpIp	Remote RTP IP address
RemoteRtpPort	Remote RTP port
InPackets	Number of received packets
OutPackets	Number of sent packets
LocalPackLoss	Local packet loss
RemotePackLoss	Remote packet loss
RTPdelay	RTP delay

CDR Field Name	Description
RTPjitter	RTP jitter
TxRTPssrc	Tx RTP SSRC
RxRTPssrc	Local RTP SSRC
LocalRFactor	Local conversation quality
RemoteRFactor	Remote conversation quality
LocalMosCQ	Local MOS for conversation
RemoteMosCQ	Remote MOS for conversation
TxRTPIPDiffServ	Media IP DiffServ
LatchedRtpIp	Remote IP address of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal.
LatchedRtpPort	Remote RTP port of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal.
LatchedT38Ip	Latching of a new T.38 stream - new IP address
LatchedT38Port	Latching of a new T.38 stream - new port

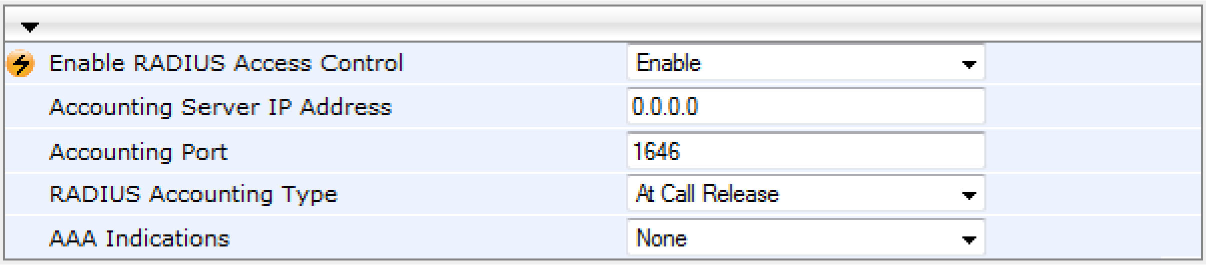
36.3 Configuring RADIUS Accounting

The device can send accounting data of SIP calls as call detail records (CDR) to a RADIUS Accounting server. The device can send the accounting messages to the RADIUS server upon call release, call connection and release, or call setup and release. For a list of the CDR attributes, see the table following the procedure below.

➢ To configure RADIUS accounting:

1. Open the RADIUS Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **RADIUS Parameters Settings**).

Figure 145: RADIUS Accounting Parameters Page



Enable RADIUS Access Control	Enable
Accounting Server IP Address	0.0.0.0
Accounting Port	1646
RADIUS Accounting Type	At Call Release
AAA Indications	None

2. Set the 'Enable RADIUS Access Control' parameter to **Enable**.
3. Configure the remaining parameters as required. For a description of these parameters, see "RADIUS Parameters" on page 414.
4. Click **Submit**.
5. For your settings to take effect, reset the device with a flash burn.

The table below lists the RADIUS Accounting CDR attributes included in the communication packets transmitted between the device and a RADIUS server.

Table 85: Supported RADIUS Accounting CDR Attributes

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
Request Attributes						
1	user-name	-	Account number or calling party number or blank	String up to 15 digits long	5421385747	Start Acc Stop Acc
4	nas-ip-address	-	IP address of the requesting device	Numeric	192.168.14.43	Start Acc Stop Acc
6	service-type	-	Type of service requested	Numeric	1: login	Start Acc Stop Acc
26	h323-incoming-conf-id	1	SIP call identifier	Up to 32 octets	-	Start Acc Stop Acc
26	h323-remote-address	23	IP address of the remote gateway	Numeric	-	Stop Acc
26	h323-conf-id	24	H.323/SIP call identifier	Up to 32 octets	-	Start Acc Stop Acc
26	h323-setup-time	25	Setup time in NTP format 1	String	-	Start Acc Stop Acc
26	h323-call-origin	26	The call's originator: Answering (IP) or Originator (PSTN)	String	Answer, Originate etc	Start Acc Stop Acc
26	h323-call-type	27	Protocol type or family used on this leg of the call	String	VoIP	Start Acc Stop Acc
26	h323-connect-time	28	Connect time in NTP format	String	-	Stop Acc
26	h323-disconnect-time	29	Disconnect time in NTP format	String	-	Stop Acc
26	H323-Disconnect-Cause	30	Q.931 disconnect cause code	Numeric	-	Stop Acc
26	h323-gw-id	33	Name of the gateway	String	SIPIDString	Start Acc Stop Acc
26	sip-call-id	34	SIP Call ID	String	abcde@ac.com	Start Acc Stop Acc
26	call-terminator	35	The call's terminator: PSTN-terminated call (Yes); IP-terminated call (No).	String	Yes, No	Stop Acc
30	called-station-id	-	Destination URI	String	8004567145	Start Acc
31	calling-station-id	-	Source URI	String	5135672127	Start Acc Stop Acc
40	acct-status-type	-	Account Request Type (start or stop) **Note:** 'start' isn't supported on the Calling Card application.	Numeric	1: start, 2: stop	Start Acc Stop Acc

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
41	acct-delay-time	-	No. of seconds tried in sending a particular record	Numeric	5	Start Acc Stop Acc
44	acct-session-id	-	A unique accounting identifier - match start & stop	String	34832	Start Acc Stop Acc
46	acct-session-time	-	For how many seconds the user received the service	Numeric	-	Stop Acc
47	acct-input-packets	-	Number of packets received during the call	Numeric	-	Stop Acc
48	acct-oputput-packets	-	Number of packets sent during the call	Numeric	-	Stop Acc
61	nas-port-type	-	Physical port type of device on which the call is active	String	0: Asynchronous	Start Acc Stop Acc
Response Attributes						
26	h323-return-code	103	The reason for failing authentication (0 = ok, other number failed)	Numeric	0 Request accepted	Stop Acc
44	acct-session-id	-	A unique accounting identifier – match start & stop	String	-	Stop Acc

Below is an example of RADIUS Accounting, where the non-standard parameters are preceded with brackets:

```
Accounting-Request (361)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2


acct-session-time = 1
acct-input-packets = 122
acct-output-packets = 220
called-station-id = 201
calling-station-id = 202
// Accounting non-standard parameters:
(4923 33) h323-gw-id =
(4923 23) h323-remote-address = 212.179.22.214
(4923 1) h323-ivr-out = h323-incoming-conf-id:02102944 600a1899 3fd61009 0e2f3cc5
(4923 30) h323-disconnect-cause = 22 (0x16)
(4923 27) h323-call-type = VOIP
(4923 26) h323-call-origin = Originate
(4923 24) h323-conf-id = 02102944 600a1899 3fd61009 0e2f3cc5
```

Reader's Notes

Part IX

Diagnostics

37 Syslog and Debug Recordings

Syslog is an event notification protocol that enables a device to send event notification messages across IP networks to event message collectors, also known as Syslog servers. The device contains an embedded Syslog client, which sends error reports / events that it generates to a remote Syslog server using the IP / UDP protocol. This information is a collection of error, warning, and system messages that records every internal operation of the device.

For receiving Syslog messages generated by the device, you can use any of the following Syslog servers:

- **Device's embedded Syslog server:** The device provides an embedded Syslog server, which is accessed through the Web interface. This provides limited Syslog server functionality.
- **Wireshark:** Third-party network protocol analyzer (http://www.wireshark.org).
- **Third-party, Syslog server:** Any third-party Syslog server program that enables filtering of messages according to parameters such as priority, IP sender address, time, and date.

37.1 Syslog Message Format

The Syslog message is sent from the device to a Syslog server as an ASCII (American Standard Code for Information Interchange) message. Syslog uses UDP as its underlying transport layer mechanism. By default, UDP port 514 is assigned to Syslog, but this can be changed (see "Configuring Syslog" on page 358).

Below is an example of a Syslog message:

```
13:10:57.811 : 10.13.4.12 : NOTICE  : [S=235][SID:1034099026] (      lgr flow)(63
)  UdpTransportObject#0- Adding socket event for address 10.33.2.42:5060 [Time: 04-
19-2012@18:29:39]
```

Table 86: Syslog Message Format Description

Message Item	Description
Message Types	Syslog generates the following types of messages: ▪ ERROR: Indicates that a problem has been identified that requires immediate handling. ▪ WARNING: Indicates an error that might occur if measures are not taken to prevent it. ▪ NOTICE: Indicates that an unusual event has occurred. ▪ INFO: Indicates an operational message. ▪ DEBUG: Messages used for debugging. **Notes:** ▪ The INFO and DEBUG messages are required only for advanced debugging. Therefore, by default, they are not sent by the device. ▪ When viewing Syslog messages in the Web interface, these message types are color coded.

Message Sequence Number [S=<number>]	By default, Syslog messages are sequentially numbered in the format [S=<number>], for example, "[S=643]". A skip in the number sequence of messages indicates a loss of message packets. For example, in the below Syslog message, messages 238 through 300 were not received. In other words, 63 Syslog messages were lost (the sequential numbers are indicated below in bold font):
	`18:38:14. 52 : 10.33.45.72 : NOTICE: [`**`S=235`**`][SID:1034099026] (lgr psbrdex)(619) recv <-- DIGIT(0) Ch:0 OnTime:0 InterTime:100 Direction:0 System:1 [File: Line:-1]`
	`18:38:14. 83 : 10.33.45.72 : NOTICE: [`**`S=236`**`][SID:1034099026] (lgr flow)(620)  #0:DIGIT EV [File: Line:-1]`
	`18:38:14. 83 : 10.33.45.72 : NOTICE:`

Message Item	Description

	`[S=237][SID:1034099026] (lgr flow)(621) | #0:DIGIT EV [File: Line:-1]`
	`18:38:14.958 : 10.33.45.72 : NOTICE: [S=301][SID:1034099026] (lgr flow)(625) | #0:DIGIT EV [File: Line:-1]`
	You can disable the inclusion of the message sequence number in Syslog messages, by setting the 'CDR Session ID' parameter to **Disable** (see "Configuring Syslog" on page 358).

Log Number (lgr)(number)	Ignore this number; it has been replaced by the Message Sequence Number (described previously).
Session ID	Automatically assigned (random), unique session identifier (session-id / SID) number per call in the CDR of sent Syslog messages and debug recording packets. This enables you to filter the information (such as SIP, Syslog, and media) according to the SID. ▪ A session is considered as both the outgoing and incoming legs, where both legs share the same SID. The benefit of this unique numbering is that it enables you to filter the information (such as SIP, Syslog, and media) according to a specific SID. **Note:** Forked legs and alternative legs share the same SID.
Message Body	Describes the message.
Timestamp	When the Network Time Protocol (NTP) is enabled, a timestamp string [hour:minutes:seconds] is added to all Syslog messages.

37.1.1 Event Representation in Syslog Messages

The Syslog message events that the device sends are represented by unique abbreviations. An example of an abbreviated event in a Syslog message indicating packet loss (PL) is shown below:

```
Apr  4 12:00:12 172.30.1.14 PL:5  [Code:3a002] [CID:3294] [Time: 20:17:00]
```

The table below lists these unique event abbreviations:

Table 87: Syslog Error Name Descriptions

Error Abbreviation	Error Name Description
AA	Invalid Accumulated Packets Counter
AC	Invalid Channel ID
AL	Invalid Header Length
AO	Invalid Codec Type
AP	Unknown Aggregation Payload Type
AR	Invalid Routing Flag Received
AT	Simple Aggregation Packets Lost
CC	Command Checksum Error
CE	Invalid Cell Coder Code
CS	Command Sequence Error
ES	8 sec Timeout Before Disconnect
HO	Host Received Overrun

Error Abbreviation	Error Name Description
IA	Invalid AMR Payload
IC	Invalid CID Error
IG	Invalid G723 Code
IP	Invalid payload length
IR	Invalid RTCP Packet
IS	Invalid SID Length
LC	Transmitter Received Illegal Command
LF	Lost Fax Frames In High Speed Mode
LM	Lost Modem Frames In High Speed Mode
MI	Misalignment Error
MR	Modem Relay Is Not Supported
OR	DSP JB Overrun
PH	Packet Header Error
PL	RTP Packet Loss
RB	Counts the number of BFI Frames Received From The Host
RD	No Available Release Descriptor
RO	RTP Reorder
RP	Unknown RTP Payload Type
RS	RTP SSRC Error
UF	Unrecognized Fax Relay Command
AA	Invalid Accumulated Packets Counter
AC	Invalid Channel ID
AL	Invalid Header Length
AO	Invalid Codec Type
AP	Unknown Aggregation Payload Type
AR	Invalid Routing Flag Received

37.1.2 Identifying AudioCodes Syslog Messages using Facility Levels

The device's Syslog messages can easily be identified and distinguished from Syslog messages from other equipment, by setting its Facility level. The Facility levels of the device's Syslog messages are numerically coded with decimal values. Facility level may use any of the "local use" facilities (0 through 7), according to RFC 3164. Implementing Facility levels is useful, for example, if you collect the device's as well as other equipments' Syslog messages on the same server. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level.

The Facility level is configured using the SyslogFacility ini file parameter, which provides the following options:

Table 88: Syslog Facility Levels

Numerical Value	Facility Level

Numerical Value	Facility Level
16 (default)	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

Syslog messages begin with a less-than ("**<**") character, followed by a number, which is followed by a greater-than ("**>**") character. This is optionally followed by a single ASCII space. The number is known as the *Priority* and represents both the Facility level and the Severity level. A Syslog message with Facility level 16 is shown below:

```
Facility: LOCAL0 - reserved for local use (16)
```

37.1.3 SNMP Alarms in Syslog Messages

SNMP alerts are sent to the Syslog server using the following formats:

- **Raised Alarms:** RAISE-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >.

  If additional information exists in the alarm, then these are also added: Additional Info1:/ Additional Info2:/ Additional Info3

  The Messages’ Severity is as follows:

Table 89: Syslog Message Severity

ITU Perceived Severity (SNMP Alarm’s Severity)	AudioCodes' Syslog Severity
Critical	RecoverableMsg
Major	RecoverableMsg
Minor	RecoverableMsg
Warning	Notice
Indeterminate	Notice
Cleared	Notice

- **Cleared Alarms:** CLEAR-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >; If exists Additional Info1:/ Additional Info2:/ Additional Info3:

37.2 Configuring Syslog Settings

The procedure below describes how to configure Syslog. This includes the following:

- Enabling Syslog
- Defining the Syslog server

- Determining the level of debug messages that are sent to the Syslog server
- Determining the types of Web actions to report to the Syslog server

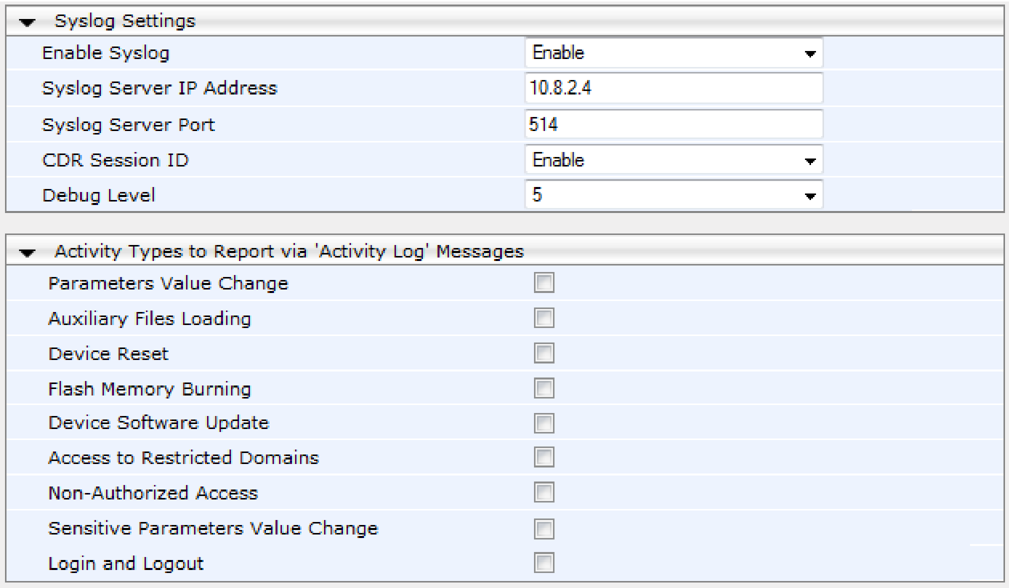
Notes:

- For configuring CDR reporting, see "Configuring CDR Reporting" on page 346.
- For viewing Syslog messages in the Web interface, see "Viewing Syslog Messages" on page 362.
- For a detailed description on the Syslog parameters, see "Syslog, CDR and Debug Parameters" on page 400.

➢ To configure Syslog:

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).

Figure 146: Syslog Settings Page



▼ Syslog Settings

Enable Syslog	Enable ▾
Syslog Server IP Address	10.8.2.4
Syslog Server Port	514
CDR Session ID	Enable ▾
Debug Level	5 ▾

▼ Activity Types to Report via 'Activity Log' Messages

Parameters Value Change	□
Auxiliary Files Loading	□
Device Reset	□
Flash Memory Burning	□
Device Software Update	□
Access to Restricted Domains	□
Non-Authorized Access	□
Sensitive Parameters Value Change	□
Login and Logout	□

2. Enable the Syslog feature by setting 'Enable Syslog' to **Enable**.
3. Define the Syslog server using the 'Syslog Server IP Address' and 'Syslog Server Port' parameters.
4. (Optional) Disable the inclusion of the Sequence Number in Syslog messages by setting the 'CDR Session ID' parameter to **Disable**.
5. Configure the debug level using the 'Debug Level' parameter.
6. Under the Activity Types to Report group, select the Web activities to report.
7. Click **Submit**.

37.3 Configuring Debug Recording

The device enables you to activate debug recording and send debug recording packets to a defined capturing server. When the debug recording is activated, the device duplicates all messages that are sent and/or received by it and then sends them to an external IP address. The debug recording can be done for different types of traffic for example, RTP/RTCP, T.38, and SIP.

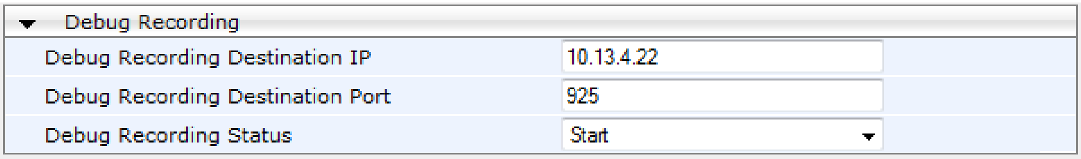
Debug recording is used for advanced debugging when you need to analyze internal messages and signals. Debug recording is also useful for recording network traffic in environments where hub or port mirroring is unavailable and for recording internal traffic between two endpoints on the same device.

Note: Debug recording is collected only on the device's OAMP interface.

➢ To configure and activate debug recording:

1. Open the Logging Settings page (**Configuration** tab > **System** menu > **Logging** > **Logging Settings**).

Figure 147: Logging Settings Page



Debug Recording

Debug Recording Destination IP	10.13.4.22
Debug Recording Destination Port	925
Debug Recording Status	Start

2. Configure the debug capturing server using the 'Debug Recording Destination IP' and 'Debug Recording Destination Port' parameters.
3. From the 'Debug Recording Status' drop-down list, select **Start** to start the debug recording or **Stop** to end the recording.
4. Click **Submit**.

For a detailed description of these parameters, see "Syslog, CDR and Debug Parameters" on page 400.

37.4 Filtering Syslog Messages and Debug Recordings

The device can filter Syslog messages and debug recording (DR) packets, which are sent to a Syslog server and packet capturing application (such as Wireshark), respectively. Filtering can be useful to reduce CPU consumption and minimize negative impact on VoIP performance.

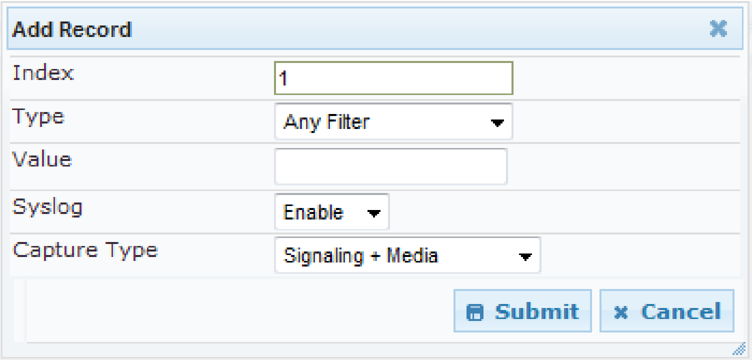
You can configure up to 30 filtering rules, each based on a selected filtering criteria (e.g., an IP Group). Each filtering criteria can be configured with a range. For example, you can filter Syslog messages for IP Groups 1 through 4. For each filter criteria, you can enable or disable Syslog messages and debug recording.

Debug recording can also be filtered using various filtering criteria such as SIP signaling or signaling and media.

➢ To configure a logging filtering rule:

1. Open the Logging Filters Table page (**Configuration** tab > **System** menu > **Logging** > **Logging Filters Table**).
2. Click **Add**; the following dialog box appears:

Figure 148: Logging Filters Table - Add Record Dialog Box



Add Record

Index	1
Type	Any Filter
Value	
Syslog	Enable
Capture Type	Signaling + Media

3. Configure a logging filter according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Note: To configure the Syslog debug level, use the 'Debug Level' parameter (see "Configuring Syslog" on page 358).

The procedure below describes how to configure Logging Filter rules in the Web interface. You can also configure Logging Filter rules using the table ini file parameter, LoggingFilters or the CLI command configure system > logging > logging-filters.

Table 90: Logging Filters Table Parameter Descriptions

Parameter	Description
Index **[LoggingFilters_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Filter Type CLI: filter-type **[LoggingFilters_FilterType]**	Defines the filter type criteria. ▪ **[1]** Any (default) ▪ **[8]** IP Group = Filters according to a specified IP Group ID listed in the IP Group table ▪ [9] SRD = Filters according to a specified SRD ID listed in the SRD table ▪ [10] Classification = Filters according to a specified Classification rule listed in the Classification table ▪ [11] IP-to-IP Routing = Filters according to a specified SBC IP-to-IP routing rule listed in the IP-to-IP Routing table ▪ **[12]** User = Filters according to a specified user defined by username or user@host. ▪ **[13]** IP Trace = Filters according to a specified IP network trace wireshark-like expression. For a detailed description on configuring IP traces, see "Filtering IP Network Traces" on page 362.
Value CLI: value **[LoggingFilters_Value]**	Defines the value of the selected filtering type in the 'Filter Type' parameter. The value can be the following: ▪ A single value ▪ A range, using a hyphen "-" between the two values, e.g., "1-3" ▪ Multiple, non-contiguous values, using commas "," between each value, e.g., "1,3,9" ▪ **Any** to indicate all ▪ For IP trace expressions, see "Filtering IP Network Traces" on page 362
Syslog CLI: syslog **[LoggingFilters_Syslog]**	Enables Syslog messages for the defined logging filter: ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter is not applicable when 'Filter Type' is set to **IP Trace**.
Capture Type CLI: capture-type **[LoggingFilters_CaptureType]**	Enables debug recordings for the defined logging filter and defines what to record: ▪ **[0]** None (default) ▪ **[1]** Signaling = Information related to signaling such as SIP signaling messages, Syslog, CDR, and the device's internal processing messages. ▪ **[2]** Signaling & Media = Signaling and media (RTP/RTCP/T.38). ▪ **[3]** Signaling & Media & PCM = Signaling, media, and PCM **Note:** This parameter is not applicable when 'Filter Type' is set to **IP Trace**.

37.4.1 Filtering IP Network Traces

You can filter Syslog and debug recording messages for IP network traces, by setting the 'Filter Type' parameter to **IP Trace** in the Logging Filters table. IP traces are used to record any IP stream, according to destination and/or source IP address, or port and Layer-4 protocol (UDP, TCP or any other IP type as defined by http://www.iana.com). Network traces are typically used to record HTTP.

When the **IP Trace** option is selected, only the 'Value' parameter is applicable; the 'Syslog' and 'Capture Type' parameters are not relevant. The 'Value' parameter configures the Wireshark-like filtering expressions for your IP trace. The following Wireshark-like expressions are supported:

Table 91: Supported Wireshark-like Expressions for 'Value' Parameter

Expression	Description
ip.src, ip.dst	Source and destination IP address
ip.addr	IP address - up to two IP addresses can be entered
ip.proto	IP protocol type (PDU) entered as an enumeration value (e.g., 1 is ICMP, 6 is TCP, 17 is UDP)
udp, tcp, icmp, sip, ldap, http, https	Single expressions for protocol type
udp.port, tcp.port	Transport layer
udp.srcport, tcp.srcport	Transport layer for source port
udp.dstport, tcp.dstport	Transport layer for destination port
and, &&, ==, <, >	Between expressions

Below are examples of configured expressions for the 'Value' parameter:

- udp && ip.addr==10.8.6.55
- ip.src==10.8.6.55 && udp.port>=5000 and udp.port<6000
- ip.dst==10.8.0.1/16
- ip.addr==10.8.6.40

For conditions requiring the "or" / "||" expression, add multiple table rows. For example, the Wireshark condition "(ip.src == 1.1.1.1 or ip.src == 2.2.2.2) and ip.dst == 3.3.3.3" can be configured using the following two table row entries:

1. ip.src == 1.1.1.1 and ip.dst == 3.3.3.3
2. ip.src == 2.2.2.2 and ip.dst == 3.3.3.3

Note: If the 'Value' field is not defined, the device records all IP traffic types.

37.5 Viewing Syslog Messages

You can use the following tools to view the Syslog messages sent by the device:

- Web interface's Message Log page (see below).
- CLI -The device sends the error messages (e.g. Syslog messages) to the CLI console as well as to the original configured destination. Use the following commands:

```
debug log                  ; Starts the debug
no debug log               ; Stops the debug
no debug log all                   ; Stops all debug process
```

- Any third-party Syslog server (e.g., Wireshark).

The procedure below describes how to view Syslog messages in the Web interface.

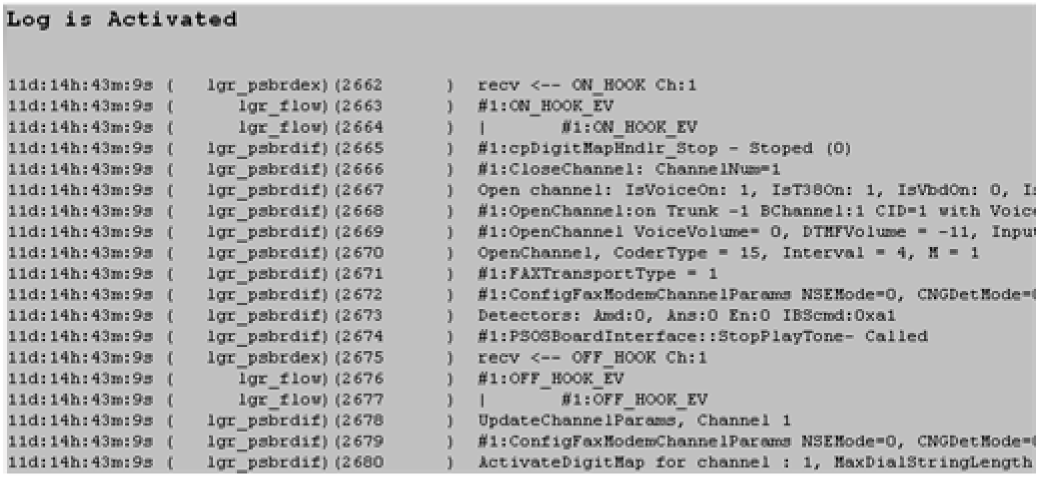
Notes:

- It's not recommended to keep a Message Log session open for a prolonged period. This may cause the device to overload. For prolonged (and detailed) debugging, use an external Syslog server.
- You can select the Syslog messages in this page, and copy and paste them into a text editor such as Notepad. This text file (*txt*) can then be sent to AudioCodes Technical Support for diagnosis and troubleshooting.

➢ To activate the Web interface's Message Log:

1. Enable Syslog (see "Configuring Syslog" on page 358).
2. Open the Message Log page (**Status & Diagnostics** tab > **System Status** menu > **Message Log**); the Message Log page is displayed and the log is activated.

Figure 149: Message Log Page



```
Log is Activated


11d:14h:43m:9s (   lgr_psbrdex)(2662       )  recv <-- ON_HOOK Ch:1
11d:14h:43m:9s (      lgr_flow)(2663       )  #1:ON_HOOK_EV
11d:14h:43m:9s (      lgr_flow)(2664       )  |       #1:ON_HOOK_EV
11d:14h:43m:9s (   lgr_psbrdif)(2665       )  #1:cpDigitMapHndlr_Stop - Stoped (0)
11d:14h:43m:9s (   lgr_psbrdif)(2666       )  #1:CloseChannel: ChannelNum=1
11d:14h:43m:9s (   lgr_psbrdif)(2667       )  Open channel: IsVoiceOn: 1, IsT38On: 1, IsVbdOn: 0, I
11d:14h:43m:9s (   lgr_psbrdif)(2668       )  #1:OpenChannel:on Trunk -1 BChannel:1 CID=1 with Voic
11d:14h:43m:9s (   lgr_psbrdif)(2669       )  #1:OpenChannel VoiceVolume= 0, DTMFVolume = -11, Inpu
11d:14h:43m:9s (   lgr_psbrdif)(2670       )  OpenChannel, CoderType = 15, Interval = 4, M = 1
11d:14h:43m:9s (   lgr_psbrdif)(2671       )  #1:FAXTransportType = 1
11d:14h:43m:9s (   lgr_psbrdif)(2672       )  #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (   lgr_psbrdif)(2673       )  Detectors: Amd:0, Ans:0 En:0 IBScmd:0xa1
11d:14h:43m:9s (   lgr_psbrdif)(2674       )  #1:PSOSBoardInterface::StopPlayTone- Called
11d:14h:43m:9s (   lgr_psbrdex)(2675       )  recv <-- OFF_HOOK Ch:1
11d:14h:43m:9s (      lgr_flow)(2676       )  #1:OFF_HOOK_EV
11d:14h:43m:9s (      lgr_flow)(2677       )  |       #1:OFF_HOOK_EV
11d:14h:43m:9s (   lgr_psbrdif)(2678       )  UpdateChannelParams, Channel 1
11d:14h:43m:9s (   lgr_psbrdif)(2679       )  #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (   lgr_psbrdif)(2680       )  ActivateDigitMap for channel : 1, MaxDialStringLength
```

The displayed logged messages are color-coded as follows:

- Yellow - fatal error message
- Blue - recoverable error message (i.e., non-fatal error)
- Black - notice message

➢ To stop and clear the Message Log:

- Close the Message Log page by accessing any another page in the Web interface.

37.6 Collecting Debug Recording Messages

To collect debug recording packets, the open source program Wireshark is used. AudioCodes proprietary plug-in files for Wireshark, which are shipped in your software kit, are also required.

Notes:

- The default debug recording port is 925. You can change the port in Wireshark (**Edit** menu > **Preferences** > **Protocols** > **AC DR**).
- The plug-ins are per major software release and are applicable to Wireshark Ver. 1.62.
- The plug-ins are backward compatible.
- From Wireshark Ver. 99.08, the tpncp.dat file must be located in the folder, ...WireShark\tpncp.

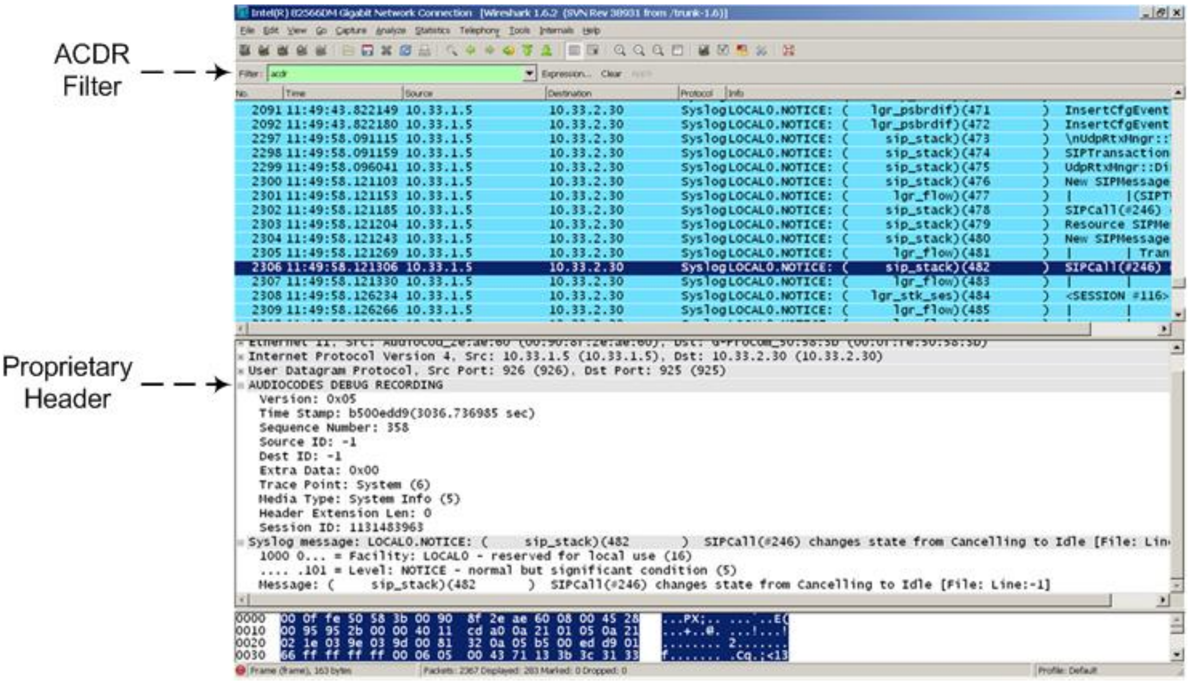
➢ To install Wireshark and the plug-ins for debug recording:

1. Install Wireshark on your computer. The Wireshark program can be downloaded from http://www.wireshark.org.
2. Copy the supplied AudioCodes plug-in files to the directory in which you installed Wireshark, as follows:

Copy this file	To this folder
...\dtds\cdr.dtd	Wireshark\dtds\
...\plugins\1.6.2\*.dll	Wireshark\plugins\1.6.2
...\tpncp\tpncp.dat	Wireshark\tpncp

1. Start Wireshark.
2. In the Filter field, type "acdr" (see the figure below) to view the debug recording messages. Note that the source IP address of the messages is always the OAMP IP address of the device.

The device adds the header "AUDIOCODES DEBUG RECORDING" to each debug recording message, as shown below:


ACDR Filter
Proprietary Header
Intel(R) 82566DM Gigabit Network Connection [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: acdr
Expression... Clear Apply

```
No.  Time             Source      Destination  Protocol Info
2091 11:49:43.822149 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (   lgr_psbrdif)(471      )  InsertCfgEvent
2092 11:49:43.822180 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (   lgr_psbrdif)(472      )  InsertCfgEvent
2297 11:49:58.091115 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(473      )  \nUdpRtxMngr::
2298 11:49:58.091159 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(474      )  SIPTransaction
2299 11:49:58.096041 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(475      )  UdpRtxMngr::Di
2300 11:49:58.121103 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(476      )  New SIPMessage
2301 11:49:58.121153 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(477      )  |          |(SIPT
2302 11:49:58.121185 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(478      )  SIPCall(#246)
2303 11:49:58.121204 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(479      )  Resource SIPMe
2304 11:49:58.121243 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(480      )  New SIPMessage
2305 11:49:58.121269 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(481      )  |          | Tran
2306 11:49:58.121306 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(482      )  SIPCall(#246)
2307 11:49:58.121330 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(483      )  |          |
2308 11:49:58.126234 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (   lgr_stk_ses)(484      )  <SESSION #116>
2309 11:49:58.126266 10.33.1.5   10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(485      )  |          |
```

```
Ethernet II, Src: AudioCod_2e:ae:60 (00:90:8f:2e:ae:60), Dst: G-ProCom_50:58:3b (00:0f:fe:50:58:3b)
Internet Protocol Version 4, Src: 10.33.1.5 (10.33.1.5), Dst: 10.33.2.30 (10.33.2.30)
User Datagram Protocol, Src Port: 926 (926), Dst Port: 925 (925)
AUDIOCODES DEBUG RECORDING
  Version: 0x05
  Time Stamp: b500edd9(3036.736985 sec)
  Sequence Number: 358
  Source ID: -1
  Dest ID: -1
  Extra Data: 0x00
  Trace Point: System (6)
  Media Type: System Info (5)
  Header Extension Len: 0
  Session ID: 1131483963
Syslog message: LOCAL0.NOTICE: (     sip_stack)(482       )  SIPCall(#246) changes state from Cancelling to Idle [File: Lin
  1000 0... = Facility: LOCAL0 - reserved for local use (16)
  .... .101 = Level: NOTICE - normal but significant condition (5)
  Message: (     sip_stack)(482       )  SIPCall(#246) changes state from Cancelling to Idle [File: Line:-1]
```

```
0000  00 0f fe 50 58 3b 00 90  8f 2e ae 60 08 00 45 28   ...PX;.. ...`..E(
0010  00 95 95 2b 00 00 40 11  cd a0 0a 21 01 05 0a 21   ...+..@. ...!...!
0020  02 1e 03 9e 03 9d 00 81  32 0a 05 b5 00 ed d9 01   ........ 2.......
0030  66 ff ff ff ff 00 06 05  00 43 71 13 3b 3c 31 33   f....... .Cq.;<13
```

37.7 Debug Capturing on Physical VoIP Interfaces

You can capture traffic on the device's physical (Ethernet LAN) VoIP interfaces (Layer-2 VLAN tagged packets). The captured traffic can be saved in a PCAP-format file (suitable for Wireshark) to a TFTP (default) or an FTP server. The generated PCAP file is in the Extensible Record Format (ERF). The maximum file size of debug captures that can be saved to the device is 100 MB.

To capture traffic on physical VoIP interfaces, use the following CLI commands:

- Starts physical VoIP debug capture:

```
# debug capture voip physical eth-lan
# debug capture voip physical start
```

- Captures packets continuously in a cyclical buffer (packets always captured until stop command):

```
# debug capture VoIP physical cyclic buffer
```

- Retrieves latest capture (PCAP file) saved on a specified server:

```
# debug capture VoIP physical get_last_capture <TFTP/FTP server IP address>
```

The file is saved to the device's memory (not flash) and erased after a device reset.

- Marks the captured file (useful for troubleshooting process):

```
# debug capture VoIP physical insert-pad
```

Before running this command, the debug capture must be started.

- Displays debug status and configured rules:

```
# debug capture VoIP physical show
```

- Specifies the destination (FTP, TFTP, or USB) where you want the PCAP file sent:

```
# debug capture VoIP physical target <ftp|tftp|usb>
```

- Stops the debug capture, creates a file named debug-capture-voip-<timestamp>.pcap, and sends it to the TFTP or FTP server:

```
# debug capture voip physical stop <TFTP/FTP server IP address>
```

If no IP address is defined, the capture is saved on the device for later retrieval.

38 Enabling Debug File upon Device Crash

You can configure the device to automatically create a debug file upon a device crash. This debug file contains the following information:

- Exception information, indicating the specific point in the code where the crash occurred
- Latest log messages that were recorded prior to the crash
- Core Dump (if enabled) - contains an image of the device's memory at the time of the crash and provides a powerful tool for determining the root cause of the crash. When Core Dump is enabled, the file is saved to the device's flash memory and can also be sent to a user-defined server (IP address).

The Core Dump file is saved with the following name:

- Filename extension: ".log"
- Filename: "debug_<device name>_ver_<firmware version>_mac_<MAC address>_<date>"

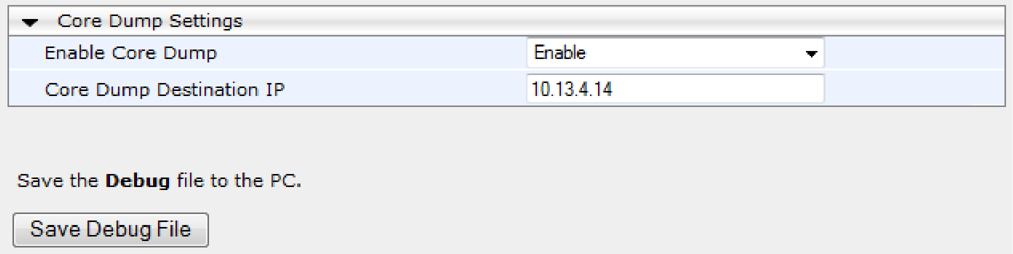
For example: debug_acMediant_ver_680-8-4_mac_00908F099096_1-11-2013_3-29-29[2].log

The Debug file can be retrieved from the device and saved to a folder on a local PC. This file can be sent to AudioCodes support team for troubleshooting.

➢ To configure Core Dump:

1. Open the Debug Utilities page (**Maintenance** tab > **Maintenance** menu > **Debug Utilities**).

Figure 150: Debug Utilities Page



Core Dump Settings

Enable Core Dump: Enable

Core Dump Destination IP: 10.13.4.14

Save the **Debug** file to the PC.

2. From the 'Enable Core Dump' drop-down list, select **Enable**.
3. In the 'Core Dump Destination IP' field, enter an IP address of the remote server to where you want the file to be sent (optional).
4. Click **Submit**.

➢ To save the Core Dump file to the local PC:

- in the Debug Utilities page, click the Save **Debug File** button.

Reader's Notes

39 Testing SIP Signaling Calls

A simulated endpoint can be configured on the device to test SIP signaling of calls between it and a remote destination. This feature is useful in that it can remotely verify SIP message flow without involving the remote end side in the debug process. The SIP test call simulates the SIP signaling process - call setup, SIP 1xx responses, through to completing the SIP transaction with a 200 OK.

The test call sends Syslog messages to a Syslog server, showing the SIP message flow, tone signals (e.g., DTMF), termination reasons, as well as voice quality statistics and thresholds (e.g., MOS).

39.1 Configuring Test Call Endpoints

The Test Call table lets you test the SIP signaling (setup and registration) and media (DTMF signals) of calls between a simulated phone on the device and a remote endpoint. These tests involve both incoming and outgoing calls, where the test endpoint can be configured as the caller or called party. Test calls can be dialed automatically at a user-defined interval and/or manually when required. The simulated phone and remote endpoints are defined as SIP URIs (user@host) and the remote destination can be defined as an IP Group, IP address, or according to an Outbound IP Routing rule. You can also enable automatic registration of the endpoint.

When a SIP test call is initiated, the device generates a SIP INVITE towards the remote endpoint (e.g., a SIP proxy server or softswitch). It simulates the SIP call setup process, managing SIP 1xx responses and completing the SIP transaction with a 200 OK.

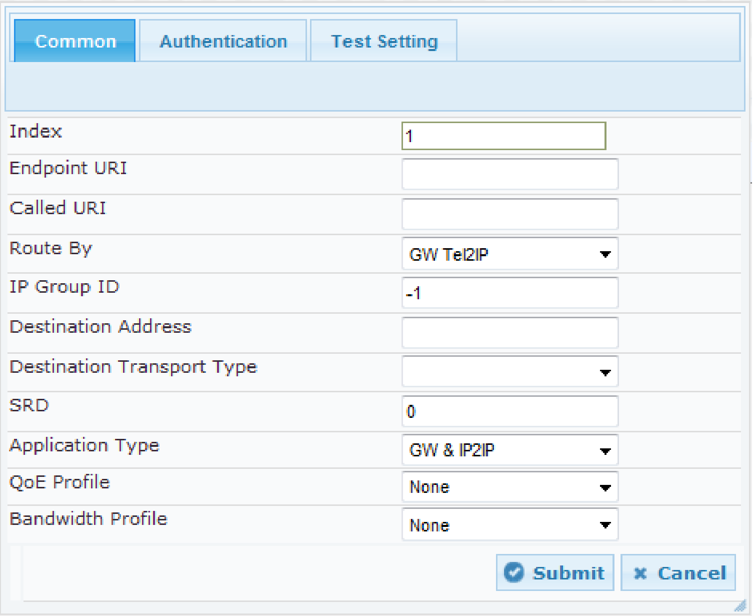
Note: By default, you can configure up to five test calls. However, this number can be increased by installing the relevant Software License Key. For more information, contact your AudioCodes sales representative.

The procedure below describes how to configure test calls in the Web interface. You can also configure this using the table ini file parameter, Test_Call or CLI command, configure system > test-call > test-call-table.

➢ To configure a test call:

1. Open the Test Call Table page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).
2. Click **Add**; the following dialog box appears:

Figure 151: General Tab of Test Call Table



Index	1
Endpoint URI	
Called URI	
Route By	GW Tel2IP
IP Group ID	-1
Destination Address	
Destination Transport Type	
SRD	0
Application Type	GW & IP2IP
QoE Profile	None
Bandwidth Profile	None

3. Configure a test call according to the parameters described in the table below.

4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 92: Test Call Table Parameter Descriptions

Parameter	Description
General Tab	
Endpoint URI CLI: endpoint-uri **[Test_Call_EndpointURI]**	Defines the endpoint's URI. This can be defined as a user or user@host. The device identifies this endpoint only by the URI's user part. The URI's host part is used in the SIP From header in REGISTER requests. The valid value is a string of up to 150 characters. By default, this parameter is not configured.
Called URI CLI: called-uri **[Test_Call_CalledURI]**	Defines the destination (called) URI (user@host). The valid value is a string of up to 150 characters. By default, this parameter is not configured.
Route By CLI: route-by **[Test_Call_DestType]**	Defines the type of routing method. This applies to incoming and outgoing calls. ▪ **[0]** GW Tel2IP = (Default) Calls are matched by (or routed to) an SRD and Application type (defined in the SRD and Application Type parameters below). ▪ **[1]** IP Group = Calls are matched by (or routed to) an IP Group ID. ▪ **[2]** Dest Address = Calls are matched by (or routed to) an SRD and application type. **Notes:** ▪ For REGISTER messages, the option [0] cannot be used as the routing method. ▪ For REGISTER messages, if option [1] is used, only Server-type IP Groups can be used.
IP Group ID CLI: ip-group-id **[Test_Call_IPGroupID]**	Defines the IP Group ID to which the test call is sent or from which it is received. **Notes:** ▪ This parameter is applicable only if option [1] is configured for the 'Route By' parameter. ▪ This IP Group is used for incoming and outgoing calls.
Destination Address CLI: dst-address **[Test_Call_DestAddress]**	Defines the destination host. This can be defined as an IP address[:port] or DNS name[:port]. **Note:** This parameter is applicable only if the 'Route By' parameter is set to [2] (Dest Address).
Destination Transport Type CLI: dst-transport **[Test_Call_DestTransportType]**	Defines the transport type for outgoing calls. ▪ **[-1]** = Not configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** This parameter is applicable only if the 'Route By' parameter is set to [2] (Dest Address).
SRD CLI: srd [Test_Call_SRD]	Defines the SRD for the endpoint. The default is SRD 0. **Note:** This parameter is applicable only if the 'Route By' parameter is set any option except [1] (IP Group).
Application Type CLI: application-type **[Test_Call_ApplicationType]**	Defines the application type for the endpoint. This, in effect, associates the IP Group and SRD to a specific SIP interface. ▪ [0] GW & IP2IP (default) ▪ [2] SBC **Note:** This parameter must always be set to SBC [2].

Parameter	Description
QoE Profile CLI: qoe-profile **[Test_Call_QOEProfile]**	Assigns a QoE Profile to the test call. To configure QoE Profiles, see "Configuring Quality of Experience Profiles" on page 155.
Bandwidth Profile CLI: bandwidth-profile **[Test_Call_BWProfile]**	Assigns a Bandwidth Profile to the test call. To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 159.
Authentication Tab **Note:** These parameters are applicable only if the Call Party parameter is set to **Caller.**	
Auto Register CLI: auto-register **[Test_Call_AutoRegister]**	Enables automatic registration of the endpoint. The endpoint can register to the device itself or to the 'Destination Address' or 'IP Group ID' parameter settings (see above). ▪ **[0]** Disable (default) ▪ **[1]** Enable
Username CLI: user-name **[Test_Call_UserName]**	Defines the authentication username. By default, no username is defined.
Password CLI: password **[Test_Call_Password]**	Defines the authentication password. By default, no password is defined.
Test Settings Tab	
Call Party CLI: call-party **[Test_Call_CallParty]**	Defines whether the test endpoint is the initiator or receiving side of the test call. ▪ **[0]** Caller (default) ▪ **[1]** Called
Maximum Channels for Session CLI: max-channels **[Test_Call_MaxChannels]**	Defines the maximum number of concurrent channels for the test session. For example, if you have configured an endpoint "101" and you set this parameter to "3", the device automatically creates three simulated endpoints - "101", "102" and "103" (i.e., consecutive endpoint URIs are assigned). The default is 1.
Call Duration CLI: call-duration **[Test_Call_CallDuration]**	Defines the call duration (in seconds). The valid value is -1 to 100000. The default is 20. A value of 0 means infinite. A value of -1 means that the parameter value is automatically calculated according to the values of the 'Calls per Second' and 'Maximum Channels for Session' parameters. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Calls per Second CLI: calls-per-second **[Test_Call_CallsPerSecond]**	Defines the number of calls per second. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.

Parameter	Description
Test Mode CLI: test-mode **[Test_Call_TestMode]**	Defines the test session mode. ▪ **[0]** Once = (Default) The test runs until the lowest value between the following is reached: ✓ Maximum channels is reached for the test session, configured by 'Maximum Channels for Session'. ✓ Call duration ('Call Duration') multiplied by calls per second ('Calls per Second'). ✓ Test duration expires, configured by 'Test Duration'. ▪ **[1]** Continuous = The test runs until the configured test duration is reached. If it reaches the maximum channels configured for the test session (in the 'Maximum Channels for Session'), it waits until the configured call duration of a currently established tested call expires before making the next test call. In this way, the test session stays within the configured maximum channels. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Test Duration CLI: test-duration **[Test_Call_TestDuration]**	Defines the test duration (in minutes). The valid value is 0 to 100000. The default is 0 (i.e., unlimited). **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Play CLI: play **[Test_Call_Play]**	Enables and defines the playing of a tone to the answered side of the call. ▪ **[0]** Disable ▪ **[1]** DTMF (default) = Plays a user-defined DTMF string, configured in "Configuring DTMF Tones for Test Calls" on page 373. ▪ **[2]** PRT = Plays a non-DTMF tone from the PRT file (Dial Tone 2). For this option, a PRT file must be loaded to the device (see "Prerecorded Tones File" on page 302). **Notes:** ▪ To configure the DTMF signaling type (e.g., out-of-band or in-band) use the 'DTMF Transport Type' parameter (see Configuring DTMF Transport Types). ▪ This parameter is applicable only if 'Call Party' is set to **Caller**.
Schedule Interval CLI: schedule-interval **[Test_Call_ScheduleInterval]**	Defines the interval (in minutes) between automatic outgoing test calls. The valid value range is 0 to 100000. The default is 0 (i.e., scheduling is disabled). **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.

39.2 Starting and Stopping Test Calls

The procedure below describes how to start, stop, and restart test calls.

➤ To start, stop, and restart a test call:

1. In the Test Call table, select the required test call entry; the **Actions** button appears above the table.
2. From the **Actions** drop-down list, choose the required command:
   - **Dial:** starts the test call (this action is applicable only if the test call party is the caller).
   - **Drop Call:** stops the test call.
   - **Restart:** ends all established calls and then starts the test call session again.

The status of the test call is displayed in the 'Test Status' field of the Test Call table:

- "Idle": test call is not active.
- "Scheduled": test call is planned to run (according to 'Schedule Interval' parameter settings)
- "Running": test call has been started (i.e., the **Dial** command was clicked)

- "Receiving": test call has been automatically activated by calls received for the test call endpoint from the remote endpoint (when all these calls end, the status returns to "Idle")
- "Terminating": test call is in the process of terminating the currently established calls (this occurs if the **Drop Call** command is clicked to stop the test)
- "Done": test call has been successfully completed (or was prematurely stopped by clicking the **Drop Call** command)

A more detailed description of this field is displayed below the table when you click the **Show/Hide** button (see "Viewing Test Call Statistics" on page 372).

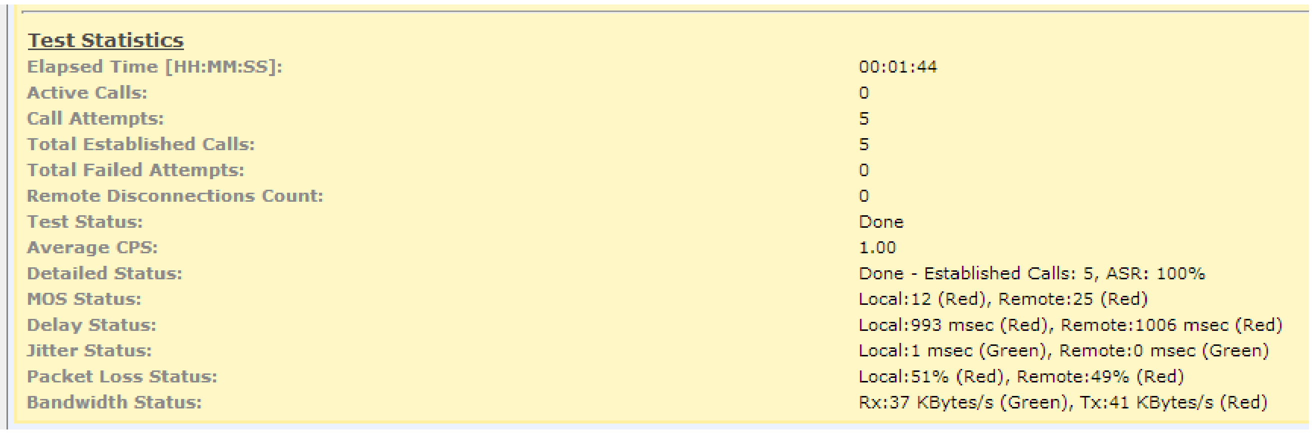
39.3 Viewing Test Call Statistics

In addition to viewing a brief status description of the test call in the 'Test Status' field (as described in "Starting, Stopping and Restarting Test Calls" on page 371), you can also view a more detailed status description which includes test call statistics.

➢ To view statistics of a test call:

1. Open the Test Call Table page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).
2. Select the test call table entry whose call statistics you want to view.
3. Click the **Show/Hide** button; the call statistics are displayed in the **Test Statistics** pane located below the table, as shown below:

Figure 152: Viewing Test Call Statistics



Test Statistics

Elapsed Time [HH:MM:SS]:	00:01:44
Active Calls:	0
Call Attempts:	5
Total Established Calls:	5
Total Failed Attempts:	0
Remote Disconnections Count:	0
Test Status:	Done
Average CPS:	1.00
Detailed Status:	Done - Established Calls: 5, ASR: 100%
MOS Status:	Local:12 (Red), Remote:25 (Red)
Delay Status:	Local:993 msec (Red), Remote:1006 msec (Red)
Jitter Status:	Local:1 msec (Green), Remote:0 msec (Green)
Packet Loss Status:	Local:51% (Red), Remote:49% (Red)
Bandwidth Status:	Rx:37 KBytes/s (Green), Tx:41 KBytes/s (Red)

The 'Test Statistics' pane displays the following test session information:

- **Elapsed Time:** Duration of the test call since it was started (or restarted).
- **Active Calls:** Number of currently established test calls.
- **Call Attempts:** Number of calls that were attempted.
- **Total Established Calls:** Total number of calls that were successfully established.
- **Total Failed Attempts:** Total number of call attempts that failed.
- **Remote Disconnections Count:** Number of calls that were disconnected by the remote side.
- **Average CPS:** Average calls per second.
- **Test Status:** Displays the status (brief description) as displayed in the 'Test Status' field (see "Starting, Stopping and Restarting Test Calls" on page 371).
- **Average CPS:** Average calls per second.
- **Detailed Status:** Displays a detailed description of the test call status:
  - "Idle": test call is currently not active.

- "Scheduled - Established Calls: <number of established calls>, ASR: <%>": test call is planned to run (according to 'Schedule Interval' parameter settings) and also shows the following summary of completed test calls:
    - Total number of test calls that were established.
    - Number of successfully answered calls out of the total number of calls attempted (ASR).
- "Running (Calls: <number of active calls>, ASR: <%>)": test call has been started (i.e., the **Dial** command was clicked) and shows the following:
    - Number of currently active test calls.
    - Number of successfully answered calls out of the total number of calls attempted (Answer Seizure Ratio or ASR).
- "Receiving (<number of active calls>)": test call has been automatically activated by calls received for this configured test call endpoint from the configured remote endpoint. When all these calls terminate, the status returns to "Idle".
- "Terminating (<number of active calls>)": the **Drop Call** command has been clicked to stop the test call and the test call is in the process of terminating the currently active test calls.
- "Done - Established Calls: <number of established calls>, ASR: <%>": test call has been successfully completed (or was prematurely stopped by clicking the **Drop Call** command) and shows the following:
    - Total number of test calls that were established.
    - Number of successfully answered calls out of the total number of calls attempted (ASR).

- **MOS Status:** MOS count and color threshold status of local and remote sides according to the assigned QoE Profile.
- **Delay Status:** Packet delay count and color-threshold status of local and remote sides according to the assigned QoE Profile.
- **Jitter Status:** Jitter count and color-threshold status of local and remote sides according to the assigned QoE Profile.
- **Packet Loss Status:** Packet loss count and color-threshold status of local and remote sides according to the assigned QoE Profile.
- **Bandwidth Status:** Tx/Rx bandwidth and color-threshold status according to the assigned Bandwidth Profile.

Note: On the receiving side, when the first call is accepted in "Idle" state, statistics are reset.

39.4 Configuring DTMF Tones for Test Calls

By default, no DTMF signal is played to an answered test call (incoming or outgoing). However, you can enable this per configured test call in the Test Call table (see "Configuring Test Call Endpoints" on page 368). If enabled, the default DTMF signal that is played is "3212333". You can change this as described below.

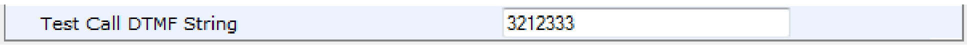
Notes:

- The DTMF signaling type (e.g., out-of-band or in-band) can be configured using the 'DTMF Transport Type' parameter. For more information, see Dual-Tone Multi-Frequency Signaling.
- To generate DTMF tones, the device's DSP resources are required.

➢ To configure the played DTMF signal to answered test call:

1. Open the Test Call Settings page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Settings**).

Figure 153: DTMF in Test Call Settings Page



Test Call DTMF String: 3212333

2. In the 'Test Call DTMF String' field, enter the DTMF string (up to 15 digits).
3. Click **Submit**.

39.5 Configuring SBC Test Call with External Proxy

The SBC Test Call feature tests incoming SBC SIP call flow between a simulated test endpoint on the device and a remote SIP endpoint, when registration and routing is done through an external proxy/registrar server such as a hosted IP PBX in the WAN. In other words, the complete SIP flow, including the path to/from the external proxy/registrar can be tested.

Notes:

- The SBC Test Call feature is initiated only upon receipt of incoming calls and with the configured prefix.
- This call test is done on all SIP interfaces.

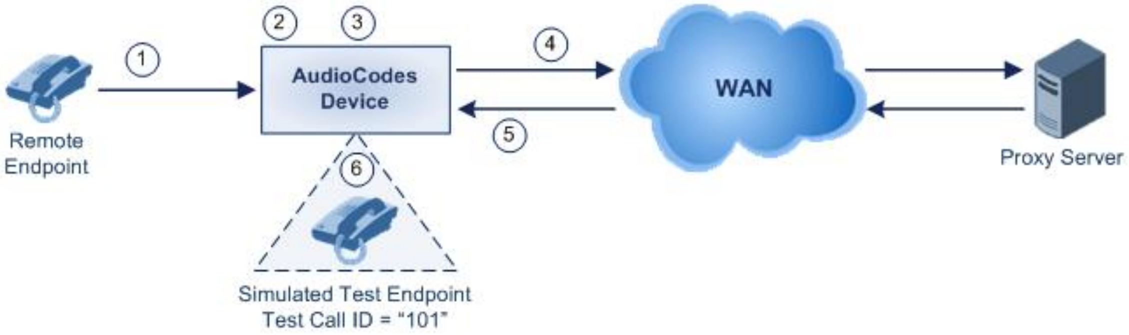
As this test call type involves an SBC call, you need to configure regular SBC rules such as classification and IP-to-IP routing. Therefore, this test call also allows you to verify correct SBC configuration.

For this test call, you also need to configure the following call IDs:

- Test Call ID - prefix number of the simulated endpoint on the device.
- SBC Test ID - prefix number of called number for identifying incoming call as SBC test call. The device removes this prefix, enabling it to route the call according to the IP-to-IP Routing rules to the external proxy/registrar, instead of directly to the simulated endpoint. Only when the device receives the call from the proxy/registrar, does it route the call to the simulated endpoint.

The figure below displays an example of an SBC test call:

Figure 154: SBC Test Call Example



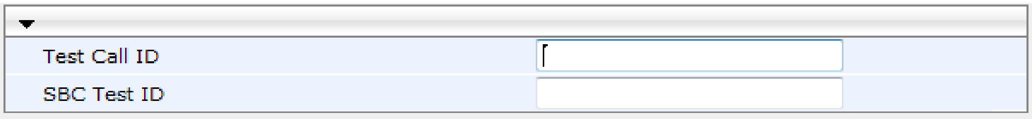
1. The call is received from the remote endpoint with the called number prefix "8101".
2. As the 'SBC Test ID' parameter is set to "8", the device identifies this call as a test call and removes the digit "8" from the called number prefix, leaving it as "101".
3. The device performs the regular SBC processing such as classification and manipulation.
4. The device routes the call, according to the configured SBC IP-to-IP routing rules, to the proxy server.
5. The device receives the call from the proxy server.

6. As the 'Test Call ID' parameter is set to "101", the device identifies the incoming call as a test call and sends it directly to the simulated test endpoint "101".

➢ To configure SBC call testing:

1. Configure the test call parameters (for a full description, see "SIP Test Call Parameters" on page 399):
   a. Open the Test Call Settings page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Settings**).

Figure 155: Test Call Settings Page



Test Call ID	
SBC Test ID	

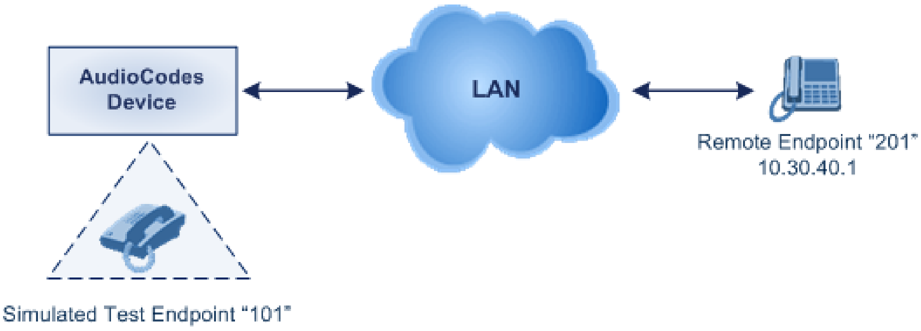
   b. In the 'Test Call ID' field, enter a prefix number for the simulated test endpoint on the device.
   c. In the 'SBC Test ID' field, enter a called prefix number for identifying the call as an SBC test call.
   d. Click **Submit**.
2. Configure regular SBC call processing rules for called number prefix "101", such as classification and IP-to-IP routing through a proxy server.

39.6 Test Call Configuration Examples

Below are a few examples of test call configurations.

- **Single Test Call Scenario:** This example describes the configuration of a simple test call scenario that includes a single test call between a simulated test endpoint on the device and a remote endpoint.

Figure 156: Single Test Call Example

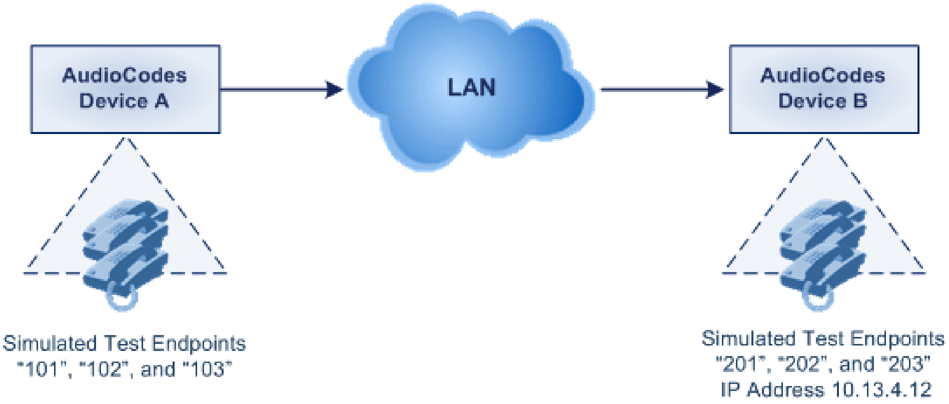


- Test Call table configuration:
  - Endpoint URI: "101"
  - Called URI: "201"
  - Route By: **Dest Address**

- Destination Address: "10.30.40.01"
- Call Party: **Caller**
- Test Mode: **Once**

- **Batch Test Call Scenario:** This example describes the configuration of a batch test call setup for scheduled and continuous call testing of multiple endpoints. The test call is done between two AudioCodes devices - Device A and Device B - with simulated test endpoints. This eliminates the need for phone users, who would otherwise need to answer and end calls many times for batch testing. The calls are initiated from Device A, where Device B serves as the remote answering endpoint.

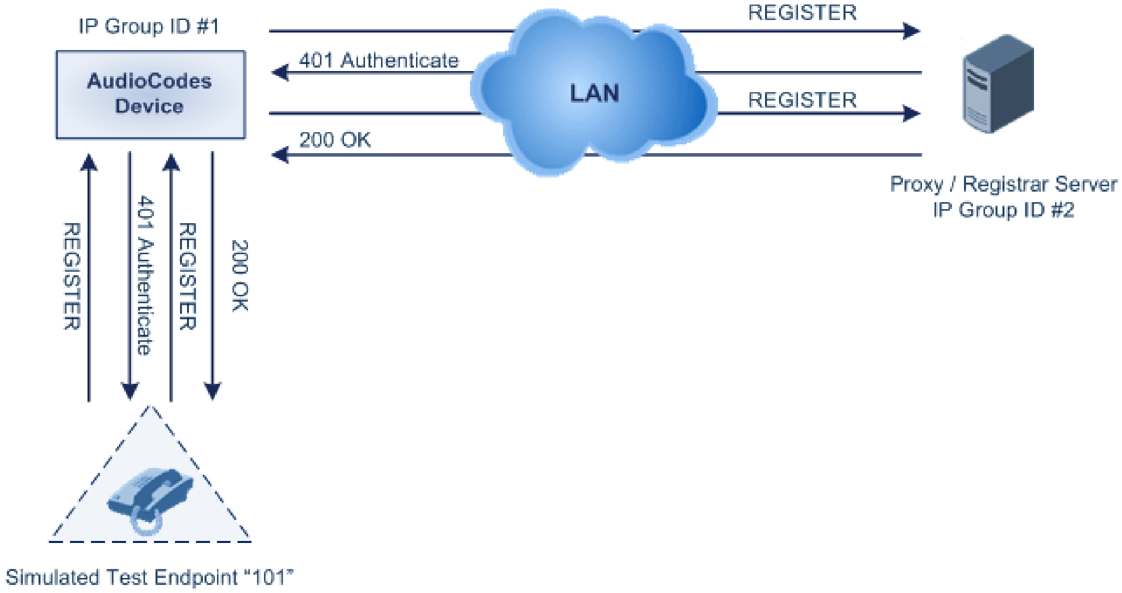
Figure 157: Batch Test Call Example


AudioCodes Device A
LAN
AudioCodes Device B
Simulated Test Endpoints "101", "102", and "103"
Simulated Test Endpoints "201", "202", and "203" IP Address 10.13.4.12

- Test Call table configuration at Device A:
  - Endpoint URI: "101"
  - Called URI: "201"
  - Route By: **Dest Address**
  - Destination Address: "10.13.4.12"
  - Call Party: **Caller**
  - Maximum Channels for Session: "3" (configures three endpoints - "101", "102" and "103)
  - Call Duration: "5" (seconds)
  - Calls per Sec: "1"
  - Test Mode: **Continuous**
  - Test Duration: "3" (minutes)
  - Schedule Interval: "180" (minutes)
- Test Call table configuration at Device B:
  - Endpoint URI: "201"
  - Maximum Channels for Session: "3" (configures three endpoints - "201", "202" and "203)

- **Registration Test Call Scenario:** This example describes the configuration for testing the registration and authentication (i.e., username and pas,sword) process of a simulated test endpoint on the device with an external proxy/registrar server. This is useful, for example, for verifying that endpoints located in the LAN can register with an external proxy and subsequently, communicate with one another.

Figure 158: Test Call Registration Example



This example assumes that you have configured your device for communication between LAN phone users such as IP Groups to represent the device (10.13.4.12) and the proxy server, and IP-to-IP routing rules to route calls between these IP Groups.

- Test Call table configuration:
    - Endpoint URI: "101"
    - Called URI: "itsp"
    - Route By: **Dest Address**
    - Destination Address: "10.13.4.12" (this is the IP address of the device itself)
    - Auto Register: **Enable**
    - User Name: "testuser"
    - Password: "12345"
    - Call Party: **Caller**

Part X

Appendix

40 Dialing Plan Notation for Routing and Manipulation

The device supports flexible dialing plan notations for denoting the prefix and/or suffix source and/or destination numbers and SIP URI user names in the routing and manipulation tables.

Table 93: Dialing Plan Notations for Prefixes and Suffixes

Notation	Description
x (letter "x")	Wildcard that denotes any single digit or character.
# (pound symbol)	▪ When used at the end of a prefix, it denotes the end of a number. For example, **54324#** represents a 7-digit number that starts with the digits 54324. ▪ When used anywhere else in the number (not at the end), it is part of the number (pound key). For example, **3#45** represents the prefix number 3#45. ▪ To denote the pound key when it appears at the end of the number, the pound key must be enclosed in square brackets. For example, 134[#] represents any number that starts with 134#.
***** (asterisk symbol)	▪ When used on its own, it denotes any number or string. ▪ When used as part of a number, it denotes the asterisk key. For example, *345 represents a number that starts with *345.
$ (dollar sign)	Denotes an empty prefix for incoming IP calls that do not have a user part in the Request-URI, or for incoming Tel calls that do not have a called or calling number. This is used for the following matching criteria: ▪ Source and Destination Phone Prefix ▪ Source and Destination Username ▪ Source and Destination Calling Name Prefix
Range of Digits **Notes:** ▪ Dial plans denoting a prefix that is a range must be enclosed in square brackets, e.g., **[4-8]** or **23xx[456]**. ▪ Dial plans denoting a prefix that is not a range is not enclosed, e.g., **12345#**. ▪ Dial plans denoting a suffix must be enclosed in parenthesis, e.g., **(4)** and **(4-8)**. ▪ Dial plans denoting a suffix that include multiple ranges, the range must be enclosed in square brackets, e.g., **(23xx[4,5,6])**. ▪ An example for entering a combined prefix and suffix dial plan - assume you want to match a rule whose destination phone prefix is 4 to 8, and suffix is 234, 235, or 236. The entered value would be the following: **[4-8](23[4,5,6])**.	
[n-m] or **(n-m)**	Represents a range of numbers. Examples: ▪ To depict prefix numbers from 5551200 to 5551300: ✓ **[5551200-5551300]#** ▪ To depict prefix numbers from 123100 to 123200: ✓ **123[100-200]#** ▪ To depict prefix and suffix numbers together: ✓ 03(100): for any number that starts with 03 and ends with 100. ✓ [100-199](100,101,105): for a number that starts with 100 to 199 and ends with 100, 101 or 105. ✓ 03(abc): for any number that starts with 03 and ends with abc. ✓ 03(5xx): for any number that starts with 03 and ends with 5xx. ✓ 03(400,401,405): for any number that starts with 03 and ends with 400 or 401 or 405. **Notes:** ▪ The value *n* must be less than the value *m*. ▪ Only numerical ranges are supported (not alphabetical letters).

Notation	Description
	▪ For suffix ranges, the starting (*n*) and ending (*m*) numbers in the range must include the same number of digits. For example, (23-34) is correct, but (3-12) is not.
[n,m,...] or **(n,m,...)**	Represents multiple numbers. The value can include digits or characters. Examples: ▪ To depict a one-digit number starting with 2, 3, 4, 5, or 6: **[2,3,4,5,6]** ▪ To depict a one-digit number ending with 7, 8, or 9: **(7,8,9)** ▪ Prefix with Suffix: **[2,3,4,5,6](7,8,9)** - prefix is denoted in square brackets; suffix in parenthesis For **prefix only**, the notations *d[n,m]e* and *d[n-m]e* can also be used: ▪ To depict a five-digit number that starts with 11, 22, or 33: **[11,22,33]xxx#** ▪ To depict a six-digit number that starts with 111 or 222: **[111,222]xxx#**
[n1-m1,n2-m2,a,b,c,n3-m3] or **(n1-m1,n2-m2,a,b,c,n3-m3)**	Represents a mixed notation of single numbers and multiple ranges. For example, to depict numbers 123 to 130, 455, 766, and 780 to 790: ▪ Prefix: **[123-130,455,766,780-790]** ▪ Suffix: **(123-130,455,766,780-790)** **Note:** The ranges and the single numbers used in the dial plan must have the same number of digits. For example, each number range and single number in the dialing plan example above consists of three digits.

Note: When configuring phone numbers or prefixes in the Web interface, enter them only as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

41 Configuration Parameters Reference

The device's configuration parameters, default values, and their descriptions are documented in this section.

Note: Parameters and values enclosed in square brackets [...] represent the *ini* file parameters and their enumeration values.

41.1 Management Parameters

This section describes the device's management-related parameters.

41.1.1 General Parameters

The general management parameters are described in the table below.

Table 94: General Management Parameters

Parameter	Description
WAN OAMP Interface CLI: bind GigabitEthernet <slot/port.vlanId> oamp [OAMPWanInterfaceName]	Binds the OAMP interface to a WAN interface, which can later be associated with a Virtual Routing and Forwarding (VRF).
Web: Allow WAN access to HTTP CLI: wan-http-allow [AllowWanHTTP]	Enables WAN access to the management interface through HTTP. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to HTTPS CLI: wan-https-allow [AllowWanHTTPS]	Enables WAN access to the management interface through HTTPS. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to SNMP CLI: wan-snmp-allow [AllowWanSNMP]	Enables WAN access to the management interface through SNMP. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to Telnet CLI: wan-telnet-allow [AllowWanTelnet	Enables WAN access to the management interface through Telnet. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to SSH CLI: wan-ssh-allow [AllowWanSSH]	Enables WAN access to the management interface through SSH. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: Web and Telnet Access List Table **[WebAccessList_x]**	This table configures up to ten IP addresses that are permitted to access the device's Web interface and Telnet interfaces. Access from an undefined IP address is denied. When no IP addresses are defined in this table, this security feature is inactive (i.e., the device can be accessed from any IP address). The default is 0.0.0.0 (i.e., the device can be accessed from any IP address). For example: WebAccessList_0 = 10.13.2.66 WebAccessList_1 = 10.13.77.7 For a description of this parameter, see "Configuring Web and Telnet Access List" on page 51.
Web: Product Key CLI: configure system > product-key [ProductKey]	Defines the device's Product Key. The valid value is a string of up to 40 characters.

41.1.2 Web Parameters

The Web parameters are described in the table below.

Table 95: Web Parameters

Parameter	Description
Web: Enable web access from all interfaces CLI: web-access-from-all-interfaces [EnableWebAccessFromAllInterfaces]	Enables Web access from any of the device's IP network interfaces. This feature applies to HTTP and HTTPS protocols. ▪ [0] = (Default) Disable – Web access is only through the OAMP interface. ▪ [1] = Enable - Web access is through any network interface.
Web: Password Change Interval **[WebUserPassChangeInterval]**	Defines the duration (in minutes) of the validity of Web login passwords. When this duration expires, the password of the Web user must be changed. The valid value is 0 to 100000, where 0 means that the password is always valid. The default is 1140. **Note:** This parameter is applicable only when using the Web Users table, where the default value of the 'Password Age' parameter in the Web Users table inherits this parameter's value.
Web: User inactivity timer **[UserInactivityTimer]**	Defines the duration (in days) for which a user has not logged in to the Web interface, after which the status of the user becomes inactive and can no longer access the Web interface. These users can only log in to the Web interface if their status is changed (to New or Valid) by a System Administrator or Master user. The valid value is 0 to 10000, where 0 means inactive. The default is 90. **Note:** This parameter is applicable only when using the Web Users table.
Web: Session Timeout **[WebSessionTimeout]**	Defines the duration (in minutes) of Web inactivity of a logged-in user, after which the user is automatically logged off the Web interface. The valid value is 0-100000, where 0 means no timeout. The default is 15. **Note:** This parameter can apply to all users, or per user when set in the Web Users table.
Web: Deny Access On Fail Count	Defines the maximum number of failed login attempts, after which the

Parameter	Description
[DenyAccessOnFailCount]	requesting IP address is blocked. The valid value range is 0 to 10. The values 0 and 1 mean immediate block. The default is 3.
Web: Deny Authentication Timer **[DenyAuthenticationTimer]**	Defines the duration (in seconds) for which login to the Web interface is denied from a specific IP address (for all users) when the number of failed login attempts has exceeded the maximum. This maximum is defined by the DenyAccessOnFailCount parameter. Only after this time expires can users attempt to login from this same IP address. The valid value is 0 to 100000, where 0 means that login is not denied regardless of number of failed login attempts. The default is 60.
Web: Display Login Information **[DisplayLoginInformation]**	Enables display of user's login information on each successful login attempt. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
[EnableMgmtTwoFactorAuthentication]	Enables Web login authentication using a third-party, smart card. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When enabled, the device retrieves the Web user's login username from the smart card, which is automatically displayed (read-only) in the Web Login screen; the user is then required to provide only the login password. Typically, a TLS connection is established between the smart card and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username. Thus, this feature implements a two-factor authentication - what the user has (the physical card) and what the user knows (i.e., the login password).
CLI: http-port **[HTTPport]**	Defines the LAN HTTP port for Web management (default is 80). To enable Web management from the LAN, configure the desired port. **Note:** For this parameter to take effect, a device reset is required.
[DisableWebConfig]	Determines whether the entire Web interface is read-only. ▪ **[0]** = (Default) Enables modifications of parameters. ▪ **[1]** = Web interface is read-only. When in read-only mode, parameters can't be modified. In addition, the following pages can't be accessed: 'Web User Accounts', 'Certificates', 'Regional Settings', 'Maintenance Actions' and all file-loading pages ('Load Auxiliary Files', 'Software Upgrade Wizard', and 'Configuration File'). **Note:** For this parameter to take effect, a device reset is required.
[ResetWebPassword]	Resets the username and password of the primary ("Admin") and secondary ("User") accounts to their default settings ("Admin" and "Admin" respectively), and deletes all other users that may have been configured. ▪ **[0]** = (Default) Password and username retain their values. ▪ **[1]** = Password and username are reset. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ You cannot reset the username and password through the Web interface (by loading an ini file or on the AdminPage). To reset the username and password: ✓ SNMP: 1) Set acSysGenericINILine to WEBPasswordControlViaSNMP =

Parameter	Description
	1, and reset the device with a flash burn (set acSysActionSetResetControl to 1 and acSysActionSetReset to 1). 2) Change the username and password in the acSysWEBAccessEntry table. Use the following format: Username acSysWEBAccessUserName: old/pass/new Password acSysWEBAccessUserCode: username/old/new
[WelcomeMessage]	Enables and defines a Welcome message that appears on the Web Login page for logging in to the Web interface. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text [\WelcomeMessage] For Example: FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "***********************************" ; WelcomeMessage 2 = "********* This is a Welcome message ***" ; WelcomeMessage 3 = "***********************************" ; **Notes:** ▪ Each index row represents a line of text in the Welcome message box. Up to 20 lines (or rows) of text can be defined. ▪ The configured text message must be enclosed in double quotation marks (i.e., "..."). ▪ If this parameter is not configured, no Welcome message is displayed.
Web: HA Device Name [HAUnitIdName]	Defines a name for the device, which is displayed on the Home page to indicate the active device. The valid value is a string of up to 128 characters. For the default value, the device assigns either "Device 1" or "Device 2", so that active and redundant devices have different default names.

41.1.3 Telnet Parameters

The Telnet parameters are described in the table below.

Table 96: Telnet Parameters

Parameter	Description
Web: Embedded Telnet Server CLI: telnet **[TelnetServerEnable]**	Enables the device's embedded Telnet server. Telnet is disabled by default for security. ▪ **[0]** Disable ▪ **[1]** Enable Unsecured (default) ▪ **[2]** Enable Secured **Note:** Only the primary Web User Account (which has Security Administration access level) can access the device using Telnet (see "Configuring Web User Accounts" on page 42).
Web: Telnet Server TCP Port CLI: telnet-port **[TelnetServerPort]**	Defines the port number for the embedded Telnet server. The valid range is all valid port numbers. The default port is 23.
Web: Telnet Server Idle	Defines the timeout (in minutes) for disconnection of an idle Telnet session. When

Parameter	Description
Timeout CLI: idle-timeout **[TelnetServerIdleDisconnect]**	set to zero, idle sessions are not disconnected. The valid range is any value. The default is 0. **Note:** For this parameter to take effect, a device reset is required.
Web: Maximum Telnet Sessions CLI: telnet-max-sessions **[TelnetMaxSessions]**	Defines the maximum number of permitted, concurrent Telnet/SSH sessions. The valid range is 1 to 5 sessions. The default is 2. **Note:** Before changing the value, make sure that not more than this number of sessions are currently active; otherwise, the new setting will not take effect.

41.1.4 ini File Parameters

The parameters relating to ini-file management are described in the table below.

Table 97: ini File Parameters

Parameter	Description
[INIPasswordsDisplayType]	Defines how passwords are displayed in the ini file. ▪ **[0]** Disable (default) = Passwords are obscured ("encoded"). The passwords are displayed in the following syntax: $1$<obscured password> (e.g., $1$S3p+fno=). ▪ **[1]** Enable = All passwords are hidden and replaced by an asterisk (*).

41.1.5 SNMP Parameters

The SNMP parameters are described in the table below.

Table 98: SNMP Parameters

Parameter	Description
Web: Enable SNMP CLI: disable **[DisableSNMP]**	Enables SNMP. ▪ **[0]** Enable (default) ▪ **[1]** Disable = SNMP is disabled and no traps are sent.
CLI: port **[SNMPPort]**	Defines the device's local (LAN) UDP port used for SNMP Get/Set commands. The range is 100 to 3999. The default port is 161. **Note:** For this parameter to take effect, a device reset is required.
[KeepAliveTrapPort]	Defines the port to which keep-alive traps are sent. The valid range is 0 - 65534. The default is port 162.
[SendKeepAliveTrap]	Enables keep-alive traps and sends them every 9/10 of the time as defined by the NATBindingDefaultTimeout parameter. ▪ **[0]** = Disable ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
CLI: sys-oid **[SNMPSysOid]**	Defines the base product system OID. The default is eSNMP_AC_PRODUCT_BASE_OID_D. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[SNMPTrapEnterpriseOid]	Defines the Trap Enterprise OID. The default is eSNMP_AC_ENTERPRISE_OID. The inner shift of the trap in the AcTrap subtree is added to the end of the OID in this parameter. **Note:** For this parameter to take effect, a device reset is required.
[acUserInputAlarmDescription]	Defines the description of the input alarm.
[acUserInputAlarmSeverity]	Defines the severity of the input alarm.
[AlarmHistoryTableMaxSize]	Defines the maximum number of rows in the Alarm History table. This parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB). The valid range is 50 to 1000. The default is 500. **Note:** For this parameter to take effect, a device reset is required.
CLI: engine-id **[SNMPEngineIDString]**	Defines the SNMP engine ID for SNMPv2/SNMPv3 agents. This is used for authenticating a user attempting to access the SNMP agent on the device. The ID can be a string of up to 36 characters. The default is 00:00:00:00:00:00:00:00:00:00:00:00 (12 Hex octets characters). The provided key must be set with 12 Hex values delimited by a colon (":") in the format xx:xx:...:xx. For example, 00:11:22:33:44:55:66:77:88:99:aa:bb **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Before setting this parameter, all SNMPv3 users must be deleted; otherwise, the parameter setting is ignored. ▪ If the supplied key does not pass validation of the 12 Hex values input or it is set with the default value, the engine ID is generated according to RFC 3411.
Web: SNMP Trap Destination Parameters CLI: configure system/snmp trap destination **Note:** Up to five SNMP trap managers can be defined.	
SNMP Manager **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled
Web: IP Address CLI: ip-address **[SNMPManagerTableIP_x]**	Defines the IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to this IP address. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Web: Trap Port CLI: port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port. The valid SNMP trap port range is 100 to 4000. The default port is 162.
Web: Trap Enable CLI: send-trap **[SNMPManagerTrapSendingEnable_x]**	Enables the sending of traps to the corresponding SNMP manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = (Default) Sending is enabled.
Web: Trap User CLI: trap-user **[SNMPManagerTrapUser_x]**	Defines the SNMPv3 USM user or SNMPv2 user to associate with the trap destination. This determines the trap format, authentication level, and encryption level. By default, it is associated with the SNMPv2 user (SNMP trap community string). The valid value is a string.
Web: Trap Manager Host Name CLI: manager-host-name **[SNMPTrapManagerHostName]**	Defines an FQDN of the remote host used as an SNMP manager. The resolved IP address replaces the last entry in the Trap Manager table (defined by the SNMPManagerTableIP parameter) and the last trap manager entry of

Parameter	Description
	snmpTargetAddrTable in the snmpTargetMIB. For example: 'mngr.corp.mycompany.com'. The valid range is a string of up to 99 characters.
SNMP Community String Parameters	
Community String CLI: ro-community-string **[SNMPReadOnlyCommunityString_x]**	Defines up to five read-only SNMP community strings (up to 19 characters each). The default string is 'public'.
Community String CLI: rw-community-string **[SNMPReadWriteCommunityString_x]**	Defines up to five read/write SNMP community strings (up to 19 characters each). The default string is 'private'.
Trap Community String CLI: community-string **[SNMPTrapCommunityString]**	Defines the Community string used in traps (up to 19 characters). The default string is 'trapuser'.
SNMP Trusted Managers Table	
Web: SNMP Trusted Managers CLI: configure system > snmp > trusted-managers **[SNMPTrustedMgr_x]**	Defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. **Notes:** ▪ By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests. ▪ If no values are assigned to these parameters any manager can access the device. ▪ Trusted managers can work with all community strings.
SNMP V3 Users Table	
Web: SNMP V3 Users CLI: configure system > snmp v3-users **[SNMPUsers]**	This *para*meter table defines SNMP v3 users. The format of this parameter is as follows: [SNMPUsers] FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group; [\SNMPUsers] For example: SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1; The example above configures user 'v3admin1' with security level authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2. For a description of this table, see "Configuring SNMP V3 Users" on page 68.

41.1.6 Serial Parameters

The RS-232 serial parameters are described in the table below.

Table 99: Serial Parameters

Parameter	Description
[DisableRS232]	Enables the device's RS-232 (serial) port.

Parameter	Description
	▪ **[0]** = Enabled ▪ **[1]** = (Default) Disabled The RS-232 serial port can be used to change the networking parameters and view error/notification messages. For how to establish a serial communication with the device, refer to the *Installation Manual*. **Note:** For this parameter to take effect, a device reset is required.
[SerialBaudRate]	Defines the RS-232 baud rate. The valid values include the following: 1200, 2400, 9600, 14400, 19200, 38400, 57600, or 115200 (default). **Note:** For this parameter to take effect, a device reset is required.
[SerialData]	Defines the RS-232 data bit. ▪ **[7]** = 7-bit ▪ **[8]** = (Default) 8-bit **Note:** For this parameter to take effect, a device reset is required.
[SerialParity]	Defines the RS-232 polarity. ▪ **[0]** = (Default) None ▪ **[1]** = Odd ▪ **[2]** = Even **Note:** For this parameter to take effect, a device reset is required.
[SerialStop]	Defines the RS-232 stop bit. ▪ **[1]** = (Default) 1-bit (default) ▪ **[2]** = 2-bit **Note:** For this parameter to take effect, a device reset is required.
[SerialFlowControl]	Defines the RS-232 flow control. ▪ **[0]** = (Default) None ▪ **[1]** = Hardware **Note:** For this parameter to take effect, a device reset is required.

41.1.7 Auxiliary and Configuration File Name Parameters

The configuration files (i.e., auxiliary files) can be loaded to the device using the Web interface. For loading these files using the *ini* file, you need to configure these files in the *ini* file and configured whether they must be stored in the non-volatile memory. The table below lists the *ini* file parameters associated with these auxiliary files. For more information on the auxiliary files, see "Loading Auxiliary Files" on page 299.

Table 100: Auxiliary and Configuration File Parameters

Parameter	Description
General Parameters	
[SetDefaultOnIniFileProcess]	Determines if all the device's parameters are set to their defaults before processing the updated *ini* file. ▪ **[0]** = Disable - parameters not included in the downloaded *ini* file are not returned to default settings (i.e., retain their current settings). ▪ **[1]** = Enable (default). **Note:** This parameter is applicable only for automatic HTTP update or Web *ini* file upload (not applicable if the *ini* file is loaded using BootP).

Parameter	Description
[SaveConfiguration]	Determines if the device's configuration (parameters and files) is saved to flash (non-volatile memory). - **[0]** = Configuration isn't saved to flash memory. - **[1]** = (Default) Configuration is saved to flash memory.
Auxiliary and Configuration File Name Parameters	
Web: Call Progress Tones File **[CallProgressTonesFilename]**	Defines the name of the file containing the Call Progress Tones definitions. For more information on how to create and load this file, refer to *DConvert Utility User's Guide*. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Prerecorded Tones File **[PrerecordedTonesFileName]**	Defines the name (and path) of the file containing the Prerecorded Tones. **Note:** For this parameter to take effect, a device reset is required.
Web: Dial Plan File EMS: Dial Plan File Name **[DialPlanFileName]**	Defines the name (and path) of the Dial Plan file. This file should be created using AudioCodes DConvert utility (refer to *DConvert Utility User's Guide*).
[UserInfoFileName]	Defines the name (and path) of the file containing the User Information data.

41.1.8 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

Table 101: Automatic Update of Software and Configuration Files Parameters

Parameter	Description
General Automatic Update Parameters	
[AutoUpdateCmpFile]	Enables the Automatic Update mechanism for the cmp file. - **[0]** = (Default) The Automatic Update mechanism doesn't apply to the cmp file. - **[1]** = The Automatic Update mechanism includes the cmp file. **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdateFrequency]	Defines the number of minutes that the device waits between automatic updates. The default is 0 (i.e., the update at fixed intervals mechanism is disabled). **Note:** For this parameter to take effect, a device reset is required.
[AutoUpdatePredefinedTime]	Defines schedules (time of day) for automatic updates. The format of this parameter is: 'HH:MM', where *HH* denotes the hour and *MM* the minutes, for example, 20:18. **Notes:** - For this parameter to take effect, a device reset is required. - The actual update time is randomized by five minutes to reduce the load on the Web servers.
CLI: automatic-update > http-user-agent **[AupdHttpUserAgent]**	Defines the User-Agent HTTP header in the HTTP Get requests sent by the device to the provisioning server for the Automatic Update mechanism. The valid value is a string of up to 511 characters. By default, this parameter is not defined. In other words, the User-Agent header is set to "Mozilla/4.0 (compatible; AudioCodes; <NAME>;<VER>)", where the tags are replaced with actual values. **Notes:** - The variable tags are case-sensitive.

Parameter	Description
	- The tags can be defined in any order. - The tags must be defined adjacent to one another (i.e., no spaces or special characters).
CLI: automatic-update > auto-firmware **[AutoCmpFileUrl]**	Defines the filename and path (URL) to the provisioning server from where the software file (.cmp) can be downloaded, based on timestamp for the Automatic Updated mechanism. The valid value is an IP address in dotted-decimal notation or an FQDN.
EMS: AUPD Verify Certificates CLI: system/tls/aupd-verify-cert **[AUPDVerifyCertificates]**	Determines whether the Automatic Update mechanism verifies server certificates when using HTTPS. - **[0]** = Disable (default) - **[1]** = Enable
[AUPDCheckIfIniChanged]	Determines whether the Automatic Update mechanism performs CRC checking to determine if the configuration file has changed prior to processing. - **[0]** = (Default) Do not check CRC. The file is loaded whenever the server provides it. - **[1]** = Check CRC for the entire file. Any change, including line order, causes the file to be re-processed. - **[2]** = Check CRC for individual lines. Use this option when the HTTP server scrambles the order of lines in the provided file.
CLI: config-system > automatic-update tftp-block-size **[AUPDTftpBlockSize]**	Defines the size of the TFTP data blocks (packets) when downloading a file from a TFTP server for the Automatic Update mechanism. This is in accordance to RFC 2348. TFTP block size is the physical packet size (in bytes) that a network can transmit. When configured to a value higher than the default (512 bytes), but lower than the client network's Maximum Transmission Unit (MTU), the file download speed can be significantly increased. The valid value is 512 to 8192. The default is 512. **Notes:** - A higher value does not necessarily mean better performance. - The block size should be small enough to avoid IP fragmentation in the client network (i.e., below MTU). - This feature is applicable only to TFTP servers that support this option.
[ResetNow]	Invokes an immediate device reset. This option can be used to activate offline (i.e., not on-the-fly) parameters that are loaded using the parameter IniFileUrl. - **[0]** = (Default) The immediate restart mechanism is disabled. - **[1]** = The device immediately resets after an *ini* file with this parameter set to 1 is loaded.
Software/Configuration File URL Path for Automatic Update Parameters	
CLI: firmware **[CmpFileURL]**	Defines the name of the *cmp* file and the path to the server (IP address or FQDN) from where the device can load the *cmp* file and update itself. The *cmp* file can be loaded using HTTP/HTTPS. For example: http://192.168.0.1/filename **Notes:** - For this parameter to take effect, a device reset is required. - When this parameter is configured, the device always loads the *cmp* file after it is reset. - The *cmp* file is validated before it's burned to flash. The checksum of the *cmp* file is also compared to the previously burnt checksum to avoid unnecessary resets.

Parameter	Description
	▪ The maximum length of the URL address is 255 characters.
CLI: voice-configuration **[IniFileURL]**	Defines the name of the *ini* file and the path to the server (IP address or FQDN) on which it is located. The *ini* file can be loaded using HTTP/HTTPS. For example: http://192.168.0.1/filename http://192.8.77.13/config<MAC> https://<username>:<password>@<IP address>/<file name> **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When using HTTP or HTTPS, the date and time of the *ini* file are validated. Only more recently dated *ini* files are loaded. ▪ The optional string <MAC> is replaced with the device's MAC address. Therefore, the device requests an *ini* file name that contains its MAC address. This option allows the loading of specific configurations for specific devices. ▪ The maximum length of the URL address is 99 characters.
CLI: prerecorded-tones **[PrtFileURL]**	Defines the name of the Prerecorded Tones (PRT) file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
CLI:call-progress-tones **[CptFileURL]**	Defines the name of the CPT file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
CLI: tls-root-cert **[TLSRootFileUrl]**	Defines the name of the TLS trusted root certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
CLI: tls-cert **[TLSCertFileUrl]**	Defines the name of the TLS certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
CLI: tls-private-key **[TLSPkeyFileUrl]**	Defines the URL for downloading a TLS private key file using the Automatic Update facility.
[UserInfoFileURL]	Defines the name of the User Information file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file **Note:** The maximum length of the URL address is 99 characters.

41.2 Networking Parameters

This subsection describes the device's networking parameters.

41.2.1 Ethernet Parameters

The Ethernet parameters are described in the table below.

Table 102: Ethernet Parameters

Parameter	Description

Parameter	Description
Physical Ports Settings Table	
Web: Physical Ports Settings CLI: configure voip/physical-port [PhysicalPortsTable]	This table parameter configures the physical Ethernet ports The format of this parameter is as follows: [PhysicalPortsTable] FORMAT PhysicalPortsTable_Index = PhysicalPortsTable_Port, PhysicalPortsTable_Mode, PhysicalPortsTable_NativeVlan, PhysicalPortsTable_SpeedDuplex, PhysicalPortsTable_PortDescription, PhysicalPortsTable_GroupMember, PhysicalPortsTable_GroupStatus; [\PhysicalPortsTable] For example: PhysicalPortsTable 0 = GE_4_1, 1, 1, 4, "User Port #0", GROUP_1, Active; PhysicalPortsTable 1 = GE_4_2, 1, 1, 4, "User Port #1", GROUP_1, Redundant; **Note:** For a description of this parameter, see Configuring Physical Ethernet Ports on page 87.
Ethernet Group Settings Table	
Web: Ethernet Group Settings CLI: configure voip/ether-group [EtherGroupTable]	Defines the transmit (Tx) and receive (Rx) settings for the Ethernet port groups. The format of this parameter is as follows: [EtherGroupTable] FORMAT EtherGroupTable_Index = EtherGroupTable_Group, EtherGroupTable_Mode, EtherGroupTable_Member1, EtherGroupTable_Member2; [\EtherGroupTable] For a description of this parameter, see Configuring Tx/Rx for Ethernet Port-Pair Groups on page 88. **Note:** For this parameter to take effect, a device reset is required.

41.2.2 Multiple VoIP Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

Table 103: IP Network Interfaces and VLAN Parameters

Parameter	Description
Interface Table	
Web: Interface Table CLI: configure voip > interface network-if display **[InterfaceTable]**	This table parameter configures the Interface table. The format of the ini file table parameter is as follows: [InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName, InterfaceTable_PrimaryDNSServerIPAddress, InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingInterface; [\InterfaceTable] For a detailed description of this table, see "Configuring IP Network Interfaces" on page 91. **Note:** For this parameter to take effect, a device reset is required.
[EnableNTPasOAM]	Defines the application type for Network Time Protocol (NTP) services. ▪ **[1]** = OAMP (default)

Parameter	Description
	- **[0]** = Control **Note:** For this parameter to take effect, a device reset is required.

41.2.3 Routing Parameters

The IP network routing parameters are described in the table below.

Table 104: IP Network Routing Parameters

Parameter	Description
Web: Send ICMP Unreachable Messages [DisableICMPUnreachable]	Enables sending of ICMP Unreachable messages. - [0] Enable = (Default) Device sends these messages. - [1] Disable = Device does not send these messages.
Web: ICMP Redirects Messages [DisableICMPRedirects]	Enables sending and receiving of ICMP Redirect messages. - [0] Enable = (Default) Device sends and accepts these messages. - [1] Disable = Device rejects these messages and also does not send them.
Static Route Table	
Web**:** Static Route Table CLI: configure voip > static **[StaticRouteTable]**	Defines up to 30 static IP routes for the device. The format of the ini file table parameter is as follows: [StaticRouteTable] FORMAT StaticRouteTable_Index = StaticRouteTable_DeviceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [\StaticRouteTable] For a description of this parameter, see "Configuring Static IP Routes" on page 99.

41.2.4 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

Table 105: QoS Parameters

Parameter	Description
Layer-2 Class Of Service (CoS) Parameters (VLAN Tag Priority Field)	
Web: DiffServ Table EMS: QoS Settings – DSCP to QoS Mapping CLI: configure voip > vlan-mapping [DiffServToVlanPriority]	This table parameter configures DiffServ-to-VLAN Priority mapping. For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet. The format of this ini file is as follows: [DiffServToVlanPriority] FORMAT DiffServToVlanPriority_Index = DiffServToVlanPriority_DiffServ, DiffServToVlanPriority_VlanPriority; [\DiffServToVlanPriority] For example: DiffServToVlanPriority 0 = 46, 6; DiffServToVlanPriority 1 = 40, 6;

Parameter	Description

	DiffServToVlanPriority 2 = 26, 4;
	DiffServToVlanPriority 3 = 10, 2;
	For a description of this table, see Configuring Quality of Service on page 102.
	Note: For this parameter to take effect, a device reset is required.

Layer-3 Class of Service (TOS/DiffServ) Parameters	
Web: Media Premium QoS CLI: media-qos **[PremiumServiceClassMediaDiffServ]**	Global parameter that defines the DiffServ value for Premium Media CoS content. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IPDiffServ). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Control Premium QoS CLI: control-qos **[PremiumServiceClassControlDiffServ]**	Global parameter that defines the DiffServ value for Premium Control CoS content (Call Control applications). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SigIPDiffServ). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Gold QoS CLI: gold-qos **[GoldServiceClassDiffServ]**	Defines the DiffServ value for the Gold CoS content (Streaming applications). The valid range is 0 to 63. The default is 26.
Web: Bronze QoS CLI: bronze-qos **[BronzeServiceClassDiffServ]**	Defines the DiffServ value for the Bronze CoS content (OAMP applications). The valid range is 0 to 63. The default is 10.

41.2.5 NAT Parameters

The Network Address Translation (NAT) parameters are described in the table below.

Table 106: NAT Parameters

Parameter	Description
Web: NAT Traversal CLI: disable-NAT-traversal **[DisableNAT]**	Enables the NAT mechanism. For more information, see "First Incoming Packet Mechanism" on page 113. ▪ **[0]** Enable ▪ **[1]** Disable (default)
Web: NAT IP Address CLI: nat-ip-addr **[StaticNatIP]**	Defines the global (public) IP address of the device to enable static NAT between the device and the Internet. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: SIP NAT Detection CLI: configure voip/sip-definition advanced-settings/sip-nat-detect **[SIPNatDetection]**	Enables the device to detect whether the incoming INVITE message is sent from an endpoint located behind NAT. ▪ **[0]** Disable = Disables the device's NAT Detection mechanism. Incoming SIP messages are processed as received from endpoints that are not located behind NAT and sent according to the SIP standard. ▪ **[1]** Enable (default) = Enables the device's NAT Detection mechanism.

41.2.6 NFS Parameters

The Network File Systems (NFS) configuration parameters are described in the table below.

Table 107: NFS Parameters

Parameter	Description
CLI: base-port **[NFSBasePort]**	Defines the start of the range of numbers used for local UDP ports used by the NFS client. The maximum number of local ports is maximum channels plus maximum NFS servers. The valid range is 0 to 65535. The default is 47000.
NFS Table	
Web: NFS Table CLI: configure system/nfs/servers **[NFSServers]**	This table parameter defines up to 16 NFS file systems so that the device can access a remote server's shared files and directories for loading cmp, ini, and auxiliary files (using the Automatic Update mechanism). The format of this table ini file parameter is as follows: [NFSServers] FORMAT NFSServers_Index = NFSServers_HostOrIP, NFSServers_RootPath, NFSServers_NfsVersion, NFSServers_AuthType, NFSServers_UID, NFSServers_GID, NFSServers_VlanType; [\NFSServers] For example: NFSServers 1 = 101.1.13, /audio1, 3, 1, 0, 1, 1; For a detailed description of this table, see "Configuring NFS Settings" on page 108.

41.2.7 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table 108: DNS Parameters

Parameter	Description
Internal DNS Table	

Parameter	Description
Web: Internal DNS Table CLI: configure voip > voip-network dns Dns2Ip **[DNS2IP]**	This table parameter defines the internal DNS table for resolving host names into IP addresses. Up to four different IP addresses (in dotted-decimal notation) can be assigned to a host name. The format of this parameter is as follows: [Dns2Ip] FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress, Dns2Ip_FourthIpAddress; [\Dns2Ip] For example: Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, 4.4.4.4; For a detailed description of this table, see "Configuring the Internal DNS Table" on page 105.
Internal SRV Table	
Web: Internal SRV Table CLI: configure voip > voip-network dns Srv2Ip **[SRV2IP]**	This table parameter defines the internal SRV table for resolving host names into DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of this parameter is as follows: [SRV2IP] FORMAT SRV2IP_Index = SRV2IP_InternalDomain, SRV2IP_TransportType, SRV2IP_Dns1, SRV2IP_Priority1, SRV2IP_Weight1, SRV2IP_Port1, SRV2IP_Dns2, SRV2IP_Priority2, SRV2IP_Weight2, SRV2IP_Port2, SRV2IP_Dns3, SRV2IP_Priority3, SRV2IP_Weight3, SRV2IP_Port3; [\SRV2IP] For example: SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,$$,0,0,0; For a detailed description of this table, see "Configuring the Internal SRV Table" on page 106.

41.2.8 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

Table 109: DHCP Parameters

Parameter	Description
Web: Enable DHCP **[DHCPEnable]**	Enables Dynamic Host Control Protocol (DHCP) functionality. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a detailed description of DHCP, see "DHCP-based Configuration Server" on page 323. ▪ This parameter is a "hidden" parameter. Once defined and saved to flash memory, its value doesn't revert to default even if the parameter doesn't appear in the *ini* file.
[DHCPSpeedFactor]	Defines the DHCP renewal speed. ▪ **[0]** = Disable ▪ **[1]** = (Default) Normal ▪ **[2]** to **[10]** = Fast When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is

Parameter	Description
	divided by this factor. Some DHCP-enabled routers perform better when set to 4. **Note:** For this parameter to take effect, a device reset is required.

41.2.9 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

Table 110: NTP and Daylight Saving Time Parameters

Parameter	Description
NTP Parameters **Note:** For more information on Network Time Protocol (NTP), see "Simple Network Time Protocol Support" on page 82.	
Web: NTP Server Address CLI: primary-server **[NTPServerIP]**	Defines the IP address (in dotted-decimal notation or as an FQDN) of the NTP server. The advantage of using an FQDN is that multiple IP addresses can be resolved from the DNS server, providing NTP server redundancy. The default IP address is 0.0.0.0 (i.e., internal NTP client is disabled).
Web: NTP Secondary Server Address **[NTPSecondaryServerIP]**	Defines a second NTP server's address as an FQDN or an IP address (in dotted-decimal notation). This NTP is used for redundancy; if the primary NTP server fails, then this NTP server is used. The default IP address is 0.0.0.0.
Web: NTP UTC Offset CLI: utc-offset **[NTPServerUTCOffset]**	Defines the Universal Time Coordinate (UTC) offset (in seconds) from the NTP server. The default offset is 0. The offset range is -43200 to 43200.
Web: NTP Update Interval CLI: update-interval **[NTPUpdateInterval]**	Defines the time interval (in seconds) that the NTP client requests for a time update. The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647. **Note:** It is not recommend to set this parameter to beyond one month (i.e., 2592000 seconds).
Web: NTP Authentication Key Identifier CLI: configure system > ntp > auth-key-id **[NtpAuthKeyId]**	Defines the NTP authentication key identifier for authenticating NTP messages. The identifier must match the value configured on the NTP server. The NTP server may have several keys configured for different clients; this number identifies which key is used. The valid value is 1 to 65535. The default is 0 (i.e., no authentication is done).
Web: NTP Authentication Secret Key CLI: configure system > ntp > auth-key-md5 **[ntpAuthMd5Key]**	Defines the secret authentication key shared between the device (client) and the NTP server, for authenticating NTP messages. The valid value is a string of up to 32 characters. By default, no key is defined.
Daylight Saving Time Parameters	
Web: Day Light Saving Time CLI: summer-time **[DayLightSavingTimeEnable]**	Enables daylight saving time. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Start Time / Day of Month Start CLI: start **[DayLightSavingTimeStart]**	Defines the date and time when daylight saving begins. This value can be configured using any of the following formats: ▪ Day of year - *mm:dd:hh:mm*, where: ✓ *mm* denotes month

Parameter	Description
	✓ *dd* denotes date of the month ✓ *hh* denotes hour ✓ *mm* denotes minutes ▪ For example, "05:01:08:00" denotes daylight saving starting from May 1 at 8 A.M. ▪ Day of month - *mm:day/wk:hh:mm*, where: ✓ *mm* denotes month (e.g., 04) ✓ *day* denotes day of week (e.g., FRI) ✓ *wk* denotes week of the month (e.g., 03) ✓ *hh* denotes hour (e.g., 23) ✓ *mm* denotes minutes (e.g., 10) ▪ For example, "04:FRI/03:23:00" denotes Friday, the third week of April, at 11 P.M. The week field can be 1-5, where 5 denotes the last occurrence of the specified day in the specified month. For example, "04:FRI/05:23:00" denotes the last Friday of April, at 11 P.M.
Web: End Time / Day of Month End CLI: end **[DayLightSavingTimeEnd]**	Defines the date and time when daylight saving ends. For a description of the format of this value, see the DayLightSavingTimeStart parameter.
Web: Offset CLI: offset **[DayLightSavingTimeOffset]**	Defines the daylight saving time offset (in minutes). The valid range is 0 to 120. The default is 60.

41.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

41.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

Table 111: General Debugging and Diagnostic Parameters

Parameter	Description
[EnableDiagnostics]	Determines the method for verifying correct functioning of the different hardware components on the device. On completion of the check and if the test fails, the device sends information on the test results of each hardware component to the Syslog server. ▪ **[0]** = (Default) Rapid and Enhanced self-test mode. ▪ **[1]** = Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY and Flash). ▪ **[2]** = A quicker version of the Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY, but partial test of Flash). **Note:** For this parameter to take effect, a device reset is required.
Web: Delay After Reset [sec] CLI: delay-after-reset **[GWAppDelayTime]**	Defines the time interval (in seconds) that the device's operation is delayed after a reset. The valid range is 0 to 45. The default is 7 seconds. **Note:** This feature helps overcome connection problems caused by some LAN routers or IP configuration parameters' modifications by a DHCP server.
[EnableAutoRAITransmitBER]	Enables the device to send a remote alarm indication (RAI) when the bit error rate (BER) is greater than 0.001. ▪ **[0]** Disable (default) ▪ **[1]** Enable

41.3.2 SIP Test Call Parameters

The SIP Signaling Test Call parameters are described in the table below.

Table 112: SIP Test Call Parameters

Parameter	Description
Web: Test Call DTMF String CLI:testcall-dtmf-string **[TestCallDtmfString]**	Defines the DTMF tone that is played for answered test calls (incoming and outgoing). The DTMF string can be up to 15 strings. The default is "3212333". If no string is defined (empty), DTMF is not played.
Web: Test Call ID CLI: testcall-id **[TestCallID]**	Defines the test call prefix number (*ID*) of the simulated phone on the device. Incoming calls received with this called prefix number are identified as test calls. This can be any string of up to 15 characters. By default, no number is defined. **Note:** This parameter is only for testing incoming calls destined to this prefix number.

Parameter	Description
Web: SBC Test ID CLI: sbc-test-id [SBCtestID]	Defines the SBC test call prefix (ID) for identifying SBC test calls that traverse the device to register with an external routing entity such as an IP PBX or proxy server. This parameter functions together with the TestCallID parameter, which defines the prefix of the simulated endpoint. Upon receiving an incoming call with this prefix, the device removes the prefix, enabling it to forward the test call to the external entity. Upon receiving the call from the external entity, the device identifies the call as a test call according to its prefix, defined by the TestCallID, and then sends the call to the simulated endpoint. For example, assume SBCTestID is set to 4 and TestCallID to 2. If a call is received with called destination 4200, the device removes the prefix 4 and routes the call to the IP PBX. When it receives the call from the IP PBX, it identifies the call as a test call (i.e., prefix 2) and therefore, sends it to the simulated endpoint. The valid value can be any string of up to 15 characters. By default, no number is defined.
Test Call Table	
Web: Test Call Table CLI: configure system > test-call > test-call-table **[Test_Call]**	Defines the local and remote endpoints to be tested. [Test_Call] FORMAT Test_Call_Index = Test_Call_EndpointURI, Test_Call_CalledURI, Test_Call_RouteBy, Test_Call_IPGroupID, Test_Call_DestAddress, Test_Call_DestTransportType, Test_Call_SRD, Test_Call_ApplicationType, Test_Call_AutoRegister, Test_Call_UserName, Test_Call_Password, Test_Call_CallParty, Test_Call_MaxChannels, Test_Call_CallDuration, Test_Call_CallsPerSecond, Test_Call_TestMode, Test_Call_TestDuration, Test_Call_Play, Test_Call_ScheduleInterval, Test_Call_QOEProfile, Test_Call_BWProfile; [\Test_Call] For a description of this table, see "Configuring Test Call Endpoints" on page 368.

41.3.3 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table 113: Syslog, CDR and Debug Parameters

Parameter	Description
Web: Enable Syslog CLI: syslog **[EnableSyslog]**	Determines whether the device sends logs and error messages (e.g., CDRs) generated by the device to a Syslog server. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If you enable Syslog, you must enter an IP address of the Syslog server (using the SyslogServerIP parameter). ▪ Syslog messages may increase the network traffic. ▪ To configure Syslog SIP message logging levels, use the GwDebugLevel parameter.
Web: Syslog Server IP Address CLI: syslog-ip **[SyslogServerIP]**	Defines the IP address (in dotted-decimal notation) of the computer on which the Syslog server is running. The Syslog server is an application designed to collect the logs and error messages generated by the device. The default IP address is 0.0.0.0.

Parameter	Description
Web: Syslog Server Port CLI: syslog-port **[SyslogServerPort]**	Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514.
CLI: mx-syslog-lgth **[MaxBundleSyslogLength]**	Defines the maximum size (in bytes) threshold of logged Syslog messages bundled into a single UDP packet, after which they are sent to a Syslog server. The valid value range is 0 to 1220 (where 0 indicates that no bundling occurs). The default is 1220. **Note:** This parameter is applicable only if the GWDebugLevel parameter is set to 7.
Web: CDR Server IP Address CLI: cdr-srvr-ip-adrr **[CDRSyslogServerIP]**	Defines the destination IP address to where CDR logs are sent. The default value is a null string, which causes CDR messages to be sent with all Syslog messages to the Syslog server. **Notes:** ▪ The CDR messages are sent to UDP port 514 (default Syslog port). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web: CDR Report Level CLI: cdr-report-level **[CDRReportLevel]**	Enables signaling-related CDRs to be sent to a Syslog server and determines the call stage at which they are sent. ▪ **[0]** None = (Default) CDRs are not used. ▪ **[1]** End Call = CDR is sent to the Syslog server at the end of each call. ▪ **[2]** Start & End Call = CDR report is sent to Syslog at the start and end of each call. ▪ **[3]** Connect & End Call = CDR report is sent to Syslog at connection and at the end of each call. ▪ **[4]** Start & End & Connect Call = CDR report is sent to Syslog at the start, at connection, and at the end of each call. **Notes:** ▪ For the SBC application, this parameter enables only signaling-related CDRs. To enable media-related CDRs for SBC calls, use the MediaCDRReportLevel parameter. ▪ The CDR Syslog message complies with RFC 3161 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web: Media CDR Report Level [MediaCDRReportLevel]	Enables media-related CDRs of SBC calls to be sent to a Syslog server and determines the call stage at which they are sent. ▪ [0] None = (Default) No media-related CDR is sent. ▪ [1] End Media = Sends a CDR only at the end of the call. ▪ [2] Start & End Media = Sends a CDR once the media starts. In some calls it may only be after the call is established, but in other calls the media may start at ringback tone. A CDR is also sent upon termination (end) of the media in the call. ▪ [3] Update & End Media = Sends a CDR when an update occurs in the media of the call. For example, a call starts and a ringback tone occurs, a re-INVITE is sent for a fax call and as a result, a CDR with the MediaReportType field set to "Update" is sent, as the media was changed from voice to T.38. A CDR is also sent upon termination (end) of the media in the call. ▪ [4] Start & End & Update Media = Sends a CDR at the start of the media, upon an update in the media (if occurs), and at the end of the media. **Note:** To enable CDR generation as well as enable signaling-related CDRs, use the CDRReportLevel parameter.

Parameter	Description
Web: Debug Level CLI: debug-level **[GwDebugLevel]**	Defines the Syslog debug logging level. ▪ **[0]** 0 = (Default) Debug is disabled. ▪ **[1]** 1 = Flow debugging is enabled. ▪ **[5]** 5 = Flow, device interface, stack interface, session manager, and device interface expanded debugging are enabled. ▪ **[7]** 7 = This option is recommended when the device is running under "heavy" traffic. In this mode: ✓ The Syslog debug level automatically changes between level 5, level 1, and level 0, depending on the device's CPU consumption so that VoIP traffic isn't affected. ✓ Syslog messages are bundled into a single UDP packet, after which they are sent to a Syslog server (bundling size is determined by the MaxBundleSyslogLength parameter). Bundling reduces the number of UDP Syslog packets, thereby improving CPU utilization. ▪ Note that when this option is used, in order to read Syslog messages with Wireshark, a special plug-in (i.e., acsyslog.dll) must be used. Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and are dispalyed using the 'acsyslog' filter instead of the regular 'syslog' filter. **Notes:** ▪ This parameter is typically set to 5 if debug traces are required. However, in cases of heavy traffic, option 7 is recommended. ▪ Options 2, 3, 4, and 6 are not recommended.
Web: Syslog Facility Number **[SyslogFacility]**	Defines the Facility level (0 through 7) of the device's Syslog messages, according to RFC 3164. This allows you to identify Syslog messages generated by the device. This is useful, for example, if you collect the device's and other equipments' Syslog messages, at one single server. The device's Syslog messages can easily be identified and distinguished from other Syslog messages by its Facility level. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level. ▪ **[16]** = (Default) local use 0 (local0) ▪ **[17]** = local use 1 (local1) ▪ **[18]** = local use 2 (local2) ▪ **[19]** = local use 3 (local3) ▪ **[20]** = local use 4 (local4) ▪ **[21]** = local use 5 (local5) ▪ **[22]** = local use 6 (local6) ▪ **[23]** = local use 7 (local7)
Web: CDR Session ID CLI: cdr-seq-num **[CDRSyslogSeqNum]**	Enables or disables the inclusion of the sequence number (S=) in CDR Syslog messages. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Web: Activity Types to Report via Activity Log Messages **[ActivityListToLog]**	Defines the Activity Log mechanism of the device, which sends log messages to a Syslog server for reporting certain types of Web operations according to the below user-defined filters. ▪ **[pvc]** Parameters Value Change = Changes made on-the-fly to parameters. Note that the *ini* file parameter, EnableParametersMonitoring can also be used to set this option, using values **[0]** (disable) or **[1]** (enable). ▪ **[afl]** Auxiliary Files Loading = Loading of auxiliary files. ▪ **[dr]** Device Reset = Reset of device via the 'Maintenance Actions page. **Note:** For this option to take effect, a device reset is required. ▪ **[fb]** Flash Memory Burning = Burning of files or parameters to flash (in 'Maintenance Actions page).

Parameter	Description
	▪ **[swu]** Device Software Update = cmp file loading via the Software Upgrade Wizard. ▪ **[ard]** Access to Restricted Domains = Access to restricted domains, which include the following Web pages: ✓ (1) ini parameters (AdminPage) ✓ (2) General Security Settings ✓ (3) Configuration File ✓ (5) Software Upgrade Key Status ✓ (7) Web & Telnet Access List ✓ (8) WEB User Accounts ▪ **[naa]** Non-Authorized Access = Attempt to access the Web interface with a false or empty user name or password. ▪ **[spc]** Sensitive Parameters Value Change = Changes made to sensitive parameters: ✓ (1) IP Address ✓ (2) Subnet Mask ✓ (3) Default Gateway IP Address ✓ (4) ActivityListToLog ▪ **[ll]** Login and Logout = Every login and logout attempt. For example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc' **Note:** For the *ini* file, values must be enclosed in single quotation marks.
Web: Debug Recording Destination IP CLI: configure system > logging > dbg-rec-dest-ip **[DebugRecordingDestIP]**	Defines the IP address of the server for capturing debug recording.
Web: Debug Recording Destination Port CLI: configure system > logging > dbg-rec-dest-port **[DebugRecordingDestPort]**	Defines the UDP port of the server for capturing debug recording. The default is 925.
Debug Recording Status CLI: configure system > logging > dbg-rec-status **[DebugRecordingStatus]**	Activates or de-activates debug recording. ▪ **[0]** Stop (default) ▪ **[1]** Start
Web: Enable Core Dump **[EnableCoreDump]**	Enables the automatic generation of a Core Dump file upon a device crash. ▪ **[0]** Disable (disable) ▪ **[1]** Enable
Web: Core Dump Destination IP **[CoreDumpDestIP]**	Defines the IP address of the remote server where you want the device to send the Core Dump file. By default, no IP address is defined.
Logging Filters Table	
Web: Logging Filters Table CLI: configure system > logging > logging-filters **[LoggingFilters]**	This table parameter defines logging filtering rules for Syslog messages and debug recordings. The format of the ini file table parameter is: [LoggingFilters] FORMAT LoggingFilters_Index = LoggingFilters_FilterType, LoggingFilters_Value, LoggingFilters_Syslog, LoggingFilters_CaptureType; [\LoggingFilters] For a detailed description of this table, see ''Filtering Syslog Messages and

Parameter	Description
	Debug Recordings" on page 360.

41.3.4 Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

Table 114: RAI Parameters

Parameter	Description
[EnableRAI]	Enables RAI alarm generation if the device's busy endpoints exceed a user-defined threshold. ▪ **[0]** = (Default) Disable RAI (Resource Available Indication) service. ▪ **[1]** = RAI service enabled and an SNMP 'acBoardCallResourcesAlarm' Alarm Trap is sent.
[RAIHighThreshold]	Defines the high threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints exceeds this high threshold, the device sends the SNMP acBoardCallResourcesAlarm alarm trap with a 'major' alarm status. The range is 0 to 100. The default is 90. **Note:** The percentage of busy endpoints is calculated by dividing the number of busy endpoints by the total number of "enabled" endpoints.
[RAILowThreshold]	Defines the low threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints falls below this low threshold, the device sends an SNMP acBoardCallResourcesAlarm alarm trap with a 'cleared' alarm status. The range is 0 to 100%. The default is 90%.
[RAILoopTime]	Defines the time interval (in seconds) that the device periodically checks call resource availability. The valid range is 1 to 200. The default is 10.

41.3.5 HA Ping Parameters

The High Availability (HA) Ping parameters are described in the table below.

Table 115: HA Ping Parameters

Parameter	Description
Web: HA Network Reachability **[HAPingEnabled]**	Enables the pinging of an active IP network destination in HA mode to test reachability from one of the device's IP network interfaces. If no reply is received from a ping and the previous ping was successful, a switchover occurs to the redundant device. ▪ **[0]** Disabled (default) ▪ **[1]** Enabled
Web: HA Network Reachability Destination Address **[HAPingDestination]**	Defines the IP address of the destination that the device pings. The default is 0.0.0.0.

Parameter	Description
Web: HA Network Reachability Source Interface Name **[HAPingSourceIfName]**	Defines the device's IP network interface from where the ping is sent. The valid value is the name of the IP interface as configured in the 'Interface Name' field of the Interface table. By default, no IP network is defined.
HA Network Reachability Ping Timeout **[HAPingTimeout]**	Defines the timeout (in seconds) for which the ping request waits for a reply. The valid value is 1 to 60. The default is 1.
HA Network Reachability Ping Retries **[HAPingRetries]**	Defines the number of ping requests that the device sends after no response is received from the destination, before the destination is declared unavailable. For example, if you specify 2, the destination is declared as down after three consecutive ping requests fail to evoke a response from the destination. The valid value is 0 to 100. The default 2.

41.4 Security Parameters

This subsection describes the device's security parameters.

41.4.1 General Security Parameters

The general security parameters are described in the table below.

Table 116: General Security Parameters

Parameter	Description
Firewall Table	
Web: Internal Firewall Parameters CLI: configure voip > access-list **[AccessList]**	This table parameter defines the device's access list (firewall), which defines network traffic filtering rules. The format of this parameter is as follows: [AccessList] FORMAT AccessList_Index = AccessList_Source_IP, AccessList_Source_Port, AccessList_PrefixLen, AccessList_Source_Port, AccessList_Start_Port, AccessList_End_Port, AccessList_Protocol, AccessList_Use_Specific_Interface, AccessList_Interface_ID, AccessList_Packet_Size, AccessList_Byte_Rate, AccessList_Byte_Burst, AccessList_Allow_Type; [\AccessList] For example: AccessList 10 = mgmt.customer.com, , , 32, 0, 80, tcp, 1, OAMP, 0, 0, 0, allow; AccessList 22 = 10.4.0.0, , , 16, 4000, 9000, any, 0, , 0, 0, 0, block; In the example above, Rule #10 allows traffic from the host 'mgmt.customer.com' destined to TCP ports 0 to 80 on interface OAMP (OAMP). Rule #22 blocks traffic from the subnet 10.4.xxx.yyy destined to ports 4000 to 9000. For a detailed description of this table, see "Configuring Firewall Settings" on page 116.
Media Latching	
Web/EMS: Inbound Media Latch Mode CLI: inbound-media-latch-mode [InboundMediaLatchMode]	Enables the Media Latching feature. ▪ [0] Strict = Device latches onto the first original stream (IP address:port). It does not latch onto any other stream during the session. ▪ [1] Dynamic = (Default) Device latches onto the first stream. If it receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) from a different source(s) and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other stream. If other packets of a different media type are received from the new stream, based on IP address and SSRC for RTCP/RTP and based on IP address only for T.38, the packet is accepted immediately. Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream. ▪ [2] Dynamic-Strict = Device latches onto the first stream. If it receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) all from the same source which is different to the first stream and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other

Parameter	Description
	stream. If other packets of different media type are received from the new stream based on IP address and SSRC for RTCP and based on IP address only for T.38, the packet is accepted immediately. Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream. ▪ [3] Strict-On-First = Typically used for NAT, where the correct IP address:port is initially unknown. The device latches onto the stream received in the first packet. The device does not change this stream unless a packet is later received from the original source.
New RTP Stream Packets [NewRtpStreamPackets]	Defines the minimum number of continuous RTP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
New RTCP Stream Packets [NewRtcpStreamPackets]	Defines the minimum number of continuous RTCP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
New SRTP Stream Packets [NewSRTPStreamPackets]	Defines the minimum number of continuous SRTP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
New SRTCP Stream Packets [NewSRTCPStreamPackets]	Defines the minimum number of continuous SRTCP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
Timeout To Relatch RTP (msec) [TimeoutToRelatchRTPMsec]	Defines a period (msec) during which if no packets are received from the current RTP session, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch SRTP [TimeoutToRelatchSRTPMsec]	Defines a period (msec) during which if no packets are received from the current SRTP session, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch Silence [TimeoutToRelatchSilenceMsec]	Defines a period (msec) during which if no packets are received from the current RTP/SRTP session and the channel is in silence mode, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch RTCP [TimeoutToRelatchRTCPMsec]	Defines a period (msec) during which if no packets are received from the current RTCP session, the channel can re-latch onto another RTCP stream. The valid range is any value from 0. The default is 10,000.
Fax Relay Rx/Tx Timeout [FaxRelayTimeoutSec]	Defines a period (sec) during which if no T.38 packets are received or sent from the current T.38 fax relay session, the channel can re-latch onto another stream. The valid range is 0 to 255. The default is 10.

41.4.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

Table 117: HTTPS Parameters

Parameter	Description
Web: Secured Web Connection (HTTPS) CLI: secured-connection **[HTTPSOnly]**	Determines the protocol used to access the Web interface. ▪ **[0]** HTTP and HTTPS (default). ▪ **[1]** HTTPs Only = Unencrypted HTTP packets are blocked. **Note:** For this parameter to take effect, a device reset is required.
CLI: https-port **[HTTPSPort]**	Defines the local Secured HTTPS port of the device. This parameter allows secure remote device Web management from the LAN. To enable secure Web management from the LAN, configure the desired port. The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443. **Note:** For this parameter to take effect, a device reset is required.
Web/: HTTPS Cipher String CLI: https-cipher-string **[HTTPSCipherString]**	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). For the valid range values, refer to URL http://www.openssl.org/docs/apps/ciphers.html. The default is 'RC4:EXP' (Export encryption algorithms). For example, use 'ALL' for all ciphers suites (e.g., for ARIA encryption for TLS). The only ciphers available are RC4 and DES, and the cipher bit strength is limited to 56 bits. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If the installed Software License Key includes the Strong Encryption feature, the default of this parameter is changed to 'RC4:EXP', enabling RC-128bit encryption. ▪ The value 'ALL' can be configured only if the installed Software License Key includes the Strong Encryption feature.
Web: HTTP Authentication Mode CLI: http-auth-mode **[WebAuthMode]**	Determines the authentication mode used for the Web interface. ▪ **[0]** Basic Mode =Basic authentication (clear text) is used. ▪ **[1]** Web Based Authentication = (Default) Digest authentication (MD5) is used. **Note:** If you enable RADIUS login (i.e., the WebRADIUSLogin parameter is set to 1), you must set the WebAuthMode parameter to Basic Mode [0].
Web: Requires Client Certificates for HTTPS connection CLI: req-client-cert **[HTTPSRequireClientCertificate]**	Determines whether client certificates are required for HTTPS connection. ▪ **[0]** Disable = (Default) Client certificates are not required. ▪ **[1]** Enable = Client certificates are required. The client certificate must be preloaded to the device and its matching private key must be installed on the managing PC. Time and date must be correctly set on the device for the client certificate to be verified. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For a description on implementing client certificates, see "Client Certificates" on page 78.

41.4.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

Table 118: SRTP Parameters

Parameter	Description
Web: Media Security	Enables Secure Real-Time Transport Protocol (SRTP).

Parameter	Description
CLI: media-security-enable **[EnableMediaSecurity]**	▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Media Security Behavior CLI: media-sec-bhvior **[MediaSecurityBehaviour]**	Global parameter that defines the handling of SRTP (when the EnableMediaSecurity parameter is set to 1). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MediaSecurityBehaviour). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Master Key Identifier (MKI) Size CLI: SRTP-tx-packet-MKI-size **[SRTPTxPacketMKISize]**	Global parameter that defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MKISize). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Symmetric MKI Negotiation CLI: symmetric-mki **[EnableSymmetricMKI]**	Global parameter that enables symmetric MKI negotiation. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableSymmetricMKI). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: SRTP offered Suites CLI: offer-srtp-cipher **[SRTPofferedSuites]**	Defines the offered crypto suites (cipher encryption algorithms) for SRTP. ▪ **[0]** = (Default) All available crypto suites. ▪ [1] CIPHER SUITES AES CM 128 HMAC SHA1 80 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 80-bit tag. ▪ **[2]** CIPHER SUITES AES CM 128 HMAC SHA1 32 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag. **Note:** This parameter also affects the selection of the crypto in the device's answer. For example, if the device receives an offer with two crypto lines containing HMAC_SHA1_80 and HMAC_SHA_32, it uses the HMAC_SHA_32 key in its SIP 200 OK response if the parameter is set to 2.
Web: Disable Authentication On Transmitted RTP Packets CLI: RTP-authentication-disable-tx **[RTPAuthenticationDisableTx]**	Enables authentication on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTP Packets CLI: RTP-encryption-disable-tx **[RTPEncryptionDisableTx]**	Enables encryption on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTCP Packets CLI: RTCP-encryption-disable-tx **[RTCPEncryptionDisableTx]**	Enables encryption on transmitted RTCP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable

Parameter	Description
CLI: srtp-state-behavior-mode **[ResetSRTPStateUponRekey]**	Global parameter that enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_ResetSRTPStateUponRekey). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.

41.4.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

Table 119: TLS Parameters

Parameter	Description
Web: TLS Version CLI: version **[TLSVersion]**	Determines the supported versions of SSL/TLS (Secure Socket Layer/Transport Layer Security. ▪ **[0]** SSL 2.0-3.0 and TLS 1.0 = (Default) SSL 2.0, SSL 3.0, and TLS 1.0 are supported. ▪ **[1]** TLS 1.0 Only = only TLS 1.0 is used. When set to 0, SSL/TLS handshakes always start with SSL 2.0 and switch to TLS 1.0 if both peers support it. When set to 1, TLS 1.0 is the only version supported; clients attempting to contact the device using SSL 2.0 are rejected. **Note:** For this parameter to take effect, a device reset is required.
Web: TLS Client Re-Handshake Interval CLI: tls-re-hndshk-int **[TLSReHandshakeInterval]**	Defines the time interval (in minutes) between TLS Re-Handshakes initiated by the device. The interval range is 0 to 1,500 minutes. The default is 0 (i.e., no TLS Re-Handshake).
Web: TLS Mutual Authentication **[SIPSRequireClientCertificate]**	Determines the device's behavior when acting as a server for TLS connections. ▪ **[0]** Disable = (Default) The device does not request the client certificate. ▪ **[1]** Enable = The device requires receipt and verification of the client certificate to establish the TLS connection. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName.
Web: Peer Host Name Verification Mode **[PeerHostNameVerificationMode]**	Determines whether the device verifies the Subject Name of a remote certificate when establishing TLS connections. ▪ **[0]** Disable (default). ▪ **[1]** Server Only = Verify Subject Name only when acting as a client for the TLS connection. ▪ **[2]** Server & Client = Verify Subject Name when acting as a server or client for the TLS connection. When a remote certificate is received and this parameter is not disabled, the value of SubjectAltName is compared with the list of available Proxies. If a match is found for any of the configured Proxies, the TLS

Parameter	Description
	connection is established. The comparison is performed if the SubjectAltName is either a DNS name (DNSName) or an IP address. If no match is found and the SubjectAltName is marked as 'critical', the TLS connection is not established. If DNSName is used, the certificate can also use wildcards ('*') to replace parts of the domain name. If the SubjectAltName is not marked as 'critical' and there is no match, the CN value of the SubjectName field is compared with the parameter TLSRemoteSubjectName. If a match is found, the connection is established. Otherwise, the connection is terminated. **Note:** If you set this parameter to [2] (Server & Client), for this functionality to operate, you also need to set the SIPSRequireClientCertificate parameter to [1] (Enable).
Web: TLS Client Verify Server Certificate CLI: tls-vrfy-srvr-cert **[VerifyServerCertificate]**	Determines whether the device, when acting as a client for TLS connections, verifies the Server certificate. The certificate is verified with the Root CA information. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** If Subject Name verification is necessary, the parameter PeerHostNameVerificationMode must be used as well.
Web: Strict Certificate Extension Validation CLI: require-strict-cert **[RequireStrictCert]**	Enables the validation of the extensions (keyUsage and extentedKeyUsage) of peer certificates. This validation ensures that the signing CA is authorized to sign certificates and that the end-entity certificate is authorized to negotiate a secure TLS connection. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: TLS Remote Subject Name CLI: tls-rmt-subs-name **[TLSRemoteSubjectName]**	Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. If the CN uses a domain name, the certificate can also use wildcards ('*') to replace parts of the domain name. The valid range is a string of up to 49 characters. **Note:** This parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.
Web: Client Cipher String CLI: client-cipher-string **[TLSClientCipherString]**	Defines the cipher-suite string for TLS clients. The valid value is up to 255 strings. The default is "ALL:!ADH". For example: TLSClientCipherString = 'EXP' This parameter complements the HTTPSCipherString parameter (which affects TLS servers). For possible values and additional details, refer to: http://www.openssl.org/docs/apps/ciphers.html
CLI: pkey-size **[TLSPkeySize]**	Defines the key size (in bits) for RSA public-key encryption for newly self-signed generated keys for SSH. ▪ **[512]** ▪ **[768]** ▪ **[1024]** (default) ▪ **[2048]**

Parameter	Description
Web: TLS Expiry Check Start CLI: expiry-check-start **[TLSExpiryCheckStart]**	Defines the number of days before the installed TLS server certificate is to expire at which the device must send a trap (acCertificateExpiryNotifiaction) to notify of this. The valid value is 0 to 3650. The default is 60.
Web: TLS Expiry Check Period CLI: expiry-check-period **[TLSExpiryCheckPeriod]**	Defines the periodical interval (in days) for checking the TLS server certificate expiry date. The valid value is 1 to 3650. The default is 7.

41.4.5 SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

Table 120: SSH Parameters

Parameter	Description
Web: Enable SSH Server CLI: ssh **[SSHServerEnable]**	Enables the device's embedded SSH server. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Server Port cli: ssh-port **[SSHServerPort]**	Defines the port number for the embedded SSH server. Range is any valid port number. The default port is 22.
Web: SSH Admin Key CLI: ssh-admin-key **[SSHAdminKey]**	Defines the RSA public key for strong authentication for logging in to the SSH interface (if enabled). The value should be a base64-encoded string. The value can be a maximum length of 511 characters.
Web: Require Public Key CLI: ssh-require-public-key **[SSHRequirePublicKey]**	Enables RSA public keys for SSH. ▪ **[0]** = (Default) RSA public keys are optional if a value is configured for the parameter SSHAdminKey. ▪ **[1]** = RSA public keys are mandatory. **Note:** To define the key size, use the TLSPkeySize parameter.
Web: Max Payload Size CLI: ssh-max-payload-size **[SSHMaxPayloadSize]**	Defines the maximum uncompressed payload size (in bytes) for SSH packets. The valid value is 550 to 32768. The default is 32768.
Web: Max Binary Packet Size CLI: ssh-max-binary-packet-size **[SSHMaxBinaryPacketSize]**	Defines the maximum packet size (in bytes) for SSH packets. The valid value is 582 to 35000. The default is 35000.
Web: Maximum SSH Sessions CLI: ssh-max-sessions **[SSHMaxSessions]**	Defines the maximum number of simultaneous SSH sessions. The valid range is 1 to 5. The default is 2 sessions.
Web: Enable Last Login Message CLI: ssh-last-login-message **[SSHEnableLastLoginMessage]**	Enables message display in SSH sessions of the time and date of the last SSH login. The SSH login message displays the number of unsuccessful login attempts since the last successful login. ▪ **[0]** Disable ▪ **[1]** Enable (default) **Note:** The last SSH login information is cleared when the device is reset.
Web: Max Login Attempts CLI: ssh-max-login-attempts	Defines the maximum SSH login attempts allowed for entering an incorrect password by an administrator before the SSH session is rejected.

Parameter	Description
[SSHMaxLoginAttempts]	The valid range is 1 to 3. the default is 3.

41.4.6 IDS Parameters

The Intrusion Detection System (IDS) parameters are described in the table below.

Table 121: IDS Parameters

Parameter	Description
Web: Intrusion Detection System (IDS) CLI: enable-ids **[EnableIDS]**	Enables the IDS feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
CLI: ids-clear-period **[IDSAlarmClearPeriod]**	Defines the interval (in seconds) after which an IDS alarm is cleared from the Active Alarms table if no thresholds are crossed during this time. However, this "quiet" period must be at least twice the Threshold Window value. For example, if IDSAlarmClearPeriod is set to 20 sec and the Threshold Window is set to 15 sec, the IDSAlarmClearPeriod parameter is ignored and the alarm is cleared only after 30 seconds (2 x 15 sec). The valid value is 0 to 86400. The default is 300.
IDS Policy Table	
Web: IDS Policy Table **[IDSPolicy]**	Defines IDS Policies. The format of the ini file parameter is: [IDSPolicy] FORMAT IDSPolicy_Index = IDSPolicy_Name, IDSPolicy_Description; [\IDSPolicy] For a detailed description of this table, see "Configuring IDS Policies" on page 121.
IDS Rule Table	
Web: IDS Rule Table **[IDSRule]**	Defines rules for IDS Policies. The format of the ini file parameter is: [IDSRule] FORMAT IDSRule_Index = IDSRule_Policy, IDSRule_RuleID, IDSRule_Reason, IDSRule_ThresholdScope, IDSRule_ThresholdWindow, IDSRule_MinorAlarmThreshold, IDSRule_MajorAlarmThreshold, IDSRule_CriticalAlarmThreshold, IDSRule_DenyThreshold, IDSRule_DenyPeriod; [\IDSRule] For a detailed description of this table, see "Configuring IDS Policies" on page 121.
IDS Match Table	
Web: IDS Match Table **[IDSMatch]**	Defines target rules per IDS Policy. The format of the ini file parameter is: [IDSMatch] FORMAT IDSMatch_Index = IDSMatch_SIPInterface, IDSMatch_ProxySet, IDSMatch_Subnet, IDSMatch_Policy; [\IDSMatch] For a detailed description of this table, see "Assigning IDS Policies" on page 125.

41.4.7 OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

Table 122: OCSP Parameters

Parameter	Description
Web: Enable OCSP Server CLI: enable **[OCSPEnable]**	Enables or disables certificate checking using OCSP. ▪ **[0]** Disable (default) ▪ **[1]** Enable For a description of OCSP, see "Configuring Certificate Revocation Checking (OCSP)" on page 80.
Web: Primary Server IP CLI: server-ip **[OCSPServerIP]**	Defines the IP address of the OCSP server. The default IP address is 0.0.0.0.
Web: Secondary Server IP CLI: secondary-server-ip **[OCSPSecondaryServerIP]**	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
Web: Server Port CLI: server-port **[OCSPServerPort]**	Defines the OCSP server's TCP port number. The default port number is 2560.
Web: Default Response When Server Unreachable CLI: default-response **[OCSPDefaultResponse]**	Determines whether the device allows or rejects peer certificates when the OCSP server cannot be contacted. ▪ **[0]** Reject (default) ▪ **[1]** Allow

41.4.8 RADIUS Parameters

The RADIUS parameters are described in the table below.

Table 123: RADIUS Parameters

Parameter	Description
RADIUS Accounting Parameters	
Web: Enable RADIUS Access Control CLI: enable **[EnableRADIUS]**	Enables the RADIUS application. ▪ **[0]** Disable (Default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Accounting Server IP Address CLI: accounting-server-ip **[RADIUSAccServerIP]**	Defines the IP address of the RADIUS accounting server.
Web: Accounting Port CLI: accounting-port **[RADIUSAccPort]**	Defines the port of the RADIUS accounting server. The default is 1646.
Web: RADIUS Accounting Type CLI: radius-accounting **[RADIUSAccountingType]**	Determines when the RADIUS accounting messages are sent to the RADIUS accounting server. ▪ **[0]** At Call Release = (Default) Sent at call release only. ▪ **[1]** At Connect & Release = Sent at call connect and release.

Parameter	Description
	▪ **[2]** At Setup & Release = Sent at call setup and release.
Web: AAA Indications CLI: aaa-indications **[AAAIndications]**	Determines the Authentication, Authorization and Accounting (AAA) indications. ▪ **[0]** None = (Default) No indications. ▪ **[3]** Accounting Only = Only accounting indications are used.
General RADIUS Parameters	
Web: Use RADIUS for Web/Telnet Login CLI: enable-mgmt-login **[WebRADIUSLogin]**	Enables RADIUS queries for Web and Telnet login authentication. When enabled, logging into the device's Web and Telnet embedded servers is done through a RADIUS server. The device communicates with a user-defined RADIUS server and verifies the given username and password against a remote database, in a secure manner. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For RADIUS login authentication to function, you also need to set the following parameters: ✓ EnableRADIUS = 1 (Enable) ✓ WebAuthMode = 0 (Basic Mode) ▪ RADIUS authentication requires HTTP basic authentication, where the username and password are transmitted in clear text over the network. Therefore, it's recommended to set the HTTPSOnly parameter to 1 to force the use of HTTPS, since the transport is encrypted. ▪ When RADIUS authentication is used to to log in to the CLI, only the primary Web User Account, which has Security Administration access level, can access the device's CLI (see "Configuring Web User Accounts" on page 42).
Web: RADIUS Authentication Server IP Address CLI: auth-server-ip **[RADIUSAuthServerIP]**	Defines the IP address of the RADIUS authentication server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Authentication Server Port CLI: auth-server-port **[RADIUSAuthPort]**	Defines the port of the RADIUS Authentication Server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Shared Secret CLI: shared-secret **[SharedSecret]**	Defines the 'Secret' used to authenticate the device to the RADIUS server. This should be a cryptically strong password.
RADIUS Authentication Parameters	
Web: Default Access Level CLI: default-access-level **[DefaultAccessLevel]**	Defines the default access level for the device when the RADIUS (authentication) response doesn't include an access level attribute. The valid range is 0 to 255. The default is 200 (i.e., Security Administrator).
Web: Device Behavior Upon RADIUS Timeout CLI: timeout-behavior **[BehaviorUponRadiusTimeout]**	Defines the device's response upon a RADIUS timeout. ▪ **[0]** Deny Access = Denies access. ▪ **[1]** Verify Access Locally = (Default) Checks password locally.
Web: Local RADIUS Password Cache Mode CLI: local-cache-mode **[RadiusLocalCacheMode]**	Determines the device's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the username and password (verified by the RADIUS server). ▪ **[0]** Absolute Expiry Timer = When you access a Web page, the timeout doesn't reset, instead it continues decreasing. ▪ **[1]** Reset Timer Upon Access = (Default) Upon each access to a Web page,

Parameter	Description
	the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Web: Local RADIUS Password Cache Timeout CLI: local-cache-timeout **[RadiusLocalCacheTimeout]**	Defines the time (in seconds) the locally stored username and password (verified by the RADIUS server) are valid. When this time expires, the username and password become invalid and a must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFF. The default is 300 (5 minutes). ▪ **[-1]** = Never expires. ▪ **[0]** = Each request requires RADIUS authentication.
Web: RADIUS VSA Vendor ID CLI: vsa-vendor-id **[RadiusVSAVendorID]**	Defines the vendor ID that the device accepts when parsing a RADIUS response packet. The valid range is 0 to 0xFFFFFFFF. The default is 5003.
Web: RADIUS VSA Access Level Attribute CLI: vsa-access-level **[RadiusVSAAccessAttribute]**	Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet. The valid range is 0 to 255. The default is 35.
[MaxRADIUSSessions]	Defines the number of concurrent calls that can communicate with the RADIUS server (optional). The valid range is 0 to 240. The default is 240.
[RADIUSRetransmission]	Defines the number of retransmission retries. The valid range is 1 to 10. The default is 3.
[RadiusTO]	Defines the time interval (measured in seconds) that the device waits for a response before a RADIUS retransmission is issued. The valid range is 1 to 30. The default is 10.

41.5 Quality of Experience Parameters

The Quality of Experience (QoE) parameters are described in the table below.

Table 124: Quality of Experience Parameters

Parameter	Description
SEM Parameters	
Web: Server IP CLI: configure voip/qoe configuration/server-ip **[QOEServerIP]**	Defines the IP address of AudioCodes Session Experience Manager (SEM) server to where the quality experience reports are sent. **Note:** For this parameter to take effect, a device reset is required.
Web: Port CLI: configure voip/qoe configuration/port **[QOEPort]**	Defines the port of the SEM server. The valid value range is 0 to 65534. The default is 5000.
Web: Interface Name CLI: configure voip/qoe configuration/interface-name **[QOEInterfaceName]**	Defines the IP network interface on which the quality experience reports are sent. The default is the OAMP interface. **Note:** For this parameter to take effect, a device reset is required.
Quality of Experience Profile Table	

Parameter	Description
Web: Quality of Experience Profile CLI: configure voip/qoe qoe-profile **[QOEProfile]**	This table parameter defines Quality of Experience Profiles. The format of the ini file table parameter is as follows: [QOEProfile] FORMAT QOEProfile_Index = QOEProfile_Name, QOEProfile_SensitivityLevel; [\QOEProfile] For a detailed description of this table, see "Configuring Quality of Experience Profiles" on page 155.
Quality of Experience Color Rules Table	
Web: Quality of Experience Color Rules CLI: configure voip/qoe qoe-profile qoe-color-rules **[QOEColorRules]**	This table parameter defines Quality of Experience Color Rules. The format of the ini file table parameter is as follows: [QOEColorRules] FORMAT QOEColorRules_Index = QOEColorRules_QoeProfile, QOEColorRules_ColorRuleIndex, QOEColorRules_monitoredParam, QOEColorRules_direction, QOEColorRules_profile, QOEColorRules_GreenYellowThreshold, QOEColorRules_GreenYellowHysteresis, QOEColorRules_YellowRedThreshold, QOEColorRules_YellowRedHysteresis; [\QOEColorRules] For a detailed description of this table, see "Configuring Quality of Experience Profiles" on page 155.
Bandwidth Profile Table	
Web: Bandwidth Profile CLI: configure voip/qoe bw-profile **[BWProfile]**	This table parameter defines Bandwidth Profiles. The format of the ini file table parameter is as follows: [BWProfile] FORMAT BWProfile_Index = BWProfile_Name, BWProfile_EgressAudioBandwidth, BWProfile_IngressAudioBandwidth, BWProfile_EgressVideoBandwidth, BWProfile_IngressVideoBandwidth, BWProfile_TotalEgressBandwidth, BWProfile_TotalIngressBandwidth, BWProfile_WarningThreshold, BWProfile_hysteresis, BWProfile_GenerateAlarms; [\BWProfile] For a detailed description of this table, see "Configuring Bandwidth Profiles" on page 159. **Note:** For this parameter to take effect, a device reset is required.
Media Enhancement Profile Table	
Web: Media Enhancement Profile CLI: configure voip/qoe media-enhancement **[MediaEnhancementProfile]**	This table parameter defines Media Enhancement Profiles. The format of the ini file table parameter is as follows: [MediaEnhancementProfile] FORMAT MediaEnhancementProfile_Index = MediaEnhancementProfile_ProfileName; [\MediaEnhancementProfile] For a detailed description of this table, see "Configuring Media Enhancement Profiles" on page 161.
Media Enhancement Rules Table	
Web: Media Enhancement Rules CLI: configure voip/qoe media-enhancement-rules **[MediaEnhancementRules]**	This table parameter defines Media Enhancement Rules. The format of the ini file table parameter is as follows: [MediaEnhancementRules] FORMAT MediaEnhancementRules_Index = MediaEnhancementRules_MediaEnhancementProfile, MediaEnhancementRules_RuleIndex, MediaEnhancementRules_Trigger, MediaEnhancementRules_Color, MediaEnhancementRules_ActionRule,

Parameter	Description
	MediaEnhancementRules_ActionValue; [\MediaEnhancementRules] For a detailed description of this table, see "Configuring Media Enhancement Profiles" on page 161.

41.6 Control Network Parameters

41.6.1 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

Table 125: Proxy, Registration and Authentication SIP Parameters

Parameter	Description
IP Group Table	
Web: IP Group Table CLI: configure voip > voip-network ip-group **[IPGroup]**	This table configures IP Groups. The ini file format of this parameter is as follows: [IPGroup] FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Description, IPGroup_ProxySetId, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_EnableSurvivability, IPGroup_ServingIPGroup, IPGroup_SipReRoutingMode, IPGroup_AlwaysUseRouteTable, IPGroup_RoutingMode, IPGroup_SRD, IPGroup_MediaRealm, IPGroup_ClassifyByProxySet, IPGroup_ProfileId, IPGroup_MaxNumOfRegUsers, IPGroup_InboundManSet, IPGroup_OutboundManSet, IPGroup_RegistrationMode, IPGroup_AuthenticationMode, IPGroup_MethodList, IPGroup_EnableSBCClientForking, IPGroup_SourceUriInput, IPGroup_DestUriInput, IPGroup_ContactName, IPGroup_Username, IPGroup_Password, IPGroup_UUIFormat, IPGroup_QOEProfile, IPGroup_BWProfile, IPGroup_MediaEnhancementProfile, IPGroup_AlwaysUseSourceAddr; [/IPGroup] For a description of this table, see "Configuring IP Groups" on page 174. **Note:** For this parameter to take effect, a device reset is required.
Account Table	
Web: Account Table CLI: configure voip > sip-definition account **[Account]**	This table parameter configures the Account table for registering and/or authenticating (digest) IP Groups (e.g., an IP-PBX) to another IP Group (e.g., an Internet Telephony Service Provider - ITSP). The format of the ini file table parameter is as follows: [Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroup, Account_ServingIPGroup, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType; [\Account] For a detailed description of this table, see "Configuring Registration Accounts" on page 187.
Proxy Registration Parameters	
Web: Use Default Proxy CLI: enable-proxy **[IsProxyUsed]**	Enables the use of a SIP proxy server. ▪ **[0]** No = (Default) Proxy isn't used and instead, the internal routing table is used. ▪ **[1]** Yes = Proxy server is used. Define the IP address of the proxy server in the Proxy Sets table (see "Configuring Proxy Sets" on page 182). **Note:** If you are not using a proxy server, you must configure routing rules to route the call.

Parameter	Description
Web: Proxy Name CLI: proxy-name **[ProxyName]**	Defines the Home Proxy domain name. If specified, this name is used as the Request-URI in REGISTER, INVITE and other SIP messages, and as the host part of the To header in INVITE messages. If not specified, the Proxy IP address is used instead. The valid value is a string of up to 49 characters. **Note:** This parameter functions together with the UseProxyIPasHost parameter.
Web: Use Proxy IP as Host CLI: use-proxy-ip-as-host **[UseProxyIPasHost]**	Enables the use of the proxy server's IP address (in dotted-decimal notation) as the host name in SIP From and To headers in REGISTER requests. ▪ **[0]** Disable (default) ▪ **[1]** Enable If this parameter is disabled and the device registers to an IP Group (i.e., proxy server), it uses the string configured by the ProxyName parameter as the host name in the REGISTER's Request-URI and uses the string configured by the IP Group table parameter, SIPGroupName as the host name in the To and From headers. If the IP Group is configured with a Proxy Set that has multiple IP addresses, all the REGISTER messages sent to these proxies are sent with the same host name. **Note:** If this parameter is disabled and the ProxyName parameter is not configured, the proxy's IP address is used as the host name in the REGISTER Request-URI.
Web: Redundancy Mode CLI: redundancy-mode **[ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy. ▪ **[0]** Parking = (Default) The device continues working with a redundant (now active) Proxy until the next failure, after which it works with the next redundant Proxy. ▪ **[1]** Homing = The device always tries to work with the primary Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Note:** To use this Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2.
Web: Proxy IP List Refresh Time CLI: proxy-ip-lst-rfrsh-time **[ProxyIPListRefreshTime]**	Defines the time interval (in seconds) between each Proxy IP list refresh. The range is 5 to 2,000,000. The default interval is 60.
Web: Always Use Proxy CLI: always-use-proxy **[AlwaysSendToProxy]**	Determines whether the device sends SIP messages and responses through a Proxy server. ▪ **[0]** Disable = (Default) Use standard SIP routing rules. ▪ **[1]** Enable = All SIP messages and responses are sent to the Proxy server. **Note:** This parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1).
Web: DNS Query Type CLI: dns-query **[DNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to resolve Proxy and Registrar servers and to resolve all domain names that appear in the SIP Contact and Record-Route headers. ▪ **[0]** A-Record = (Default) No NAPTR or SRV queries are performed. ▪ **[1]** SRV = If the Proxy/Registrar IP address parameter, Contact/Record-Route headers, or IP address configured in the routing tables contain a domain name, an SRV query is performed. The device uses the first host name received from the SRV query.

Parameter	Description
	The device then performs a DNS A-record query for the host name to locate an IP address. - **[2]** NAPTR = An NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. **Notes:** - If the Proxy/Registrar IP address parameter, the domain name in the Contact/Record-Route headers, or the IP address configured in the routing tables contain a domain name with a port definition, the device performs a regular DNS A-record query. - If a specific Transport Type is configured, a NAPTR query is not performed. - To enable NAPTR/SRV queries for Proxy servers only, use the global parameter ProxyDNSQueryType, or use the proxy Set table.
Web: Proxy DNS Query Type CLI: proxy-dns-query **[ProxyDNSQueryType]**	Global parameter that defines the DNS query record type for resolving the Proxy server's configured domain name (FQDN) into an IP address. - **[0]** A-Record (default) = A-record DNS query. - **[1]** SRV = If the Proxy IP address parameter contains a domain name without port definition (e.g., ProxyIP = domain.com), an SRV query is performed. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Thus, if the first SRV query returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed. - **[2]** NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Notes:** - This functionality can be configured per Proxy Set in the Proxy Set table (see "Configuring Proxy Sets" on page 182). - When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Web: Password CLI: password-4-auth **[Password]**	Defines the password for Basic/Digest authentication with a Proxy/Registrar server. A single password is used for all device ports. The default is 'Default_Passwd'.
Web: Cnonce CLI: cnonce-4-auth **[Cnonce]**	Defines the Cnonce string used by the SIP server and client to provide mutual authentication. The value is free format, i.e., 'Cnonce = 0a4f113b'. The default is 'Default_Cnonce'.
Web: Mutual Authentication Mode CLI: mutual-authentication **[MutualAuthenticationMode]**	Determines the device's mode of operation when Authentication and Key Agreement (AKA) Digest Authentication is used. - **[0]** Optional = (Default) Incoming requests that don't include AKA authentication information are accepted. - **[1]** Mandatory = Incoming requests that don't include AKA authentication information are rejected.
Proxy IP Table	

Parameter	Description
Web: Proxy IP Table CLI: configure voip > voip-network proxy-ip **[ProxyIP]**	This table parameter defines the Proxy Set table with Proxy Set IDs, each with up to 10 Proxy server IP addresses (or FQDN). The format of the ini file table parameter is as follows: [ProxyIP] FORMAT ProxyIp_Index = ProxyIp_IpAddress, ProxyIp_TransportType, ProxyIp_ProxySetId; [\ProxyIP] For a description of this table, see "Configuring Proxy Sets" on page 182. To configure the Proxy Set attributes (such as Proxy Load Balancing) in the ini file, use the ProxySet parameter.
Proxy Set Table	
Web: Proxy Set Table CLI: configure voip > voip-network proxy-set **[ProxySet]**	This table parameter defines the Proxy Set ID table. This includes, for example, Proxy keep-alive and load balancing and redundancy mechanisms. The format of the ini file table parameter is as follows: [ProxySet] FORMAT ProxySet_Index = ProxySet_ProxyName, ProxySet_EnableProxyKeepAlive, ProxySet_ProxyKeepAliveTime, ProxySet_ProxyLoadBalancingMethod, ProxySet_IsProxyHotSwap, ProxySet_SRD, ProxySet_ClassificationInput, ProxySet_ProxyRedundancyMode, ProxySet_DNSResolveMethod, ProxySet_KeepAliveFailureResp; [\ProxySet] For a description of this table, see "Configuring Proxy Sets" on page 182. For configuring the IP addresses per Proxy Set in the ini file, use the ProxyIP parameter.
Registrar Parameters	
Web: Registration Time CLI: registration-time **[RegistrationTime]**	Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. This parameter also defines the time interval between Keep-Alive messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER). Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider. The valid range is 10 to 2,000,000. The default is 180.
Web: Re-registration Timing [%] CLI: re-registration-timing **[RegistrationTimeDivider]**	Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server. The valid range is 50 to 100. The default is 50. For example: If this parameter is set to 70% and the Registration Expires time is 3600, the device re-sends its registration request after 3600 x 70% (i.e., 2520 sec). **Notes**: ▪ This parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0.
Web: Registration Retry Time CLI: registration-retry-time **[RegistrationRetryTime]**	Defines the time interval (in seconds) after which a registration request is re-sent if registration fails with a 4xx response or if there is no response from the Proxy/Registrar server. The default is 30 seconds. The range is 10 to 3600.
Web: Registration Time Threshold	Defines a threshold (in seconds) for re-registration timing. If this

Parameter	Description
CLI: registration-time-thres **[RegistrationTimeThreshold]**	parameter is greater than 0, but lower than the computed re-registration timing (according to the parameter RegistrationTimeDivider), the re-registration timing is set to the following: timing set by the Registration server in the SIP Expires header minus the value of the parameter RegistrationTimeThreshold. The valid range is 0 to 2,000,000. The default is 0.
Web: Re-register On INVITE Failure CLI: reg-on-invite-fail **[RegisterOnInviteFailure]**	Enables immediate re-registration if no response is received for an INVITE request sent by the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device immediately expires its re-registration timer and commences re-registration to the same Proxy upon any of the following scenarios: ▪ The response to an INVITE request is 407 (Proxy Authentication Required) without an authentication header included. ▪ The remote SIP UA abandons a call before the device has received any provisional response (indicative of an outbound proxy server failure). ▪ The remote SIP UA abandons a call and the only provisional response the device has received for the call is 100 Trying (indicative of a home proxy server failure, i.e., the failure of a proxy in the route after the outbound proxy). ▪ The device terminates a call due to the expiration of RFC 3261 Timer B or due to the receipt of a 408 (Request Timeout) response and the device has not received any provisional response for the call (indicative of an outbound proxy server failure). ▪ The device terminates a call due to the receipt of a 408 (Request Timeout) response and the only provisional response the device has received for the call is the 100 Trying provisional response (indicative of a home proxy server failure).
Web: ReRegister On Connection Failure CLI: reg-on-conn-failure **[ReRegisterOnConnectionFailure]**	Enables the device to perform SIP re-registration upon TCP/TLS connection failure. ▪ **[0]** Disable (default) ▪ **[1]** Enable
CLI: expl-un-reg **[UnregistrationMode]**	Enables the device to perform explicit unregisters. ▪ **[0]** Disable (default) ▪ **[1]** Enable = The device sends an asterisk ("*") value in the SIP Contact header, instructing the Registrar server to remove all previous registration bindings. The device removes SIP User Agent (UA) registration bindings in a Registrar, according to RFC 3261. Registrations are soft state and expire unless refreshed, but they can also be explicitly removed. A client can attempt to influence the expiration interval selected by the Registrar. A UA requests the immediate removal of a binding by specifying an expiration interval of "0" for that contact address in a REGISTER request. UA's should support this mechanism so that bindings can be removed before their expiration interval has passed. Use of the "*" Contact header field value allows a registering UA to remove all bindings associated with an address-of-record (AOR) without knowing their precise values. **Note:** The REGISTER-specific Contact header field value of "*" applies to all registrations, but it can only be used if the Expires header field is present with a value of "0".
Web: Add Empty Authorization Header CLI: add-empty-author-hdr	Enables the inclusion of the SIP Authorization header in initial

Parameter	Description
[EmptyAuthorizationHeader]	registration (REGISTER) requests sent by the device. ▪ **[0]** Disable (default) ▪ **[1]** Enable The Authorization header carries the credentials of a user agent (UA) in a request to a server. The sent REGISTER message populates the Authorization header with the following parameters: ▪ username - set to the value of the private user identity ▪ realm - set to the domain name of the home network ▪ uri - set to the SIP URI of the domain name of the home network ▪ nonce - set to an empty value ▪ response - set to an empty value For example: `Authorization: Digest username=alice private@home1.net, realm="home1.net", nonce="", response="e56131d19580cd833064787ecc"` **Note:** This registration header is according to the IMS 3GPP TS24.229 and PKT-SP-24.220 specifications.

Web: Add initial Route Header
CLI: add-init-rte-hdr
[InitialRouteHeader]
	Enables the inclusion of the SIP Route header in initial registration or re-registration (REGISTER) requests sent by the device.
	▪ **[0]** Disable (default)
	▪ **[1]** Enable
	When the device sends a REGISTER message, the Route header includes either the Proxy's FQDN, or IP address and port according to the configured Proxy Set, for example:
	`Route: <sip:10.10.10.10;lr;transport=udp>`
	or
	`Route: <sip: pcscf-gm.ims.rr.com;lr;transport=udp>`

[UsePingPongKeepAlive]	Enables the use of the carriage-return and line-feed sequences (CRLF) Keep-Alive mechanism, according to RFC 5626 "Managing Client-Initiated Connections in the Session Initiation Protocol (SIP)" for reliable, connection-orientated transport types such as TCP. ▪ **[0]** Disable (default) ▪ **[1]** Enable The SIP user agent/client (i.e., device) uses a simple periodic message as a keep-alive mechanism to keep their flow to the proxy or registrar alive (used for example, to keep NAT bindings open). For connection-oriented transports such as TCP/TLS this is based on CRLF. This mechanism uses a client-to-server "ping" keep-alive and a corresponding server-to-client "pong" message. This ping-pong sequence allows the client, and optionally the server, to tell if its flow is still active and useful for SIP traffic. If the client does not receive a pong in response to its ping, it declares the flow "dead" and opens a new flow in its place. In the CRLF Keep-Alive mechanism the client periodically (defined by the PingPongKeepAliveTime parameter) sends a double-CRLF (the "ping") then waits to receive a single CRLF (the "pong"). If the client does not receive a "pong" within an appropriate amount of time, it considers the flow failed. **Note:** The device sends a CRLF message to the Proxy Set only if the Proxy Keep-Alive feature (EnableProxyKeepAlive parameter) is enabled and its transport type is set to TCP or TLS. The device first sends a SIP OPTION message to establish the TCP/TLS connection and if it receives any SIP response, it continues sending the CRLF keep-alive sequences.

Parameter	Description
[PingPongKeepAliveTime]	Defines the periodic interval (in seconds) after which a "ping" (double-CRLF) keep-alive is sent to a proxy/registrar, using the CRLF Keep-Alive mechanism. The default range is 5 to 2,000,000. The default is 120. The device uses the range of 80-100% of this user-defined value as the actual interval. For example, if the parameter value is set to 200 sec, the interval used is any random time between 160 to 200 seconds. This prevents an "avalanche" of keep-alive by multiple SIP UAs to a specific server.

41.6.2 Network Application Parameters

The SIP network application parameters are described in the table below.

Table 126: SIP Network Application Parameters

Parameter	Description
Signaling Routing Domain Table	
Web: SRD Settings CLI: configure voip > voip-network srd **[SRD]**	This table parameter configures the Signaling Routing Domains (SRD). The format of the ini file table parameter is as follows: [SRD] FORMAT SRD_Index = SRD_Name, SRD_MediaRealm, SRD_IntraSRDMediaAnchoring, SRD_BlockUnRegUsers, SRD_MaxNumOfRegUsers, SRD_EnableUnAuthenticatedRegistrations; [\SRD] For a detailed description of this table, see "Configuring SRDs" on page 169.
SIP Interface Table	
Web: SIP Interface Table CLI: configure voip > voip-network sip-interface **[SIPInterface]**	This table parameter configures SIP Interfaces. The SIP Interface represents a SIP signaling entity, comprising ports (UDP, TCP, and TLS) and associated with a specific IP interface and an SRD. The format of the ini file table parameter is as follows: [SIPInterface] FORMAT SIPInterface_Index = SIPInterface_InterfaceName, SIPInterface_NetworkInterface, SIPInterface_ApplicationType, SIPInterface_UDPPort, SIPInterface_TCPPort, SIPInterface_TLSPort, SIPInterface_SRD, SIPInterface_MessagePolicy, SIPInterface_TLSMutualAuthentication, SIPInterface_TCPKeepaliveEnable, SIPInterface_ClassificationFailureResponseType; [\SIPInterface] For a detailed description of this table, see "Configuring SIP Interfaces" on page 171.
[TCPKeepAliveTime]	Defines the interval (in sec) between the last data packet sent and the first keep-alive probe to send. The valid value is 10 to 65,000. The default is 60. **Notes:** ▪ Simple ACKs such as keepalives are not considered data packets. ▪ TCP keepalive is enabled per SIP Interface in the SIP Interface table.
[TCPKeepAliveInterval]	Defines the interval (in sec) between consecutive keep-alive probes, regardless of what the connection has exchanged in the meantime.

Parameter	Description
	The valid value is 10 to 65,000. The default is 10. **Note:** TCP keepalive is enabled per SIP Interface in the SIP Interface table.
[TCPKeepAliveRetry]	Defines the number of unacknowledged keep-alive probes to send before considering the connection down. The valid value is 1 to 100. The default is 5. **Note:** TCP keepalive is enabled per SIP Interface in the SIP Interface table.
NAT Translation Table	
Web: NAT Translation Table CLI: configure voip > voip-network NATTranslation **[NATTranslation]**	This table parameter defines NAT rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses. This allows, for example, the separation of VoIP traffic between different ISTP's, and topology hiding (of internal IP addresses to the "public" network). Each IP interface (configured in the Interface table - InterfaceTable parameter) can be associated with a NAT rule in this table, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range). The format of the ini file table parameter is as follows: [NATTranslation] FORMAT NATTranslation_Index = NATTranslation_SourceIPInterfaceName, NATTranslation_TargetIPAddress, NATTranslation_SourceStartPort, NATTranslation_SourceEndPort, NATTranslation_TargetStartPort, NATTranslation_TargetEndPort; [\NATTranslation] For a detailed description of this table, see "Configuring NAT Translation per IP Interface" on page 111.
Media Realm Table	
Web: Media Realm Table CLI: configure voip > voip-network realm **[CpMediaRealm]**	This table parameter defines Media Realms. The Media Realm table allows you to divide a Media-type interface (defined in the Interface table) into several realms, where each realm is specified by a UDP port range. The format of the ini file table parameter is as follows: [CpMediaRealm] FORMAT CpMediaRealm_Index = CpMediaRealm_MediaRealmName, CpMediaRealm_IPv4IF, CpMediaRealm_IPv6IF, CpMediaRealm_PortRangeStart, CpMediaRealm_MediaSessionLeg, CpMediaRealm_PortRangeEnd, CpMediaRealm_IsDefault, CpMediaRealm_QoeProfile, CpMediaRealm_BWProfile; [\CpMediaRealm] For a detailed description of this table, see "Configuring Media Realms" on page 164.
Remote Media Subnet Table	
Web: Remote Media Subnet CLI: configure voip > voip-network realm remotemediasubnet **[SubRealm]**	This table parameter defines Remote Media Subnets. The format of the ini file table parameter is as follows: [RemoteMediaSubnet] FORMAT RemoteMediaSubnet_Index = RemoteMediaSubnet_Realm, RemoteMediaSubnet_RemoteMediaSubnetIndex, RemoteMediaSubnet_RemoteMediaSubnetName, RemoteMediaSubnet_PrefixLength, RemoteMediaSubnet_AddressFamily, RemoteMediaSubnet_DstIPAddress, RemoteMediaSubnet_QOEProfileName, RemoteMediaSubnet_BWProfileName; [\RemoteMediaSubnet] For a detailed description of this table, see "Configuring Remote Media Subnets" on page 166.

41.7 General SIP Parameters

The general SIP parameters are described in the table below.

Table 127: General SIP Parameters

Parameter	Description
Web: SIP 408 Response upon non-INVITE CLI: enbl-non-inv-408 **[EnableNonInvite408Reply]**	Enables the device to send SIP 408 responses (Request Timeout) upon receipt of non-INVITE transactions. Disabling this response complies with RFC 4320/4321. By default, and in certain circumstances such as a timeout expiry, the device sends a SIP 408 Request Timeout in response to non-INVITE requests (e.g., REGISTER). ▪ **[0]** Disable = SIP 408 response is not sent upon receipt of non-INVITE messages (to comply with RFC 4320). ▪ **[1]** Enable = (Default) SIP 408 response is sent upon receipt of non-INVITE messages, if necessary.
Web: SIP Remote Reset CLI: sip-remote-reset **[EnableSIPRemoteReset]**	Enables a specific device action upon the receipt of a SIP NOTIFY request, where the action depends on the value received in the Event header. ▪ **[0]** Disable (default) ▪ **[1]** Enable The action depends on the Event header value: ▪ 'check-sync;reboot=false': triggers the regular Automatic Update feature (if Automatic Update has been enabled on the device) ▪ 'check-sync;reboot=true': triggers a device reset **Note:** The Event header value is proprietary to AudioCodes.
Web: Max SIP Message Length [KB] **[MaxSIPMessageLength]**	Defines the maximum size (in Kbytes) for each SIP message that can be sent over the network. The device rejects messages exceeding this user-defined size. The valid value range is 1 to 50. The default is 50.
[SIPForceRport]	Determines whether the device sends SIP responses to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the SIP Via header. ▪ **[0]** = (Default) Disabled. The device sends the SIP response to the UDP port defined in the Via header. If the Via header contains the 'rport' parameter, the response is sent to the UDP port from where the SIP request is received. ▪ **[1]** = Enabled. SIP responses are sent to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the Via header.
Web: Reject Cancel after Connect CLI: reject-cancel-after-connect **[RejectCancelAfterConnect]**	Determines whether the device accepts or rejects a SIP CANCEL request received after the receipt of a 200 OK, during an established call. ▪ **[0]** = (Default) Accepts the CANCEL, by responding with a 200 OK and terminating the call session. ▪ **[1]** = Rejects the CANCEL, by responding with a SIP 481 Call/Transaction Does Not Exist, and maintaining the call session.
Web: Verify Received RequestURI CLI: verify-rcvd-requri **[VerifyReceevedRequestUri]**	Enables the device to reject SIP requests (such as ACK, BYE, or re-INVITE) whose user part in the Request-URI is different from the user part received in the Contact header of the last sent SIP request. ▪ **[0]** Disable = (Default) Even if the user is different, the device accepts the SIP request. ▪ **[1]** Enable = If the user is different, the device rejects the SIP request (BYE is responded with 481; re-INVITE is responded with 404; ACK is ignored).
Web: Max Number of Active Calls CLI: max-nb-of--act-calls **[MaxActiveCalls]**	Defines the maximum number of simultaneous active calls supported by the device. If the maximum number of calls is reached, new calls are not established. The valid range is 1 to the maximum number of supported channels. The default value is the maximum available channels (i.e., no restriction on the maximum

Parameter	Description
	number of calls).
Web: Number of Calls Limit **[IpProfile_CallLimit,]**	Defines the maximum number of concurrent calls per IP Profile (see "Configuring IP Profiles" on page 201).

Web: QoS statistics in SIP Release Call
[QoSStatistics]
	Enables the device to include call quality of service (QoS) statistics in SIP BYE and SIP 200 OK response to BYE, using the proprietary SIP header X-RTP-Stat.
	▪ **[0]** = Disable (default)
	▪ **[1]** = Enable
	The X-RTP-Stat header provides the following statistics:
	▪ Number of received and sent voice packets
	▪ Number of received and sent voice octets
	▪ Received packet loss, jitter (in ms), and latency (in ms)
	The X-RTP-Stat header contains the following fields:
	▪ PS=<voice packets sent>
	▪ OS=<voice octets sent>
	▪ PR=<voice packets received>
	▪ OR=<voice octets received>
	▪ PL=<receive packet loss>
	▪ JI=<jitter in ms>
	▪ LA=<latency in ms>
	Below is an example of the X-RTP-Stat header in a SIP BYE message:
	`BYE sip:302@10.33.4.125 SIP/2.0`
	`Via: SIP/2.0/UDP 10.33.4.126;branch=z9hG4bKac2127550866`
	`Max-Forwards: 70`
	`From: <sip:401@10.33.4.126;user=phone>;tag=1c2113553324`
	`To: <sip:302@company.com>;tag=1c991751121`
	`Call-ID: 991750671245200001912@10.33.4.125`
	`CSeq: 1 BYE`
	`X-RTP-Stat: PS=207;OS=49680;;PR=314;OR=50240;PL=0;JI=600;LA=40;`
	`Supported: em,timer,replaces,path,resource-priority`
	`Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE`
	`User-Agent: Sip-Gateway-/v.6.2A.008.006`
	`Reason: Q.850 ;cause=16 ;text="local"`
	`Content-Length: 0`

Web: PRACK Mode CLI: prack-mode **[PrackMode]**	Determines the PRACK (Provisional Acknowledgment) mechanism mode for SIP 1xx reliable responses. ▪ **[0]** Disable ▪ **[1]** Supported (default) ▪ **[2]** Required **Notes:** ▪ The Supported and Required headers contain the '100rel' tag. ▪ The device sends PRACK messages if 180/183 responses are received with '100rel' in the Supported or Required headers.
Web: Enable Early Media CLI: early-media **[EnableEarlyMedia]**	Global parameter that enables the Early Media feature for sending media (e.g., ringing) before the call is established. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEarlyMedia). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.

Parameter	Description
Web: 183 Message Behavior CLI: 183-msg-behavior **[SIP183Behaviour]**	- **[0]** Progress = (Default) . - **[1]** Alert =
Web: Session-Expires Time CLI: session-expires-time **[SIPSessionExpires]**	Defines the numerical value sent in the Session-Expires header in the first INVITE request or response (if the call is answered). The valid range is 1 to 86,400 sec. The default is 0 (i.e., the Session-Expires header is disabled).
Web: Minimum Session-Expires CLI: min-session-expires **[MinSE]**	Defines the time (in seconds) that is used in the Min-SE header. This header defines the minimum time that the user agent refreshes the session. The valid range is 10 to 100,000. The default is 90.
Web: Session Expires Disconnect Time CLI: session-exp-disconnect-time **[SessionExpiresDisconnectTime]**	Defines a session expiry timeout. The device disconnects the session (sends a SIP BYE) if the refresher did not send a refresh request before one-third (1/3) of the session expires time, or before the time configured by this parameter (the minimum of the two). The valid range is 0 to 32 (in seconds). The default is 32.
Web: Session Expires Method CLI: session-exp-method **[SessionExpiresMethod]**	Determines the SIP method used for session-timer updates. - **[0]** Re-INVITE = (Default) Uses re-INVITE messages for session-timer updates. - **[1]** UPDATE = Uses UPDATE messages. **Notes:** - The device can receive session-timer refreshes using both methods. - The UPDATE message used for session-timer is excluded from the SDP body.
[RemoveToTagInFailureResponse]	Determines whether the device removes the 'to' header tag from final SIP failure responses to INVITE transactions. - **[0]** = (Default) Do not remove tag. - **[1]** = Remove tag.
[EnableRTCPAttribute]	Enables the use of the 'rtcp' attribute in the outgoing SDP. - **[0]** = Disable (default) - **[1]** = Enable
[OPTIONSUserPart]	Defines the user part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the configuration parameter 'Username' valueis used. A special value is 'empty', indicating that no user part in the Request-URI (host part only) is used. The valid range is a 30-character string. By default, this value is not defined.
Web: Fax Signaling Method CLI: fax-sig-method **[IsFaxUsed]**	Global parameter that defines the SIP signaling method for establishing and transmitting a fax session when the device detects a fax. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IsFaxUsed). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
[HandleG711asVBD]	Enables the handling of G.711 as a G.711 Voice Band Data (VBD) coder. - **[0]** = (Default) Disable. The device negotiates G.711 as a regular audio coder and sends an answer only with G.729 coder. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing only the G.729 coder. - **[1]** = Enable. The device assumes that the G.711 coder received in the INVITE

Parameter	Description
	SDP offer is a VBD coder. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing G.729 and G.711 VBD coders, allowing a subsequent bypass (passthrough) session if fax/modem signals are detected during the call. **Note:** This parameter is applicable only if G.711 VBD coder(s) with regular G.711 payload types 0 or 8 are configured for the device (using the CodersGroup parameter).
CLI: fax-vbd-behvr **[FaxVBDBehavior]**	Determines the device's fax transport behavior when G.711 VBD coder is negotiated at call start. ▪ **[0]** = (Default) If the device is configured with a VBD coder (see the CodersGroup parameter) and is negotiated OK at call start, then both fax and modem signals are sent over RTP using the bypass payload type (and no mid-call VBD or T.38 Re-INVITEs occur). ▪ **[1]** = If the IsFaxUsed parameter is set to 1, the channel opens with the FaxTransportMode parameter set to 1 (relay). This is required to detect mid-call fax tones and to send T.38 Re-INVITE messages upon fax detection. If the remote party supports T.38, the fax is relayed over T.38. **Notes:** ▪ If VBD coder negotiation fails at call start and if the IsFaxUsed parameter is set to 1 (or 3), then the channel opens with the FaxTransportMode parameter set to 1 (relay) to allow future detection of fax tones and sending of T.38 Re-INVITES. In such a scenario, the FaxVBDBehavior parameter has no effect. ▪ This feature can be used only if the remote party supports T.38 fax relay; otherwise, the fax fails.
[NoAudioPayloadType]	Defines the payload type of the outgoing SDP offer. The valid value range is 96 to 127 (dynamic payload type). The default is 0 (i.e. NoAudio is not supported). For example, if set to 120, the following is added to the INVITE SDP: `a=rtpmap:120 NoAudio/8000\r\n` **Note:** For incoming SDP offers, NoAudio is always supported.
Web: SIP Transport Type CLI: app-sip-transport-type **[SIPTransportType]**	Determines the default transport layer for outgoing SIP calls initiated by the device. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS (SIPS) **Notes:** ▪ It's recommended to use TLS for communication with a SIP Proxy and not for direct device-to-device communication. ▪ For received calls (i.e., incoming), the device accepts all these protocols. ▪ The value of this parameter is also used by the SAS application as the default transport layer for outgoing SIP calls.
Web: SIP UDP Local Port CLI: sip-udp-local-port **[LocalSIPPort]**	Defines the local UDP port for SIP messages. The valid range is 1 to 65534. The default is 5060.
Web: SIP TCP Local Port CLI: sip-tcp-local-port **[TCPLocalSIPPort]**	Defines the local TCP port for SIP messages. The valid range is 1 to 65535. The default is 5060.
Web: SIP TLS Local Port CLI: sip-tls-local-port **[TLSLocalSIPPort]**	Defines the local TLS port for SIP messages. The valid range is 1 to 65535. The default is 5061. **Note:** The value of this parameter must be different from the value of the parameter TCPLocalSIPPort.

Parameter	Description
Web: Enable SIPS CLI: enable-sips **[EnableSIPS]**	Enables secured SIP (SIPS URI) connections over multiple hops. ▪ **[0]** Disable (default) ▪ **[1]** Enable When the SIPTransportType parameter is set to 2 (i.e., TLS) and the parameter EnableSIPS is disabled, TLS is used for the next network hop only. When the parameter SIPTransportType is set to 2 or 1 (i.e., TCP or TLS) and EnableSIPS is enabled, TLS is used through the entire connection (over multiple hops). **Note:** If this parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails.
Web: Enable TCP Connection Reuse CLI: tcp-conn-reuse **[EnableTCPConnectionReuse]**	Enables the reuse of the same TCP connection for all calls to the same destination. ▪ **[0]** Disable = Uses a separate TCP connection for each call. ▪ **[1]** Enable = (Default) Uses the same TCP connection for all calls. **Note:** For the SAS application, this feature is configured using the SASConnectionReuse parameter.
Web: Fake TCP alias CLI: fake-tcp-alias **[FakeTCPalias]**	Enables the re-use of the same TCP/TLS connection for sessions with the same user, even if the "alias" parameter is not present in the SIP Via header of the first INVITE. ▪ **[0]** Disable = (Default) TCP/TLS connection reuse is done only if the "alias" parameter is present in the Via header of the first INVITE. ▪ **[1]** Enable **Note:** To enable TCP/TLS connection re-use, set the EnableTCPConnectionReuse parameter to 1.
Web: Reliable Connection Persistent Mode CLI: reliable-conn-persistent **[ReliableConnectionPersistentMode]**	Enables setting of all TCP/TLS connections as persistent and therefore, not released. ▪ **[0]** = (Default) Disable. All TCP connections (except those that are set to a proxy IP) are released if not used by any SIP dialog\transaction. ▪ **[1]** = Enable - TCP connections to all destinations are persistent and not released unless the device reaches 70% of its maximum TCP resources. While trying to send a SIP message connection, reuse policy determines whether live connections to the specific destination are re-used. Persistent TCP connection ensures less network traffic due to fewer setting up and tearing down of TCP connections and reduced latency on subsequent requests due to avoidance of initial TCP handshake. For TLS, persistent connection may reduce the number of costly TLS handshakes to establish security associations, in addition to the initial TCP connection set up. **Note:** If the destination is a Proxy server, the TCP/TLS connection is persistent regardless of the settings of this parameter.
Web: TCP Timeout CLI: tcp-timeout **[SIPTCPTimeout]**	Defines the Timer B (INVITE transaction timeout timer) and Timer F (non-INVITE transaction timeout timer), as defined in RFC 3261, when the SIP transport type is TCP. The valid range is 0 to 40 sec. The default is 64 * SipT1Rtx parameter value. For example, if SipT1Rtx is set to 500 msec, then the default of SIPTCPTimeout is 32 sec.
Web: SIP Destination Port CLI: sip-dst-port **[SIPDestinationPort]**	Defines the SIP destination port for sending initial SIP requests. The valid range is 1 to 65534. The default port is 5060. **Note:** SIP responses are sent to the port specified in the Via header.
Web: Use user=phone in SIP URL CLI: user=phone-in-url **[IsUserPhone]**	Determines whether the 'user=phone' string is added to the SIP URI and SIP To header. ▪ **[0]** No = 'user=phone' string is not added.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>▪ <b>[1]</b> Yes = (Default) 'user=phone' string is part of the SIP URI and SIP To header.</td></tr>
<tr><td>Web: Use user=phone in From Header
CLI: phone-in-from-hdr
<b>[IsUserPhoneInFrom]</b></td><td>Determines whether the 'user=phone' string is added to the From and Contact SIP headers.
▪ <b>[0]</b> No = (Default) Doesn't add 'user=phone' string.
▪ <b>[1]</b> Yes = 'user=phone' string is part of the From and Contact headers.</td></tr>
<tr><td>Web: Use Tel URI for Asserted Identity
CLI: uri-for-assert-id
<b>[UseTelURIForAssertedID]</b></td><td>Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers.
▪ <b>[0]</b> Disable = (Default) 'sip:'
▪ <b>[1]</b> Enable = 'tel:'</td></tr>
<tr><td>Web: Tel to IP No Answer Timeout
CLI: tel2ip-no-ans-timeout
<b>[IPAlertTimeout]</b></td><td>Defines the time (in seconds) that the device waits for a 200 OK response from the called party (IP side) after sending an INVITE message. If the timer expires, the call is released.
The valid range is 0 to 3600. The default is 180.</td></tr>
<tr><td>Web: Enable Remote Party ID
CLI: remote-party-id
<b>[EnableRPIheader]</b></td><td>Enables Remote-Party-Identity headers for calling and called numbers for Tel-to-IP calls.
▪ <b>[0]</b> Disable (default).
▪ <b>[1]</b> Enable = Remote-Party-Identity headers are generated in SIP INVITE messages for both called and calling numbers.</td></tr>
<tr><td>Web: Enable History-Info Header
CLI: hist-info-hdr
<b>[EnableHistoryInfo]</b></td><td>Enables usage of the History-Info header.
▪ <b>[0]</b> Disable (default)
▪ <b>[1]</b> Enable
<b>User Agent Client (UAC) Behavior:</b>
▪ Initial request: The History-Info header is equal to the Request-URI. If a PSTN Redirect number is received, it is added as an additional History-Info header with an appropriate reason.
▪ Upon receiving the final failure response, the device copies the History-Info as is, adds the reason of the failure response to the last entry, and concatenates a new destination to it (if an additional request is sent). The order of the reasons is as follows:
a. Q.850 Reason
b. SIP Reason
c. SIP Response code
▪ Upon receiving the final response (success or failure), the device searches for a Redirect reason in the History-Info (i.e., 3xx/4xx SIP reason). If found, it is passed to ISDN according to the following table:
<table>
<tr><th>SIP Reason Code</th><th>ISDN Redirecting Reason</th></tr>
<tr><td>302 - Moved Temporarily</td><td>Call Forward Universal (CFU)</td></tr>
<tr><td>408 - Request Timeout</td><td>Call Forward No Answer (CFNA)</td></tr>
<tr><td>480 - Temporarily Unavailable</td><td></td></tr>
<tr><td>487 - Request Terminated</td><td></td></tr>
<tr><td>486 - Busy Here</td><td>Call Forward Busy (CFB)</td></tr>
<tr><td>600 - Busy Everywhere</td><td></td></tr>
</table>
▪ If history reason is a Q.850 reason, it is translated to the SIP reason (according to the SIP-ISDN tables) and then to ISDN Redirect reason according to the table above.
<b>User Agent Server (UAS) Behavior:</b>
▪ The History-Info header is sent only in the final response.</td></tr>
</table>

Parameter	Description
	▪ Upon receiving a request with History-Info, the UAS checks the policy in the request. If a 'session', 'header', or 'history' policy tag is found, the (final) response is sent without History-Info; otherwise, it is copied from the request.

Web: Enable GRUU
CLI: enable-gruu
[EnableGRUU]
	Determines whether the Globally Routable User Agent URIs (GRUU) mechanism is used, according to RFC 5627. This is used for obtaining a GRUU from a registrar and for communicating a GRUU to a peer within a dialog.
	▪ **[0]** Disable (default)
	▪ **[1]** Enable
	A GRUU is a SIP URI that routes to an instance-specific UA and can be reachable from anywhere. There are a number of contexts in which it is desirable to have an identifier that addresses a single UA (using GRUU) rather than the group of UA's indicated by an Address of Record (AOR). For example, in call transfer where user A is talking to user B, and user A wants to transfer the call to user C. User A sends a REFER to user C:
	`REFER sip:C@domain.com SIP/2.0`
	`From: sip:A@domain.com;tag=99asd`
	`To: sip:C@domain.com`
	`Refer-To: (URI that identifies B's UA)`
	The Refer-To header needs to contain a URI that user C can use to place a call to user B. This call needs to route to the specific UA instance that user B is using to talk to user A. User B should provide user A with a URI that has to be usable by anyone. It needs to be a GRUU.
	▪ Obtaining a GRUU: The mechanism for obtaining a GRUU is through registrations. A UA can obtain a GRUU by generating a REGISTER request containing a Supported header field with the value “gruu”. The UA includes a “+sip.instance” Contact header parameter of each contact for which the GRUU is desired. This Contact parameter contains a globally unique ID that identifies the UA instance. The global unique ID is created from one of the following:
	✓ If the REGISTER is per the device's client (endpoint), it is the MAC address concatenated with the phone number of the client.
	✓ If the REGISTER is per device, it is the MAC address only.
	✓ When using TP, “User Info” can be used for registering per endpoint. Thus, each endpoint can get a unique id – its phone number. The globally unique ID in TP is the MAC address concatenated with the phone number of the endpoint.
	If the remote server doesn't support GRUU, it ignores the parameters of the GRUU. Otherwise, if the remote side also supports GRUU, the REGISTER responses contain the “gruu” parameter in each Contact header. This parameter contains a SIP or SIPS URI that represents a GRUU corresponding to the UA instance that registered the contact. The server provides the same GRUU for the same AOR and instance-id when sending REGISTER again after registration expiration. RFC 5627 specifies that the remote target is a GRUU target if its' Contact URL has the "gr" parameter with or without a value.
	▪ Using GRUU: The UA can place the GRUU in any header field that can contain a URI. It must use the GRUU in the following messages: INVITE request, its 2xx response, SUBSCRIBE request, its 2xx response, NOTIFY request, REFER request and its 2xx response.

[IsCiscoSCEMode]	Determines whether a Cisco gateway exists at the remote side. ▪ **[0]** = (Default) No Cisco gateway exists at the remote side. ▪ **[1]** = A Cisco gateway exists at the remote side. When a Cisco gateway exists at the remote side, the device must set the value of the 'annexb' parameter of the fmtp attribute in the SDP to 'no'. This logic is used if the parameter EnableSilenceCompression is set to 2 (enable without adaptation). In this case, Silence Suppression is used on the channel but not declared in the SDP.

Parameter	Description
	Note: The IsCiscoSCEMode parameter is applicable only when the selected coder is G.729.

Web: User-Agent Information
CLI: user-agent-info
[UserAgentDisplayInfo]
	Defines the string that is used in the SIP User-Agent and Server response headers. When configured, the string <UserAgentDisplayInfo value>/software version' is used, for example:
	`User-Agent: myproduct/v.6.40.010.006`
	If not configured, the default string, <AudioCodes product-name>/software version' is used, for example:
	`User-Agent: Audiocodes-Sip-Gateway-Mediant Software E-SBC/v.6.40.010.006`
	The maximum string length is 50 characters.
	Note: The software version number and preceding forward slash (/) cannot be modified. Therefore, it is recommended not to include a forward slash in the parameter's value (to avoid two forward slashes in the SIP header, which may cause problems).

Web: SDP Session Owner CLI: sdp-session-owner **[SIPSDPSessionOwner]**	Defines the value of the Owner line ('o' field) in outgoing SDP messages. The valid range is a string of up to 39 characters. The default is "AudiocodesGW". For example: `o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126`
CLI: sdp-ver-nego **[EnableSDPVersionNegotiation]**	Enables the device to ignore new SDP re-offers (from the media negotiation perspective) in certain scenarios (such as session expires). According to RFC 3264, once an SDP session is established, a new SDP offer is considered a new offer only when the SDP origin value is incremented. In scenarios such as session expires, SDP negotiation is irrelevant and thus, the origin field is not changed. Even though some SIP devices don't follow this behavior and don't increment the origin value even in scenarios where they want to re-negotiate, the device can assume that the remote party operates according to RFC 3264, and in cases where the origin field is not incremented, the device does not re-negotiate SDP capabilities. ▪ **[0]** Disable = (Default) The device negotiates any new SDP re-offer, regardless of the origin field. ▪ **[1]** Enable = The device negotiates only an SDP re-offer with an incremented origin field.
Web: Subject CLI: usr-def-subject **[SIPSubject]**	Defines the Subject header value in outgoing INVITE messages. If not specified, the Subject header isn't included (default). The maximum length is up to 50 characters.
Web: Multiple Packetization Time Format CLI: mult-ptime-format **[MultiPtimeFormat]**	Determines whether the 'mptime' attribute is included in the outgoing SDP. ▪ **[0]** None = (Default) Disabled. ▪ **[1]** PacketCable = Includes the 'mptime' attribute in the outgoing SDP - PacketCable-defined format. The mptime' attribute enables the device to define a separate packetization period for each negotiated coder in the SDP. The 'mptime' attribute is only included if this parameter is enabled even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value.
[EnablePtime]	Determines whether the 'ptime' attribute is included in the SDP. ▪ **[0]** = Remove the 'ptime' attribute from SDP. ▪ **[1]** = (Default) Include the 'ptime' attribute in SDP.
Web: 3xx Behavior CLI: 3xx-behavior **[3xxBehavior]**	Determines the device's behavior regarding call identifiers when a 3xx response is received for an outgoing INVITE request. The device can either use the same call identifiers (Call-ID, To, and From tags) or change them in the new initiated INVITE.

Parameter	Description
	▪ **[0]** Forward = (Default) Use different call identifiers for a redirected INVITE message. ▪ **[1]** Redirect = Use the same call identifiers.
Web: Enable P-Charging Vector CLI: p-charging-vector **[EnablePChargingVector]**	Enables the inclusion of the P-Charging-Vector header to all outgoing INVITE messages. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Retry-After Time CLI: retry-aftr-time **[RetryAfterTime]**	Defines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device. The time range is 0 to 3,600. The default is 0.
Web: Fake Retry After [sec] CLI: fake-retry-after **[FakeRetryAfter]**	Determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by this parameter. ▪ **[0]** Disable (default) ▪ Any positive value (in seconds) for defining the period When enabled, this feature allows the device to operate with Proxy servers that do not include the Retry-After SIP header in SIP 503 (Service Unavailable) responses to indicate an unavailable service. The Retry-After header is used with the 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable to the requesting SIP client. The device maintains a list of available proxies, by using the Keep-Alive mechanism. The device checks the availability of proxies by sending SIP OPTIONS every keep-alive timeout to all proxies. If the device receives a SIP 503 response to an INVITE, it also marks that the proxy is out of service for the defined "Retry-After" period.
Web: Enable P-Associated-URI Header CLI: p-associated-uri-hdr **[EnablePAssociatedURIHeader]**	Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From/P-Asserted-Identity headers value. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).
Web: Source Number Preference CLI: src-nb-preference **[SourceNumberPreference]**	Determines from which SIP header the source (calling) number is obtained in incoming INVITE messages. ▪ If not configured or if any string other than "From" or "Pai2" is configured, the calling number is obtained from a specific header using the following logic: **a.** P-Preferred-Identity header. **b.** If the above header is not present, then the first P-Asserted-Identity header is used. **c.** If the above header is not present, then the Remote-Party-ID header is used. **d.** If the above header is not present, then the From header is used. ▪ "**From**" = The calling number is obtained from the From header. ▪ "**Pai2**" = The calling number is obtained using the following logic: **a.** If a P-Preferred-Identity header is present, the number is obtained from it. **b.** If no P-Preferred-Identity header is present and two P-Asserted-Identity headers are present, the number is obtained from the second P-Asserted-Identity header. **c.** If only one P-Asserted-Identity header is present, the calling number is obtained from it. **Notes:**

Parameter	Description
	▪ The "From" and "Pai2" values are not case-sensitive. ▪ Once a URL is selected, all the calling party parameters are set from this header. If P-Asserted-Identity is selected and the Privacy header is set to 'id', the calling number is assumed restricted.
CLI: src-hdr-4-called-nb **[SelectSourceHeaderForCalledNumber]**	Determines the SIP header used for obtaining the called number (destination) for IP-to-Tel calls. ▪ **[0]** Request-URI header = (Default) Obtains the destination number from the user part of the Request-URI. ▪ **[1]** To header = Obtains the destination number from the user part of the To header. ▪ **[2]** P-Called-Party-ID header = Obtains the destination number from the P-Called-Party-ID header.
Web: Enable Reason Header CLI: reason-header **[EnableReasonHeader]**	Enables the usage of the SIP Reason header. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Web: Gateway Name CLI: gw-name **[SIPGatewayName]**	Defines a name for the device (e.g., device123.com). This name is used as the host part of the SIP URI in the From header. If not specified, the device's IP address is used instead (default). **Notes:** ▪ Ensure that the parameter value is the one with which the Proxy has been configured with to identify the device. ▪ This parameter can also be configured for an IP Group (in the IP Group table).
[ZeroSDPHandling]	Determines the device's response to an incoming SDP that includes an IP address of 0.0.0.0 in the SDP's Connection Information field (i.e., "c=IN IP4 0.0.0.0"). ▪ **[0]** = (Default) Sets the IP address of the outgoing SDP's c= field to 0.0.0.0. ▪ **[1]** = Sets the IP address of the outgoing SDP c= field to the IP address of the device. If the incoming SDP doesn't contain the "a=inactive" line, the returned SDP contains the "a=recvonly" line.
Web: Enable Delayed Offer CLI: delayed-offer **[EnableDelayedOffer]**	Determines whether the device sends the initial INVITE message with or without an SDP. Sending the first INVITE without SDP is typically done by clients for obtaining the far-end's full list of capabilities before sending their own offer. (An alternative method for obtaining the list of supported capabilities is by using SIP OPTIONS, which is not supported by every SIP agent.) ▪ **[0]** Disable = (Default) The device sends the initial INVITE message with an SDP. ▪ **[1]** Enable = The device sends the initial INVITE message without an SDP.
[DisableCryptoLifeTimeInSDP]	Enables the device to send "a=crypto" lines without the lifetime parameter in the SDP. For example, if the SDP contains "a=crypto:12 AES_CM_128_HMAC_SHA1_80 inline:hhQe10yZRcRcpIFPkH5xYY9R1de37ogh9G1MpvNp|2^31", it removes the lifetime parameter "2^31". ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Enable Contact Restriction CLI: contact-restriction **[EnableContactRestriction]**	Determines whether the device sets the Contact header of outgoing INVITE requests to 'anonymous' for restricted calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable
CLI: anonymous-mode **[AnonymousMode]**	Determines whether the device's IP address is used as the URI host part instead of "anonymous.invalid" in the INVITE's From header for Tel-to-IP calls. ▪ **[0]** = (Default) If the device receives a call from the Tel with blocked caller ID, it sends an INVITE with

Parameter	Description
	From: "anonymous"<anonymous@anonymous.invalid> - **[1]** = The device's IP address is used as the URI host part instead of "anonymous.invalid". This parameter may be useful, for example, for service providers who identify their SIP Trunking customers by their source phone number or IP address, reflected in the From header of the SIP INVITE. Therefore, even customers blocking their Caller ID can be identified by the service provider. Typically, if the device receives a call with blocked Caller ID from the PSTN side (e.g., Trunk connected to a PBX), it sends an INVITE to the IP with a From header as follows: From: "anonymous" <anonymous@anonymous.invalid>. This is in accordance with RFC 3325. However, when this parameter is set to 1, the device replaces the "anonymous.invalid" with its IP address.
CLI: p-assrtd-usr-name **[PAssertedUserName]**	Defines a 'representative number' (up to 50 characters) that is used as the user part of the Request-URI in the P-Asserted-Identity header of an outgoing INVITE for Tel-to-IP calls. The default is null.
[UseAORInReferToHeader]	Defines the source for the SIP URI set in the Refer-To header of outgoing REFER messages. - **[0]** = (Default) Use SIP URI from Contact header of the initial call. - **[1]** = Use SIP URI from To/From header of the initial call.
Web: Enable User-Information Usage CLI: user-inf-usage **[EnableUserInfoUsage]**	Enables the usage of the User Information, which is loaded to the device in the User Information auxiliary file. For a description on User Information, see "Loading Auxiliary Files" on page 299. - **[0]** Disable (default) - **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
[HandleReasonHeader]	Determines whether the device uses the value of the incoming SIP Reason header for Release Reason mapping. - **[0]** = Disregard Reason header in incoming SIP messages. - **[1]** = (Default) Use the Reason header value for Release Reason mapping.
[EnableSilenceSuppInSDP]	Determines the device's behavior upon receipt of SIP Re-INVITE messages that include the SDP's 'silencesupp:off' attribute. - **[0]** = (Default) Disregard the 'silecesupp' attribute. - **[1]** = Handle incoming Re-INVITE messages that include the 'silencesupp:off' attribute in the SDP as a request to switch to the Voice-Band-Data (VBD) mode. In addition, the device includes the attribute 'a=silencesupp:off' in its SDP offer. **Note:** This parameter is applicable only if the G.711 coder is used.
[EnableRport]	Enables the usage of the 'rport' parameter in the Via header. - **[0]** = Disabled (default) - **[1]** = Enabled The device adds an 'rport' parameter to the Via header of each outgoing SIP message. The first Proxy that receives this message sets the 'rport' value of the response to the actual port from where the request was received. This method is used, for example, to enable the device to identify its port mapping outside a NAT. If the Via header doesn't include the 'rport' parameter, the destination port of the response is obtained from the host part of the Via header. If the Via header includes the 'rport' parameter without a port value, the destination port of the response is the source port of the incoming request. If the Via header includes 'rport' with a port value (e.g., rport=1001), the destination port of the response is the port indicated in the 'rport' parmeter.

Parameter	Description
CLI: x-channel-header **[XChannelHeader]**	Determines whether the SIP X-Channel header is added to SIP messages for providing information on the physical channel on which the call is received or placed. ▪ **[0]** Disable = (Default) X-Channel header is not used. ▪ **[1]** Enable = X-Channel header is generated by the device and sent in INVITE messages and 180, 183, and 200 OK SIP responses. The header includes the channel, and the device's IP address. For example, 'x-channel: DS/DS1-/8;IP=192.168.13.1', where: ✓ 'DS/DS-1' is a constant string ✓ '' is ✓ '8' is the channel ✓ 'IP=192.168.13.1' is the device's IP address
[EnableRekeyAfter181]	Enables the device to send a re-INVITE with a new (different) SRTP key (in the SDP) if a SIP 181 response is received ("call is being forwarded"). The re-INVITE is sent immediately upon receipt of the 200 OK (when the call is answered). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only if SRTP is used.
[NumberOfActiveDialogs]	Defines the maximum number of concurrent, outgoing SIP REGISTER dialogs. This parameter is used to control the registration rate. The valid range is 1 to 20. The default is 20. **Notes:** ▪ Once a 200 OK is received in response to a REGISTER message, the REGISTER message is not considered in this maximum count limit. ▪ This parameter applies only to outgoing REGISTER messages (i.e., incoming is unlimited).
Web: Network Node ID CLI: net-node-id **[NetworkNodeId]**	Defines the Network Node Identifier of the device for Avaya UCID. The valid value range is1 to 0x7FFF. The default is 0. **Notes:** ▪ To use this feature, you must set this parameter to any value other than 0. ▪ To enable the generation by the device of the Avaya UCID value and adding it to the outgoing INVITE sent to the IP Group (Avaya entity), use the IP Group table's parameter 'UUI Format'.
Web: Default Release Cause CLI: dflt-release-cse **[DefaultReleaseCause]**	Defines the default Release Cause (sent to IP) for IP-to-Tel calls when the device initiates a call release and an explicit matching cause for this release is not found. The default release cause is NO_ROUTE_TO_DESTINATION (3). Other common values include NO_CIRCUIT_AVAILABLE (34), DESTINATION_OUT_OF_ORDER (27), etc. **Notes:** ▪ The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values (e.g., 3 to SIP 404, and 34 to SIP 503). ▪ For a list of SIP responses-Q.931 release cause mapping, see Alternative Routing to Trunk upon Q.931 Call Release Cause Code.
Web: Enable Microsoft Extension CLI: microsoft-ext **[EnableMicrosoftExt]**	Enables the modification of the called and calling number for numbers received with Microsoft's proprietary "ext=xxx" parameter in the SIP INVITE URI user part. Microsoft Office Communications Server sometimes uses this proprietary parameter to indicate the extension number of the called or calling party. ▪ **[0]** Disable (default) ▪ **[1]** Enable For example, if a calling party makes a call to telephone number 622125519100 Ext. 104, the device receives the SIP INVITE (from Microsoft's application) with the

Parameter	Description
	URI user part as INVITE sip:622125519100;ext=104@10.1.1.10 (or INVITE tel:622125519100;ext=104). If the parameter EnableMicrosofExt is enabled, the device modifies the called number by adding an "e" as the prefix, removing the "ext=" parameter, and adding the extension number as the suffix (e.g., **e**622125519100**104**). Once modified, the device can then manipulate the number further, using the Number Manipulation tables to leave only the last 3 digits (for example) for sending to a PBX.
[UseSIPURIForDiversionHeader]	Defines the URI format in the SIP Diversion header. ▪ **[0]** = 'tel:' (default) ▪ **[1]** = 'sip:'
[TimeoutBetween100And18x]	Defines the timeout (in msec) between receiving a 100 Trying response and a subsequent 18x response. If a 18x response is not received within this timeout period, the call is disconnected. The valid range is 0 to 180,000 (i.e., 3 minutes). The default is 32000 (i.e., 32 sec).
[IgnoreRemoteSDPMKI]	Determines whether the device ignores the Master Key Identifier (MKI) if present in the SDP received from the remote side. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Comfort Noise Generation Negotiation CLI: com-noise-gen-nego **[ComfortNoiseNegotiation]**	Enables negotiation and usage of Comfort Noise (CN). ▪ **[0]** Disable ▪ **[1]** Enable (default) The use of CN is indicated by including a payload type for CN on the media description line of the SDP. The device can use CN with a codec whose RTP time stamp clock rate is 8,000 Hz (G.711/G.726). The static payload type 13 is used. The use of CN is negotiated between sides. Therefore, if the remote side doesn't support CN, it is not used. Regardless of the device's settings, it always attempts to adapt to the remote SIP UA's request for CNG, as described below. To determine CNG support, the device uses the ComfortNoiseNegotiation parameter and the codec's SCE (silence suppression setting) using the CodersGroup parameter. If the ComfortNoiseNegotiation parameter is enabled, then the following occurs: ▪ If the device is the initiator, it sends a "CN" in the SDP only if the SCE of the codec is enabled. If the remote UA responds with a "CN" in the SDP, then CNG occurs; otherwise, CNG does not occur. ▪ If the device is the receiver and the remote SIP UA does not send a "CN" in the SDP, then no CNG occurs. If the remote side sends a "CN", the device attempts to be compatible with the remote side and even if the codec's SCE is disabled, CNG occurs. If the ComfortNoiseNegotiation parameter is disabled, then the device does not send "CN" in the SDP. However, if the codec's SCE is enabled, then CNG occurs.
CLI: sdp-ecan-frmt **[SDPEcanFormat]**	Defines the echo canceller format in the outgoing SDP. The 'ecan' attribute is used in the SDP to indicate the use of echo cancellation. ▪ **[0]** = (Default) The 'ecan' attribute appears on the 'a=gpmd' line. ▪ **[1]** = The 'ecan' attribute appears as a separate attribute. ▪ **[2]** = The 'ecan' attribute is not included in the SDP. ▪ **[3]** = The 'ecan' attribute and the 'vbd' parameter are not included in the SDP. **Note:** This parameter is applicable only when the IsFaxUsed parameter is set to 2, and for re-INVITE messages generated by the device as result of modem or fax tone detection.
Web: First Call Ringback Tone ID CLI: 1st-call-rbt-id	Defines the index of the first ringback tone in the CPT file. This option enables an Application server to request the device to play a distinctive ringback tone to the calling party according to the destination of the call. The tone is played according

Parameter	Description
[FirstCallRBTId]	to the Alert-Info header received in the 180 Ringing SIP response (the value of the Alert-Info header is added to the value of this parameter). The valid range is -1 to 1,000. The default is -1 (i.e., play standard ringback tone). **Notes:** ▪ It is assumed that all ringback tones are defined in sequence in the CPT file. ▪ In case of an MLPP call, the device uses the value of this parameter plus 1 as the index of the ringback tone in the CPT file (e.g., if this value is set to 1, then the index is 2, i.e., 1 + 1).
Web: RTP Only Mode CLI: rtp-only-mode **[RTPOnlyMode]**	Enables the device to send and receive RTP packets to and from remote endpoints without the need to establish a SIP session. The remote IP address is determined according to the Outbound IP Routing table (Prefix parameter). The port is the same port as the local RTP port (configured by the BaseUDPPort parameter and the channel on which the call is received). ▪ **[0]** Disable (default) ▪ **[1]** Transmit & Receive = Send and receive RTP packets. ▪ **[2]** Transmit Only= Send RTP packets only. ▪ **[3]** Receive Only= Receive RTP packets only. **Notes:** ▪ To configure the RTP Only mode per trunk, use the RTPOnlyModeForTrunk_x parameter. ▪ If per trunk configuration (using the RTPOnlyModeForTrunk_ID parameter) is set to a value other than the default, the RTPOnlyMode parameter value is ignored.
Web/EMS: Media IP Version Preference CLI: media-ip-ver-pref [MediaIPVersionPreference]	Global parameter that defines the preferred RTP media IP addressing version (IPv4 or IPv6) for outgoing SIP calls. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MediaIPVersionPreference). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 201.
Retransmission Parameters	
Web: SIP T1 Retransmission Timer [msec] CLI: t1-re-tx-time **[SipT1Rtx]**	Defines the time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message. The default is 500. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx. For INVITE requests, it is multiplied by two for each new retransmitted message. For all other SIP messages, it is multiplied by two until SipT2Rtx. For example, assuming SipT1Rtx = 500 and SipT2Rtx = 4000: ▪ The first retransmission is sent after 500 msec. ▪ The second retransmission is sent after 1000 (2*500) msec. ▪ The third retransmission is sent after 2000 (2*1000) msec. ▪ The fourth retransmission and subsequent retransmissions until SIPMaxRtx are sent after 4000 (2*2000) msec.
Web: SIP T2 Retransmission Timer [msec] CLI: t2-re-tx-time **[SipT2Rtx]**	Defines the maximum interval (in msec) between retransmissions of SIP messages (except for INVITE requests). The default is 4000. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx.
Web: SIP Maximum RTX CLI: sip-max-rtx **[SIPMaxRtx]**	Defines the maximum number of UDP transmissions of SIP messages (first transmission plus retransmissions). The range is 1 to 30. The default is 7.
Web: Number of RTX Before Hot-Swap	Defines the number of retransmitted INVITE/REGISTER messages before the call

Parameter	Description
CLI: nb-of-rtx-b4-hot-swap **[HotSwapRtx]**	is routed (hot swap) to another Proxy/Registrar. The valid range is 1 to 30. The default is 3. **Note:** This parameter is also used for alternative routing. If a domain name in the SBC IP-to-IP Routing table is resolved into two IP addresses, and if there is no response for HotSwapRtx retransmissions to the INVITE message that is sent to the first IP address, the device immediately initiates a call to the second IP address.
SIP Message Manipulations Table	
Web: Message Manipulations CLI: configure voip > sbc manipulations message-manipulations [MessageManipulations]	This table parameter defines manipulation rules for SIP header messages. The format of the ini file table parameter is as follows: [MessageManipulations] FORMAT MessageManipulations_Index = MessageManipulations_ManSetID, MessageManipulations_MessageType, MessageManipulations_Condition, MessageManipulations_ActionSubject, MessageManipulations_ActionType, MessageManipulations_ActionValue, MessageManipulations_RowRole; [\MessageManipulations] For example, the below configuration changes the user part of the SIP From header to 200: MessageManipulations 1 = 0, Invite.Request, , Header.From.Url.User, 2, 200, 0; For a detailed description of this table, see Configuring SIP Message Manipulation on page 193.
Message Policy Table	
Web: Message Policy Table CLI: configure voip > sbc message-policy [MessagePolicy]	This table parameter configures SIP message policy rules for blocking (blacklist) unwanted incoming SIP messages or allowing (whitelist) receipt of desired messages. The format of the ini file table parameter is as follows: [MessagePolicy] FORMAT MessagePolicy_Index = MessagePolicy_Policy, MessagePolicy_MaxMessageLength, MessagePolicy_MaxHeaderLength, MessagePoliy_MaxBodyLength, MessagePolicy_MaxNumHeaders, MessagePolicy_MaxNumBodies, MessagePolicy_SendRejection, MessagePolicy_MethodListType, MessagePolicy_MethodList, MessagePolicy_BodyListType, MessagePolicy_BodyList; [/MessagePolicy] For a detailed description of this table, see Configuring SIP Message Policy Rules.

41.8 Coders and Profile Parameters

The profile parameters are described in the table below.

Table 128: Profile Parameters

Parameter	Description
IP Profile Table	
Web: IP Profile Settings CLI: configure voip > coders-and-profiles ip-profile **[IPProfile]**	This table parameter configures the IP Profile table. Each IP Profile ID includes a set of parameters (which are typically configured separately using their individual "global" parameters). You can later assign these IP Profiles to outbound IP routing rules (Prefix parameter), inbound IP routing rules and IP Groups. The format of the ini file table parameter is as follows: [IPProfile] FORMAT IpProfile_Index = IpProfile_ProfileName, IpProfile_IpPreference,

Parameter	Description
	IpProfile_CodersGroupID, IpProfile_IsFaxUsed, IpProfile_JitterBufMinDelay, IpProfile_JitterBufOptFactor, IpProfile_IPDiffServ, IpProfile_SigIPDiffServ, IpProfile_SCE, IpProfile_RTPRedundancyDepth, IpProfile_RemoteBaseUDPPort, IpProfile_CNGmode, IpProfile_VxxTransportType, IpProfile_NSEMode, IpProfile_IsDTMFUsed, IpProfile_PlayRBTone2IP, IpProfile_EnableEarlyMedia, IpProfile_ProgressIndicator2IP, IpProfile_EnableEchoCanceller, IpProfile_CopyDest2RedirectNumber, IpProfile_MediaSecurityBehaviour, IpProfile_CallLimit, IpProfile_DisconnectOnBrokenConnection, IpProfile_FirstTxDtmfOption, IpProfile_SecondTxDtmfOption, IpProfile_RxDTMFOption, IpProfile_EnableHold, IpProfile_InputGain, IpProfile_VoiceVolume, IpProfile_AddIEInSetup, IpProfile_SBCExtensionCodersGroupID, IpProfile_MediaIPVersionPreference, IpProfile_TranscodingMode, IpProfile_SBCAllowedMediaTypes, IpProfile_SBCAllowedCodersGroupID, IpProfile_SBCAllowedVideoCodersGroupID, IpProfile_SBCAllowedCodersMode, IpProfile_SBCMediaSecurityBehaviour, IpProfile_SBCRFC2833Behavior, IpProfile_SBCAlternativeDTMFMethod, IpProfile_SBCAssertIdentity, IpProfile_AMDSensitivityParameterSuit, IpProfile_AMDSensitivityLevel, IpProfile_AMDMaxGreetingTime, IpProfile_AMDMaxPostSilenceGreetingTime, IpProfile_SBCDiversionMode, IpProfile_SBCHistoryInfoMode, IpProfile_EnableQSIGTunneling, IpProfile_SBCFaxCodersGroupID, IpProfile_SBCFaxBehavior, IpProfile_SBCFaxOfferMode, IpProfile_SBCFaxAnswerMode, IpProfile_SbcPrackMode, IpProfile_SBCSessionExpiresMode, IpProfile_SBCRemoteUpdateSupport, IpProfile_SBCRemoteReinviteSupport, IpProfile_SBCRemoteDelayedOfferSupport, IpProfile_SBCRemoteReferBehavior, IpProfile_SBCRemote3xxBehavior, IpProfile_SBCRemoteMultiple18xSupport, IpProfile_SBCRemoteEarlyMediaResponseType, IpProfile_SBCRemoteEarlyMediaSupport, IpProfile_EnableSymmetricMKI, IpProfile_MKISize, IpProfile_SBCEnforceMKISize, IpProfile_SBCRemoteEarlyMediaRTP, IpProfile_SBCRemoteSupportsRFC3960, IpProfile_SBCRemoteCanPlayRingback, IpProfile_EnableEarly183, IpProfile_EarlyAnswerTimeout, IpProfile_SBC2833DTMFPayloadType, IpProfile_SBCUserRegistrationTime, IpProfile_ResetSRTPStateUponRekey, IpProfile_AmdMode, IpProfile_SBCReliableHeldToneSource, IpProfile_GenerateSRTPKeys, IpProfile_SBCPlayHeldTone, IpProfile_SBCRemoteHoldFormat, IpProfile_SBCRemoteReplacesBehavior, IpProfile_SBCSDPPtimeAnswer, IpProfile_SBCPreferredPTime, IpProfile_SBCUseSilenceSupp, IpProfile_SBCRTPRedundancyBehavior, IpProfile_SBCPlayRBTToTransferee, IpProfile_SBCRTCPMode, IpProfile_SBCJitterCompensation, IpProfile_SBCRemoteRenegotiateOnFaxDetection, IpProfile_JitterBufMaxDelay; [\IPProfile] For a description of this table, see "Configuring IP Profiles" on page 201.

41.9 Channel Parameters

This subsection describes the device's channel parameters.

41.9.1 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

Table 129: RTP/RTCP and T.38 Parameters

Parameter	Description

Parameter	Description
Web: RTP Base UDP Port EMS: Base UDP Port **[BaseUDPport]**	Global parameter that defines the lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_RemoteBaseUDPPort). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
EMS: No Op Enable CLI: no-operation-enable **[NoOpEnable]**	Enables the transmission of RTP or T.38 No-Op packets. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.
EMS: No Op Interval **[NoOpInterval]**	Defines the time interval in which RTP or T.38 No-Op packets are sent in the case of silence (no RTP/T.38 traffic) when No-Op packet transmission is enabled. The valid range is 20 to 65,000 msec. The default is 10,000. **Note:** To enable No-Op packet transmission, use the NoOpEnable parameter.
EMS: No Op Payload Type CLI: no-operation-interval **[RTPNoOpPayloadType]**	Defines the payload type of No-Op packets. The valid range is 96 to 127 (for the range of Dynamic RTP Payload Type for all types of non hard-coded RTP Payload types, refer to RFC 3551). The default is 120. **Note:** When defining this parameter, ensure that it doesn't cause collision with other payload types.
RTP Control Protocol Extended Reports (RTCP XR) Parameters	
Web: Enable RTCP XR EMS: RTCP XR Enable CLI: voice-quality-monitoring-enable **[VQMonEnable]**	Enables voice quality monitoring and RTCP XR, according to Internet-Draft draft-ietf-sipping-rtcp-summary-13. ▪ **[0]** CE_VQMON_DISABLE = (Default) Disable ▪ **[1]** CE_VQMON_ENABLE_FULLY = Calculates voice quality metrics, uses them for QoE calculations, reports them to SEM (if configured), and sends them to remote side using RTCP XR. ▪ **[2]** CE_VQMON_ENABLE_ONLY_CALCULATE = Calculates voice quality metrics, uses them for QoE calculations, reports them to SEM (if configured), but does not send them to remote side using RTCP XR. **Note:** For this parameter to take effect, a device reset is required.
Web: Minimum Gap Size EMS: GMin **[VQMonGMin]**	Defines the voice quality monitoring - minimum gap size (number of frames). The default is 16.
Web/EMS: Burst Threshold **[VQMonBurstHR]**	Defines the voice quality monitoring - excessive burst alert threshold. The default is -1 (i.e., no alerts are issued).
Web/EMS: Delay Threshold **[VQMonDelayTHR]**	Defines the voice quality monitoring - excessive delay alert threshold. The default is -1 (i.e., no alerts are issued).
Web: R-Value Delay Threshold EMS: End of Call Rval Delay Threshold **[VQMonEOCRValTHR]**	Defines the voice quality monitoring - end of call low quality alert threshold. The default is -1 (i.e., no alerts are issued).

Parameter	Description
Web: RTCP XR Packet Interval EMS: Packet Interval CLI: rtcp-interval **[RTCPInterval]**	Defines the time interval (in msec) between adjacent RTCP XR reports. This interval starts from call establishment. Thus, the device can send RTCP XR reports during the call, in addition to at the end of the call. If the duration of the call is shorter than this interval, RTCP XR is sent only at the end of the call. The valid value range is 0 to 65,535. The default is 5,000.
Web: Disable RTCP XR Interval Randomization EMS: Disable Interval Randomization CLI: disable-RTCP-randomization **[DisableRTCPRandomize]**	Determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval. ▪ **[0]** Disable = (Default) Randomize ▪ **[1]** Enable = No Randomize
EMS: RTCP XR Collection Server Transport Type **[RTCPXRESCTransportType]**	Defines the transport layer used for outgoing SIP dialogs initiated by the device to the RTCP XR Collection Server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to [-1], the value of the SIPTransportType parameter is used.
Web: RTCP XR Collection Server EMS: Esc IP CLI: rtcp-xr-coll-srvr **[RTCPXREscIP]**	Defines the IP address of the Event State Compositor (ESC). The device sends RTCP XR reports to this server, using SIP PUBLISH messages. The address can be configured as a numerical IP address or as a domain name.
Web: RTCP XR Report Mode EMS: Report Mode CLI: rtcp-xr-rep-mode **[RTCPXRReportMode]**	Determines whether RTCP XR reports are sent to the Event State Compositor (ESC) and defines the interval at which they are sent. ▪ [0] Disable = (Default) RTCP XR reports are not sent. ▪ [1] End Call = RTCP XR reports are sent at the end of each call. ▪ [2] End Call & Periodic = RTCP XR reports are sent at the end of each call and periodically according to the RTCPInterval parameter.
Web: SBC RTCP XR Report Mode CLI: sbc-rtcpxr-report-mode [SBCRtcpXrReportMode]	Enables the sending of RTCP XR reports of QoE metrics at the end of each call session (i.e., after a SIP BYE). The RTCP XR is sent in the SIP PUBLISH message. ▪ [0] Disable (default) ▪ [1] End of Call

41.10 SBC Parameters

The SBC parameters are described in the table below.

Table 130: SBC Parameters

Parameter	Description
Web: Enable SBC CLI: enable-sbc **[EnableSBCApplication]**	Enables the Session Border Control (SBC) application. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ In addition to enabling this parameter, the number of maximum SBC/IP-to-IP sessions must be included in the Software License Key.

Parameter	Description
Web: Allow Unclassified Calls CLI: unclassified-calls **[AllowUnclassifiedCalls]**	Determines whether incoming calls that cannot be classified (i.e. classification process fails) to a Source IP Group are rejected or processed. ▪ **[0]** Reject = Call is rejected if classification fails. ▪ **[1]** Allow = (Default) If classification fails, the incoming packet is assigned to a source IP Group (and subsequently processed) as follows: ✓ The source SRD is determined according to the SIP Interface to where the SIP-initiating dialog request is sent. The source IP Group is set to the default IP Group associated with this SRD. ✓ If the source SRD is ID 0, then source IP Group ID 0 is chosen. In case of any other SRD, then the first IP Group associated with this SRD is chosen as the source IP Group or the call. If no IP Group is associated with this SRD, the call is rejected.
Web: SBC No Answer Timeout CLI: sbc-no-arelt-timeout **[SBCAlertTimeout]**	Defines the timeout (in seconds) for SBC outgoing (outbound IP routing) SIP INVITE messages. If the called IP party does not answer the call within this user-defined interval, the device disconnects the session. The device starts the timeout count upon receipt of a SIP 180 Ringing response from the called party. If no other SIP response (for example, 200 OK) is received thereafter within this timeout, the call is released. The valid range is 0 to 3600 seconds. the default is 600.
CLI: configure voip/sbc general-setting/num-of-subscribes **[NumOfSubscribes]**	Defines the maximum number of concurrent SIP SUBSCRIBE sessions permitted on the device. The valid value is any value between 0 and the maximum supported SUBSCRIBE sessions. When set to -1, the device uses the default value. For more information, contact your AudioCodes sales representative. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The maximum number of SUBSCRIBE sessions can be increased by reducing the maximum number of SBC channels in the Software License Key. For every reduced SBC session, the device gains two SUBSCRIBE sessions.
CLI: sbc-max-fwd-limit **[SBCMaxForwardsLimit]**	Defines the Max-Forwards SIP header value. The Max-Forwards header is used to limit the number of servers (such as proxies) that can forward the SIP request. The Max-Forwards value indicates the remaining number of times this request message is allowed to be forwarded. This count is decremented by each server that forwards the request. This parameter affects the Max-Forwards header in the received message as follows: ▪ If the received header's original value is 0, the message is not passed on and is rejected. ▪ If the received header's original value is less than this parameter's value, the header's value is decremented before being sent on. ▪ If the received header's original value is greater than the parameter's value, the header's value is replaced by the user-defined parameter's value. The valid value range is 1-70. The default is 10.
Web: SBC Session-Expires CLI: sbc-sess-exp-time **[SBCSessionExpires]**	Defines the SBC session refresh timer (in seconds) in the Session-Expires header of outgoing INVITE messages. The valid value range is 90 (according to RFC 4028) to 86400. The default is 180.
Web: Minimum Session-Expires CLI: min-session-expires **[SBCMinSE]**	Defines the minimum amount of time (in seconds) between session refresh requests in a dialog before the session is considered timed out. This value is conveyed in the SIP Min-SE header. The valid range is 0 (default) to 1,000,000, where 0 means that the device does

Parameter	Description
	not limit Session-Expires.
Web: User Registration Grace Time CLI: configure voip/sbc general-setting/sbc-usr-reg-grace-time **[SBCUserRegistrationGraceTime]**	Defines additional time (in seconds) to add to the registration expiry time of registered users in the device's Users Registration database. The valid value is 0 to 300 (i.e., 5 minutes). The default is 0.
Web: Handle P-Asserted-Identity CLI: p-assert-id **[SBCAssertIdentity]**	Global parameter that defines the handling of the SIP P-Asserted-Identity header. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCAssertIdentity). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Keep original user in Register **[SBCKeepContactUserinRegister]**	Determines whether the device replaces the Contact user with a unique Contact user in the outgoing message in response to a REGISTER request. ▪ **[0]** Disable = (Default) The device replaces the original Contact user with a unique Contact user, for example: ✓ Received Contact: <sip:123@domain.com> ✓ Outgoing (unique) Contact: <sip:FEU1_7_1@SBC> ▪ **[1]** Enable = The original Contact user is retained and used in the outgoing REGISTER request. **Note:** This parameter is applicable only to REGISTER messages received from User-type IP Groups and that are sent to Server-type IP Groups.
Web: SBC Remote Refer Behavior CLI: sbc-refer-bhvr **[SBCReferBehavior]**	Global parameter that defines the handling of SIP REFER requests. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCRemoteReferBehavior). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
CLI: sbc-xfer-prefix **[SBCXferPrefix]**	When the SBCReferBehavior is set to 1, the device, while interworking the SIP REFER message, adds the prefix "T~&R-" to the user part of the URI in the Refer-To header. After this, the device can receive an INVITE with such a prefix (the INVITE is sent by the UA that receives the REFER message or 302 response). If the device receives an INVITE with such a prefix, it replaces the prefix with the value defined for the SBCXferPrefix parameter. By default, no value is defined. **Note:** This feature is also applicable to 3xx redirect responses. The device adds the prefix "T~&R-" to the URI user part in the Contact header if the SBC3xxBehavior parameter is set to 1.
CLI: sbc-3xx-bhvt **[SBC3xxBehavior]**	Global parameter that defines the handling of SIP 3xx redirect responses. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCRemote3xxBehavior). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
[SBCEnforceMediaOrder]	Enables the device to include all previously negotiated media lines within the current session ('m=' line) in the SDP offer-answer exchange (RFC 3264). ▪ **[0]** Disable (default) ▪ **[1]** Enable

Parameter	Description

	For example, assume a call (audio) has been established between two endpoints and one endpoint wants to subsequently send an image in the same call session. If this parameter is enabled, the endpoint includes the previously negotiated media type (i.e., audio) with the new negotiated media type (i.e., image) in its SDP offer:
	`v=0`
	`o=bob 2890844730 2890844731 IN IP4 host.example.com`
	`s=`
	`c=IN IP4 host.example.com`
	`t=0 0`
	`m=audio 0 RTP/AVP 0`
	`m=image 12345 udptl t38`
	If this parameter is disabled, the only 'm=' line included in the SDP is the newly negotiated media (i.e., image).

Web: SBC Server Auth Mode CLI: sbc-server-auth-mode **[SBCServerAuthMode]**	Defines whether authentication of the SIP client is done locally (by the device) or by a RADIUS server. ▪ **[0]** (default) = Authentication is done by the device (locally). ▪ **[1]** = Authentication is done by the RFC 5090 compliant RADIUS server ▪ **[2]** = Authentication is done according to the Draft Sterman-aaa-sip-01 method. **Note:** Currently, option [1] is not supported.
Web: Lifetime of the nonce in seconds CLI: lifetime-of-nonce **[AuthNonceDuration]**	Defines the lifetime (in seconds) that the current nonce is valid for server-based authentication. The device challenges a message that attempts to use a server nonce beyond this period. This parameter is used to provide replay protection (i.e., ensures that old communication streams are not used in replay attacks). The valid value range is 30 to 600. The default is 300.
Web: Authentication Challenge Method CLI: auth-chlng-mthd **[AuthChallengeMethod]**	Defines the type of server-based authentication challenge. ▪ **[0]** 0 = (Default) Send SIP 401 "Unauthorized" with a WWW-Authenticate header as the authentication challenge response. ▪ **[1]** 1 = Send SIP 407 "Proxy Authentication Required" with a Proxy-Authenticate header as the authentication challenge response.
Web: Authentication Quality of Protection CLI: auth-qop **[AuthQOP]**	Defines the authentication and integrity level of quality of protection (QoP) for digest authentication offered to the client. When the device challenges a SIP request (e.g., INVITE), it sends a SIP 401 response with the Proxy-Authenticate header or WWW-Authenticate header containing the 'qop' parameter. The QoP offered in the 401 response can be 'auth', 'auth-int', both 'auth' and 'auth-int', or the 'qop' parameter can be omitted from the 401 response. In response to the 401, the client needs to send the device another INVITE with the MD5 hash of the INVITE message and indicate the selected auth type. ▪ **[0]** 0 = The device sends 'qop=auth' in the SIP response, requesting authentication (i.e., validates user by checking user name and password). This option does not authenticate the message body (i.e., SDP). ▪ **[1]** 1 = The device sends 'qop=auth-int' in the SIP response, indicating required authentication and authentication with integrity (e.g., checksum). This option restricts the client to authenticating the entire SIP message, including the body, if present. ▪ **[2]** 2 = (Default) The device sends 'qop=auth, auth-int' in the SIP response, indicating either authentication or integrity. This enables the client to choose 'auth' or 'auth-int'. If the client chooses 'auth-int', then the body is included in the authentication. If the client chooses 'auth', then the body is not authenticated. ▪ **[3]** 3 = No 'qop' parameter is offered in the SIP 401 challenge message.
Web: SBC User Registration Time	Global parameter that defines the duration (in seconds) of the periodic registrations that occur between the user and the device (the device responds

Parameter	Description
CLI: sbc-usr-reg-time **[SBCUserRegistrationTime]**	with this value to the user). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCUserRegistrationTime). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: SBC Proxy Registration Time CLI: sbc-prxy-reg-time **[SBCProxyRegistrationTime]**	Defines the duration (in seconds) for which the user is registered in the proxy database (after the device forwards the REGISTER message). When set to 0, the device sends the Expires header's value as received from the user to the proxy. The valid range is 0 to 2,000,000 seconds. The default is 0.
Web: SBC Survivability Registration Time CLI: sbc-surv-reg-time **[SBCSurvivabilityRegistrationTime]**	Defines the duration of the periodic registrations between the user and the device, when the device is in survivability state (i.e., when REGISTER requests cannot be forwarded to the proxy and are terminated by the device). When set to 0, the device uses the value set by the SBCUserRegistrationTime parameter for the device's response. The valid range is 0 to 2,000,000 seconds. The default is 0.

[SBCEnableAASTRASurvivabilityNotice]	Enables the device to notify Aastra IP phones that the device is currently operating in Survivability mode. When this occurs, the Aastra IP phones display the message, "Stand Alone Mode" on their LCD screens. Survivability mode occurs when connectivity with the WAN fails and as a result, the device enables communication between IP phone users within the LAN enterprise.
	▪ **[0]** = Disable
	▪ **[1]** = Enable
	When this feature is enabled and the SBC device is in Survivability mode, it responds to SIP REGISTER messages from the IP phones with a SIP 200 OK containing the following XML body:
	`Content-Type: application/xml`
	`<?xml version="1.0" encoding="utf-8"?>`
	`<LMIDocument version="1.0">`
	`<LocalModeStatus>`
	`   <LocalModeActive>true</LocalModeActive>`
	`   <LocalModeDisplay>StandAlone Mode</LocalModeDisplay>`
	`</LocalModeStatus>`
	`</LMIDocument>`

Web: SBC Dialog-Info Interworking CLI: configure voip/sbc general-setting/sbc-dialog-info-interwork **[EnableSBCDialogInfoInterworking]**	Enables the interworking of dialog information (parsing of call identifiers in XML body) in SIP NOTIFY messages received from a remote application server. ▪ **[0]** Disable (default) ▪ **[1]** Enable For more information, see "Interworking Dialog Information in SIP NOTIFY Messages" on page 242.
CLI: sbc-keep-call-id **[SBCKeepOriginalCallId]**	Enables the device to use the same call identification value received in incoming messages for the call identification in outgoing messages. The call identification value is contained in the SIP Call-ID header. ▪ **[0]** = (Default) Disable - the device creates a new Call-ID value for the outgoing message. ▪ **[1]** = Enable - the device uses the received Call-ID value of the incoming message in the outgoing message. **Note:** When the device sends an INVITE as a result of a REFER/3xx termination, the device always creates a new Call-ID value and ignores this parameter's settings.
Web: SBC GRUU Mode	Determines the Globally Routable User Agent (UA) URI (GRUU) support,

Parameter	Description
CLI: sbc-gruu-mode **[SBCGruuMode]**	according to RFC 5627. ▪ **[0]** None = No GRUU is supplied to users. ▪ **[1]** As Proxy = (Default) The device provides same GRUU types as the proxy provided the device's GRUU clients. ▪ **[2]** Temporary only = Supply only temporary GRUU to users. (Currently not supported.) ▪ **[3]** Public only = The device provides only public GRUU to users. ▪ **[4]** Both = The device provides temporary and public GRUU to users. (Currently not supported.) This parameter allows the device to act as a GRUU server for its SIP UA clients, providing them with public GRUU's, according to RFC 5627. The public GRUU provided to the client is denoted in the SIP Contact header parameters, "pub-gruu". Public GRUU remains the same over registration expirations. On the other SBC leg communicating with the Proxy/Registrar, the device acts as a GRUU client. The device creates a GRUU value for each of its registered clients, which is mapped to the GRUU value received from the Proxy server. In other words, the created GRUU value is only used between the device and its clients (endpoints). `Public-GRUU: sip:userA@domain.com;gr=unique-id`
Web: Bye Authentication CLI: sbc-bye-auth **[SBCEnableByeAuthentication]**	Enables authenticating a SIP BYE request before disconnecting the call. This feature prevents, for example, a scenario in which the SBC SIP client receives a BYE request from a third-party imposer assuming the identity of a participant in the call and as a consequence, the call between the first and second parties is inappropriately disconnected. ▪ **[0]** Disable (default) ▪ **[1]** Enable = The device forwards the SIP authentication response (for the BYE request) to the request sender and waits for the user to authenticate it. The call is disconnected only if the authenticating server responds with a 200 OK.
Web: SBC Enable Subscribe Trying CLI: configure voip > sbc general-setting > set sbc-subs-try **[SBCSendTryingToSubscribe]**	Enables the device to send SIP 100 Trying responses upon receipt of SUBSCRIBE or NOTIFY messages. ▪ **[0]** Disable (Default) ▪ **[1]** Enable
[SBCExtensionsProvisioningMode]	Enables SBC user registration for interoperability with BroadSoft's BroadWorks server, to provide call survivability in case of connectivity failure with the BroadWorks server. ▪ **[0]** = (Default) Normal processing of REGISTER messages. ▪ **[1]** = Registration method for BroadWorks server. In a failure scenario with BroadWorks, the device acts as a backup SIP proxy server, maintaining call continuity between the enterprise LAN users (subscribers) and between the subscribers and the PSTN (if provided). **Note:** For a detailed description of this feature, see "Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server" on page 234.
Web: SBC Direct Media CLI: sbc-direct-media **[SBCDirectMedia]**	Enables the No Media Anchoring feature (i.e., direct media) for all SBC calls, whereby SIP signaling is handled by the device without handling the RTP/SRTP (media) flow between the user agents (UA). The RTP packets do not traverse the device. Instead, the two SIP UAs establish a direct RTP/SRTP flow between one another. Signaling continues to traverse the device with minimal intermediation and involvement to enable certain SBC abilities such as routing ▪ **[0]** Disable = (Default) All calls traverse the device (i.e., no direct media). If No Media Anchoring is enabled for an SRD (in the SRD table), then calls between endpoints belonging to that SRD use No Media Anchoring. ▪ **[1]** Enable = All SBC calls use the No Media Anchoring feature (i.e., direct

Parameter	Description
	media). **Note:** For more information on No Media Anchoring, see "No Media Anchoring (Anti Tromboning)" on page 222.
SBC RTCP Mode CLI: sbc-rtcp-mode **[SBCRTCPMode]**	Global parameter that defines the handling of RTCP packets. You can also configure this functionality per specific calls, using IP Profiles (IPProfile_SBCRTCPMode). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 201. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: SBC Send Invite To All Contacts CLI: sbc-send-invite-to-all-contacts **[SBCSendInviteToAllContacts]**	Enables call forking of INVITE message received with a Request-URI of a specific contact registered in the device's database, to all users under the same AOR as the contact. ▪ **[0]** Disable (default) = Sends the INVITE only to the contact of the received Request-URI. ▪ **[1]** Enable To configure call forking initiated by the device, see "Initiating SIP Call Forking" on page 239.
Web: SBC Shared Line Registration Mode CLI: sbc-shared-line-reg-mode **[SBCSharedLineRegMode]**	Enables the termination on the device of SIP REGISTER messages from secondary lines pertaining to the Shared Line feature. ▪ **[0]** Disable = (Default) Device forwards the REGISTER messages as is (i.e., not terminated on the device). ▪ **[1]** Enable = REGISTER messages of secondary lines are terminated on the device. **Note:** The device always forwards REGISTER messages of the primary line.
Web: SBC Forking Handling Mode CLI: sbc-forking-handling-mode **[SBCForkingHandlingMode]**	Defines the handling of SIP 18x responses received due to call forking of an INVITE. ▪ **[0]** Latch On First = (Default) Only the first 18x is forwarded to the INVITE-initiating UA. If SIP 18x with SDP is received, the device opens a voice stream according to the received SDP and disregards any subsequent 18x forking responses (with or without SDP). If the first response is 180 without SDP, the device responds according to the PlayRBTone2TEL parameter and disregards the subsequent forking 18x responses. ▪ **[1]** Sequential = All 18x responses are forwarded, one at a time (sequentially) to the INVITE-initiating UA. If a 18x arrives with an offer only, then only the first offer is forwarded to the INVITE-initiating UA and subsequent 18x responses are discarded.
CLI: sbc-media-sync **[EnableSBCMediaSync]**	Enables SBC media synchronization process for calls established from SIP forking that is initiated by external proxy servers. It is possible that a call is established with the media not synchronized between the SBC legs. Media synchronization resolves this issue. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Admission Control Table	
Web: Admission Control EMS: Call Admission Control CLI: configure voip > sbc sbc-admission-control **[SBCAdmissionControl]**	This table parameter defines Call Admission Control (CAC) rules for limiting the number of allowed concurrent calls (SIP dialogs). The format of the ini file table parameter is as follows: [SBCAdmissionControl] FORMAT SBCAdmissionControl_Index = SBCAdmissionControl_AdmissionControlName, SBCAdmissionControl_LimitType, SBCAdmissionControl_IPGroupID, SBCAdmissionControl_SRDID, SBCAdmissionControl_RequestType,

Parameter	Description
	SBCAdmissionControl_RequestDirection, SBCAdmissionControl_Limit, SBCAdmissionControl_LimitPerUser, SBCAdmissionControl_Rate, SBCAdmissionControl_MaxBurst, SBCAdmissionControl_Reservation; [\SBCAdmissionControl] For a detailed description of this table, see "Configuring Admission Control" on page 244.
Allowed Audio Coders Table	
Web: Allowed Audio Coders CLI: configure voip > sbc allowed-coders-group AllowedCodersGroup0 **[AllowedCodersGroupX]**	This table parameter defines Allowed Coders Groups, which determine the audio (voice) coders that can be used for a specific SIP entity. The format of the ini file table parameter is as follows: [AllowedCodersGroupX] FORMAT AllowedCodersGroup_Index = AllowedCodersGroup_Name; [\AllowedCodersGroup] Where *X* represents the index number. For a detailed description of this table, see "Configuring Allowed Audio Coder Groups" on page 247.
Allowed Video Coders Table	
CLI: configure voip/sbc allowed-video-coders-group group-X **[AllowedVideoCodersGroupX]**	This table parameter defines Allowed Video Coders Groups, which determine the video coders that can be used for a specific SIP entity. The format of the ini file table parameter is as follows: [AllowedVideoCodersGroup0] FORMAT AllowedVideoCodersGroup_Index = AllowedVideoCodersGroup_Name; [\AllowedVideoCodersGroup] Where *X* represents the index number. For a detailed description of this table, see "Configuring Allowed Video Coder Groups" on page 248.
Classification Table	
Web: Classification Table CLI: configure voip > sbc routing classification **[Classification]**	This table parameter configures the Classification table. This table classifies incoming SIP dialogs to Source IP Groups. The format of the ini file table parameter is as follows: [Classification] FORMAT Classification_Index = Classification_ClassificationName, Classification_MessageCondition, Classification_SrcSRDID, Classification_SrcAddress, Classification_SrcPort, Classification_SrcTransportType, Classification_SrcUsernamePrefix, Classification_SrcHost, Classification_DestUsernamePrefix, Classification_DestHost, Classification_ActionType, Classification_SrcIPGroupID; [\Classification] For a detailed description of this table, see "Configuring Classification Rules" on page 249.
Condition Table	
Web: Condition Table CLI: configure voip > sbc routing condition-table **[ConditionTable]**	This table parameter configures Message Condition rules for SIP messages. [ConditionTable] FORMAT ConditionTable_Index = ConditionTable_Condition, ConditionTable_Description; [\ConditionTable] For a detailed description of this table, see "Configuring Message Condition Rules" on page 254.

Parameter	Description
SBC IP-to-IP Routing Table	
Web: IP-to-IP Routing Table CLI: configure voip > sbc routing ip2ip-routing **[IP2IPRouting]**	This table parameter configures the SBC IP-to-IP Routing table for routing incoming SIP messages such as INVITE messages to an IP destination. The format of the ini file table parameter is as follows: [IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_RouteName, IP2IPRouting_SrcIPGroupID, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_RequestType, IP2IPRouting_MessageCondition, IP2IPRouting_ReRouteIPGroupID, IP2IPRouting_Trigger, IP2IPRouting_CallSetupRulesSetId, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupID, IP2IPRouting_DestSRDID, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions, IP2IPRouting_GroupPolicy, IP2IPRouting_CostGroup; [\IP2IPRouting] For a detailed description of this table, see "Configuring SBC IP-to-IP Routing Rules" on page 255.
SBC Alternative Routing Reasons Table	
Web: SBC Alternative Routing Reasons CLI: configure voip > sbc routing sbc-alternative-routing-reasons **[SBCAlternativeRoutingReasons]**	This table parameter configures the SBC Alternative Routing Reasons table. The format of the ini file table parameter is as follows: [SBCAlternativeRoutingReasons] FORMAT SBCAlternativeRoutingReasons_Index = SBCAlternativeRoutingReasons_ReleaseCause; [\SBCAlternativeRoutingReasons] For a detailed description of this table, see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 262.
IP to IP Inbound Manipulation Table	
Web: IP to IP Inbound Manipulation CLI: configure voip > sbc manipulations ip-inbound-manipulation **[IPInboundManipulation]**	This table parameter configures the IP to IP Inbound Manipulation table. This table allows you to manipulate the SIP URI user part (source and/or destination) of the inbound SIP dialog message. The format of the ini file table parameter is as follows: [IPInboundManipulation] FORMAT IPInboundManipulation_Index = IPInboundManipulation_ManipulationName IPInboundManipulation_IsAdditionalManipulation, IPInboundManipulation_ManipulatedURI, IPInboundManipulation_ManipulationPurpose, IPInboundManipulation_SrcIPGroupID, IPInboundManipulation_SrcUsernamePrefix, IPInboundManipulation_SrcHost, IPInboundManipulation_DestUsernamePrefix, IPInboundManipulation_DestHost, IPInboundManipulation_RequestType, IPInboundManipulation_RemoveFromLeft, IPInboundManipulation_RemoveFromRight, IPInboundManipulation_LeaveFromRight, IPInboundManipulation_Prefix2Add, IPInboundManipulation_Suffix2Add; [\IPInboundManipulation] For a detailed description of this table, see "Configuring IP-to-IP Inbound Manipulations" on page 266.
IP to IP Outbound Manipulation Table	
Web: IP to IP Outbound Manipulation CLI: configure voip > sbc	This table parameter configures the IP to IP Outbound Manipulation table. This table allows you to manipulate the SIP URI user part (source and/or destination) of the outbound SIP dialog message. The format of the ini file table parameter is

Parameter	Description
manipulations ip-outbound-manipulation **[IPOutboundManipulation]**	as follows: FORMAT IPOutboundManipulation_Index = IPOutboundManipulation_ManipulationName, IPOutboundManipulation_IsAdditionalManipulation, IPOutboundManipulation_SrcIPGroupID, IPOutboundManipulation_DestIPGroupID, IPOutboundManipulation_SrcUsernamePrefix, IPOutboundManipulation_SrcHost, IPOutboundManipulation_DestUsernamePrefix, IPOutboundManipulation_DestHost, IPOutboundManipulation_RequestType, IPOutboundManipulation_ReRouteIPGroupID, IPOutboundManipulation_Trigger, IPOutboundManipulation_ManipulatedURI, IPOutboundManipulation_RemoveFromLeft, IPOutboundManipulation_RemoveFromRight, IPOutboundManipulation_LeaveFromRight, IPOutboundManipulation_Prefix2Add, IPOutboundManipulation_Suffix2Add, IPOutboundManipulation_PrivacyRestrictionMode; For a detailed description of this table, see "Configuring IP-to-IP Outbound Manipulations" on page 268.

41.11 Services

41.11.1 SIP-based Media Recording Parameters

The SIP-based media recording parameters are described in the table below.

Table 131: SIP-based Media Recording Parameters

Parameter	Description
Web: SIP Recording Application CLI: configure voip/services sip-recording general-setting/enable-sip-rec **[EnableSIPRec]**	Enables the SIP-based Media Recording feature: ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Recording Server (SRS) Destination Username CLI: configure voip/services sip-recording general-setting/siprec-server-dest-username **[SIPRecServerDestUsername]**	Defines the SIP user part for the recording server. This user part is added in the SIP To header of the INVITE message that the device sends to the recording server. The valid value is a string of up to 50 characters. By default, no user part is defined.
SIP Recording Routing Table	
Web: SIP Recording Routing table CLI: configure voip/services sip-recording sip-rec-routing **[SIPRecRouting]**	Defines SIP Recording Routing rules (calls to record). The format of the ini file table parameter is as follows: [SIPRecRouting] FORMAT SIPRecRouting_Index = SIPRecRouting_RecordedIPGroupID, SIPRecRouting_RecordedSourcePrefix, SIPRecRouting_RecordedDestinationPrefix, SIPRecRouting_PeerIPGroupID, SIPRecRouting_PeerTrunkGroupID, SIPRecRouting_Caller, SIPRecRouting_SRSIPGroupID; [\SIPRecRouting] For a detailed description of this table, see "Configuring SIP Recording Routing Rules" on page 135.

41.11.2 LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below.

Table 132: LDAP Parameters

Parameter	Description
LDAP Configuration Table	
Web: LDAP Configuration Table CLI: configure voip/ldap/ldap-configuration **[LdapConfiguration]**	Defines the LDAP servers. [LdapConfiguration] FORMAT LdapConfiguration_Index = LdapConfiguration_LdapConfServerIp, LdapConfiguration_LdapConfServerPort, LdapConfiguration_LdapConfServerMaxRespondTime, LdapConfiguration_LdapConfServerDomainName, LdapConfiguration_LdapConfPassword, LdapConfiguration_LdapConfBindDn, LdapConfiguration_LdapConfInterfaceType, LdapConfiguration_ConnectionStatus; [\LdapConfiguration] For a detailed description of this table, see "Configuring LDAP Servers" on page 139.
LDAP Servers Search DNs Table	
Web: Ldap Servers Search Dns Table CLI: configure voip/ldap/ldap-servers-search-dns/ **[LdapServersSearchDNs]**	Defines the full base path (i.e., distinguished name / DN) to the objects in the AD where the query is done, per LDAP server. [LdapServersSearchDNs] FORMAT LdapServersSearchDNs_Index = LdapServersSearchDNs_Base_Path, LdapServersSearchDNs_LdapConfigurationIndex, LdapServersSearchDNs_SearchDnInternalIndex; [\LdapServersSearchDNs] For a detailed description of this table, see "Configuring LDAP Servers" on page 139.
Web: LDAP Service CLI: configure voip/ldap/enable **[LDAPServiceEnable]**	Enables the LDAP feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
CLI: search-dns-in-parallel **[LDAPSearchDNsinParallel**]	Defines the method of how the device queries the DN object within each LDAP server. ▪ **[0]** Sequential = (Default) The query is done in each DN object, one by one, until a result is returned. ▪ **[1]** Parallel = The query is done in all DN objects at the same time.
Web: LDAP Search Server Method CLI: ldap-search-server-method **[LDAPSearchServerMethod]**	Defines the method of how the device queries between two LDAP severs. ▪ **[0]** Sequential = The device first queries one of the LDAP servers, and if the DN object is not found, it queries the second LDAP server. ▪ **[1]** Parallel = (Default) The device queries the LDAP servers at the same time.
[LDAPDebugMode]	Determines whether to enable the LDAP task debug messages. This is used for providing debug information regarding LDAP tasks. The valid value range is 0 to 3. The default is 0.

Parameter	Description
Web: MS LDAP OCS Number attribute name CLI: ldap-ocs-nm-attr **[MSLDAPOCSNumAttributeName]**	Defines the name of the attribute that represents the user OCS number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "msRTCSIP-PrimaryUserAddress".
Web: MS LDAP PBX Number attribute name CLI: ldap-pbx-nm-attr **[MSLDAPPBXNumAttributeName]**	Defines the name of the attribute that represents the user PBX number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "telephoneNumber".
Web: MS LDAP MOBILE Number attribute name CLI: ldap-mobile-nm-attr **[MSLDAPMobileNumAttributeName]**	Defines the name of the attribute that represents the user Mobile number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "mobile".
CLI: ldap-private-nm-attr **[MSLDAPPrivateNumAttributeName]**	Defines the name of the attribute that represents the user's private number in the AD. If this value equals the value of the MSLDAPPrimaryKey or MSLDAPSecondaryKey parameter, then the device queries the AD for the destination number in this private attribute name; otherwise, this parameter is not used as a search key. The default is "msRTCSIP-PrivateLine".
Web: MS LDAP DISPLAY Name Attribute Name CLI: ldap-display-nm-attr **[MSLDAPDisplayNameAttributeName]**	Defines the attribute name that represents the Calling Name in the AD for LDAP queries based on calling number. The valid value is a string of up to 49 characters. The default is "displayName".
CLI: ldap-primary-key **[MSLDAPPrimaryKey]**	Defines the name of the attribute used as a query search key for the destination number in the AD. This is used instead of the "PBX" attribute name (configured by the MSLDAPPBXNumAttributeName parameter). The default is not configured.
CLI: ldap-secondary-key **[MSLDAPSecondaryKey]**	Defines the name of the attribute used as the second query search key for the destination number in the AD, if the primary search key or PBX search is not found.
LDAP Cache Service CLI: cache **[LDAPCacheEnable]**	Enables the LDAP cache service. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For more information on LDAP caching, see "Configuring the Device's LDAP Cache" on page 138.
LDAP Cache Entry Timeout CLI: entry-timeout **[LDAPCacheEntryTimeout]**	Defines the duration (in minutes) that an entry in the LDAP cache is valid. If the timeout expires, the cached entry is only used if there is no connectivity with the LDAP server. The default is 1200.
LDAP Cache Entry Removal Timeout CLI: entry-removal-timemout **[LDAPCacheEntryRemovalTimeout]**	Defines the duration (in hours) after which the LDAP entry is removed from the cache. The default is 0.

41.11.3 Least Cost Routing Parameters

The Least Cost Routing (LCR) parameters are described in the table below.

Table 133: LCR Parameters

Parameter	Description
Web: Routing Rule Groups Table CLI: configure voip > services least-cost-routing routing-rule-groups **[RoutingRuleGroups]**	This table parameter enables the LCR feature and configures the average call duration and default call cost. The default call cost determines whether routing rules that are not configured with a Cost Group are considered as a higher or lower cost route compared to other matching routing rules that are assigned Cost Groups. [RoutingRuleGroups] FORMAT RoutingRuleGroups_Index = RoutingRuleGroups_LCREnable, RoutingRuleGroups_LCRAverageCallLength, RoutingRuleGroups_LCRDefaultCost; [\RoutingRuleGroups] **Note:** For a detailed description of this table, see "Enabling LCR and Configuring Default LCR" on page 144.
Web: Cost Group Table CLI: configure voip > services least-cost-routing cost-group **[CostGroupTable]**	This table parameter configures the Cost Groups for LCR, where each Cost Group is configured with a name, fixed call connection charge, and a call rate (charge per minute). [CostGroupTable] FORMAT CostGroupTable_Index = CostGroupTable_CostGroupName, CostGroupTable_DefaultConnectionCost, CostGroupTable_DefaultMinuteCost; [\CostGroupTable] For example: CostGroupTable 2 = "Local Calls", 2, 1; **Note:** For a detailed description of this table, see "Configuring Cost Groups" on page 146.
Web: Cost Group > Time Band Table CLI: configure voip > services least-cost-routing cost-group-time-bands **[CostGroupTimebands]**	This table parameter configures time bands and associates them with Cost Groups. [CostGroupTimebands] FORMAT CostGroupTimebands_TimebandIndex = CostGroupTimebands_StartTime, CostGroupTimebands_EndTime, CostGroupTimebands_ConnectionCost, CostGroupTimebands_MinuteCost; [\CostGroupTimebands] **Note:** For a detailed description of this table, see "Configuring Time Bands for Cost Groups" on page 146.

41.11.4 Call Setup Rules Parameters

The Call Setup Rules parameters are described in the table below.

Table 134: Call Setup Rules Parameters

Parameter	Description
Web: Call Setup Rules CLI: configure voip/services call-setup-rules **[CallSetupRules]**	This table parameter defines Call Setup Rules that the device runs at call setup for LDAP-based routing and other advanced routing logic requirements including manipulation. [CallSetupRules] FORMAT CallSetupRules_Index = CallSetupRules_RulesSetID, CallSetupRules_AttributesToQuery, CallSetupRules_AttributesToGet, CallSetupRules_RowRole, CallSetupRules_Condition, CallSetupRules_ActionSubject, CallSetupRules_ActionType, CallSetupRules_ActionValue; [\CallSetupRules] **Note:** For a detailed description of this table, see "Configuring Call Setup Rules" on page 148.

42 SBC and DSP Channel Capacity

This chapter lists the supported DSP firmware templates and channel capacity.

Notes:

- Installation and use of voice coders is subject to obtaining the appropriate license and royalty payments.
- The number of channels refers to the maximum channel capacity of the device.
- For additional DSP templates, contact your AudioCodes sales representative.

42.1 Signaling-Media Sessions & User Registrations

The table below lists the maximum capacity. This includes SIP signaling sessions, SBC sessions, and registered users.

Notes:

- The capacity figures listed in the table below are accurate at the time of publication of this document. However, these figures may change due to a later software update. For the latest figures, please contact your AudioCodes sales representative.
- The maximum number of SBC signaling and media sessions are specified in the installed Software License Key, which defines maximum figures for each one separately.
- The maximum number of voice transcoding sessions is specified in the installed Software License Key.
- *Registered Users* indicates the maximum number of users that can be registered with the device (i.e., in the device's registration database). This applies to all the supported applications.
- Regarding signaling, media, and transcoding session resources:
    - √ A signaling session is a SIP dialog session between two SIP entities, traversing the SBC and using one signaling session resource.
    - √ A media session is an audio (RTP or SRTP), fax (T.38), or video session between two SIP entities, traversing the SBC and using one media session resource.
    - √ In case of direct media (i.e., *Anti-tromboning* / *Non-Media Anchoring*), where only SIP signaling traverses the SBC and media flows directly between the SIP entities, only a signaling session resource is used. Thus, if a greater signaling session capacity exists than media, even when media session resources have been exhausted, additional signaling sessions can still be handled for direct-media calls.
    - √ For call sessions requiring transcoding, one transcoding session resource is also used. For example, for a non-direct media call in which one leg uses G.711 and the other leg uses G.729, one signaling resource, one media session resource, and one transcoding session resource is used.

Table 135: Maximum Call Sessions and Registered Users

Model	Signaling Sessions	Media Sessions			Registered Users
		RTP-to-RTP	SRTP-RTP	Codec Transcoding	
Mediant VE SBC	250	250	250	-	2,500
Mediant SE SBC	4000	4000	2000	-	20,000

42.2 Mediant Server Edition

The device does not implement digital signal processing (DSP). Thus, it supports only SBC functionalities that do not require media signal processing.

42.3 Mediant Virtual Edition SBC

The device does not implement digital signal processing (DSP). Thus, it supports only SBC functionalities that do not require media signal processing.

43 Technical Specifications

43.1 Technical Specifications

The device's technical specifications are listed in the table below.

Notes:

- All specifications in this document are subject to change without prior notice.
- The compliance and regulatory information can be downloaded from AudioCodes Web site at http://www.audiocodes.com/library.

Table 136: Technical Specifications

Function	Specification
Networking Interfaces	
LAN	▪ Up to 10 physical Gigabit Ethernet (1000Base-T) port interfaces. ▪ Up to 5 groups of Ethernet port pairs, where each port-pair behaves as active-standby for 1+1 port redundancy. Up to 10 Ethernet port groups if each group is assigned a single port. ▪ Physical port separation by selecting port group per network interface.
High Availability (HA)	
Full HA	Two deployed devices for 1+1 high availability, communicating through a Maintenance network interface. Upon failure of the active device, all functionality is switched over to the redundant device
Media Processing	
IP Transport	VoIP (RTP/RTCP) per IETF RFC 3550 and 3551, IPv6
Control and Management	
Control Protocols	▪ SIP-TCP, UDP, TLS and MSCML ▪ Stand Alone Survivability (SAS) for service continuity
Operations & Management	▪ Embedded HTTP Web Server, Telnet, SNMP V2/V3 ▪ Remote configuration and software download via TFTP, HTTP, HTTPS, DHCP ▪ RADIUS, Syslog (for events, alarms and CDRs)
IP/VoIP Quality of Service	
	▪ IEEE 802.1p, TOS, DiffServ ▪ IEEE 802.1Q VLAN tagging ▪ Shaping, Policing, Queuing, Bandwidth Reservation
Stand Alone Survivability (SAS) Application	
	SAS ensures call continuity between LAN SIP clients upon connectivity failure with IP Centrex services (e.g., WAN IP PBX).
Session Border Controller	
	▪ SIP Header conversion: IP to IP Routing translations of SIP, UDP, TCP, TLS. ▪ Translation of RTP, SRTP; Support SIP trunk with multi-ITSP (Registrations to ITSPs is invoked independently); Topology hiding; Call Admission Control; Call Black/White list. ▪ Intrusion detection/prevention (NIDS); Anti SPIT & SPAM mechanisms.

Function	Specification
Mediant VE SBC - Hardware Requirements	
Hypervisor	▪ VMware ESXi version 5.1 or later ▪ Processor type: 64-bit Intel CPU with support for hardware virtualization (Intel VT-x) enabled ▪ Number of CPU cores: 2 or more ▪ Memory: 3 GB or more ▪ Disk space: 60 GB or more ▪ Network interfaces: 2 or more
Virtual Machine (VM)	▪ Virtual CPU: 2 (recommended to reserve 1 or 2 full physical cores for the virtual machine) ▪ Memory: 2 GB or more ▪ Disk space: 10 GB or more ▪ Virtual Network Interfaces: 2 vNICs are recommended (for trusted / untrusted traffic), an additional vNIC is recommended for HA configurations
Mediant SE SBC - Hardware Requirements	
Server	▪ HP ProLiant DL120 G7 ▪ HP ProLiant DL320e G8
CPU	▪ DL120: Intel Xeon E3-1220 (4 cores, 3.1 GHz, 8M Cache) ▪ DL320e: Intel Xeon E3-1220v2 (4 cores, 3.1 GHz, 8M Cache)
Memory	4 GB
Network Cards	One of the following add-on network cards may be used: ▪ NC112T ▪ NC360T ▪ NC361T ▪ NC364T ▪ NC365T ▪ NC382T ▪ NC331FLR ▪ NC366FLR
Disk	Mechanical hard drive, 72 GB or more, no RAID
Installation From	CD/DVD drive
Installation Interface	VGA Monitor and Keyboard

Reader's Notes

Legal notice:

Alcatel, Lucent, Alcatel-Lucent and the Alcatel-Lucent logo are trademarks of Alcatel-Lucent.
All other trademarks are the property of their respective owners.
The information presented is subject to change without notice.
Alcatel-Lucent assumes no responsibility for inaccuracies contained herein.