

Mediant 9000 SBC

Version 6.8



Table of Contents

1	Overview	17
	Getting Started with Initial Connectivity.....	19
2	Default OAMP IP Address.....	21
3	Changing Default IP Address to Suit your Network Addressing Scheme....	23
	Management Tools	25
4	Introduction	27
5	Web-Based Management.....	29
5.1	Getting Acquainted with the Web Interface	29
5.1.1	Computer Requirements.....	29
5.1.2	Accessing the Web Interface	30
5.1.3	Areas of the GUI	31
5.1.4	Toolbar Description.....	32
5.1.5	Navigation Tree	33
5.1.5.1	Displaying Navigation Tree in Basic and Full View	33
5.1.5.2	Showing / Hiding the Navigation Pane.....	34
5.1.6	Working with Configuration Pages	35
5.1.6.1	Accessing Pages.....	35
5.1.6.2	Viewing Parameters	35
5.1.6.3	Modifying and Saving Parameters	37
5.1.6.4	Working with Tables.....	38
5.1.7	Searching for Configuration Parameters	39
5.1.8	Creating a Login Welcome Message.....	41
5.1.9	Getting Help.....	42
5.1.10	Logging Off the Web Interface.....	43
5.2	Viewing the Home Page.....	44
5.3	Configuring Web User Accounts	46
5.3.1	Basic User Accounts Configuration	47
5.3.2	Advanced User Accounts Configuration.....	49
5.4	Displaying Login Information upon Login	53
5.5	Configuring Web Security Settings	53
5.6	Web Login Authentication using Smart Cards	54
5.7	Configuring Web and Telnet Access List	55
6	CLI-Based Management.....	57
6.1	Getting Familiar with CLI.....	57
6.1.1	Understanding Configuration Modes	57
6.1.2	Using CLI Shortcuts.....	58
6.1.3	Common CLI Commands	59
6.1.4	Configuring Tables in CLI	60
6.1.5	Understanding CLI Error Messages	61
6.2	Enabling CLI.....	62
6.2.1	Enabling Telnet for CLI	62
6.2.2	Enabling SSH with RSA Public Key for CLI.....	62
6.3	Establishing a CLI Session	65
6.4	Configuring Maximum Telnet/SSH Sessions	66

6.5	Viewing Current CLI Sessions	66
6.6	Terminating a User's CLI Session.....	67
6.7	Configuring Displayed Output Lines in CLI Terminal Window	67
7	SNMP-Based Management	69
7.1	Enabling SNMP and Configuring SNMP Community Strings.....	69
7.2	Configuring SNMP Trap Destinations	72
7.3	Configuring SNMP Trusted Managers	73
7.4	Configuring SNMP V3 Users.....	74
8	INI File-Based Management.....	77
8.1	INI File Format	77
8.1.1	Configuring Individual ini File Parameters	77
8.1.2	Configuring Table ini File Parameters	78
8.1.3	General ini File Formatting Rules	79
8.2	Configuring an ini File	80
8.3	Loading an ini File to the Device	81
8.4	Secured Encoded ini File	81
8.5	Configuring Password Display in ini File	82
8.6	INI Viewer and Editor Utility	82
General System Settings		83
9	Configuring SSL/TLS Certificates.....	85
9.1.1	Configuring TLS Certificate Contexts	85
9.1.2	Assigning CSR-based Certificates to TLS Contexts	90
9.1.3	Assigning Externally Created Private Keys to TLS Contexts	91
9.1.4	Generating Private Keys for TLS Contexts.....	92
9.1.5	Creating Self-Signed Certificates for TLS Contexts	93
9.1.6	Importing Certificates and Certificate Chain into Trusted Certificate Store.....	94
9.1.7	Configuring Mutual TLS Authentication	95
9.1.7.1	TLS for SIP Clients.....	95
9.1.7.2	TLS for Remote Device Management.....	96
9.1.8	Configuring TLS Server Certificate Expiry Check	97
10	Date and Time.....	99
10.1	Configuring Date and Time Manually.....	99
10.2	Configuring Automatic Date and Time using SNTP	99
10.3	Configuring Daylight Saving Time.....	101
General VoIP Configuration.....		103
11	Network.....	105
11.1	Configuring Physical Ethernet Ports	105
11.2	Configuring Ethernet Port Groups.....	107
11.3	Configuring Underlying Ethernet Devices	110
11.4	Configuring IP Network Interfaces	112
11.4.1	Assigning NTP Services to Application Types	115
11.4.2	Multiple Interface Table Configuration Summary and Guidelines	115
11.4.3	Networking Configuration Examples	116
11.4.3.1	One VoIP Interface for All Applications	116

11.4.3.2	VoIP Interface per Application Type.....	117
11.4.3.3	VoIP Interfaces for Combined Application Types	117
11.4.3.4	VoIP Interfaces with Multiple Default Gateways	119
11.5	Configuring Static IP Routes	119
11.5.1	Configuration Example of Static IP Routes	122
11.5.2	Troubleshooting the Routing Table	122
11.6	Configuring Quality of Service.....	123
11.7	Configuring ICMP Messages	125
11.8	DNS.....	126
11.8.1	Configuring the Internal DNS Table.....	126
11.8.2	Configuring the Internal SRV Table.....	127
11.9	Network Address Translation Support	129
11.9.1	Device Located behind NAT	129
11.9.1.1	Configuring a Static NAT IP Address for All Interfaces	130
11.9.1.2	Configuring NAT Translation per IP Interface	130
11.9.2	Remote UA behind NAT	132
11.9.2.1	SIP Signaling Messages	132
11.9.2.2	Media (RTP/RTCP/T.38).....	133
11.10	Robust Receipt of Media Streams by Media Latching	135
11.11	Multiple Routers Support.....	137
12	Security	139
12.1	Configuring Firewall Settings	139
12.2	Configuring General Security Settings	143
12.3	Intrusion Detection System	144
12.3.1	Enabling IDS.....	145
12.3.2	Configuring IDS Policies.....	145
12.3.3	Assigning IDS Policies.....	149
12.3.4	Viewing IDS Alarms	150
13	Media	153
13.1	Configuring RTP/RTCP Settings.....	153
13.1.1	Configuring RTP Base UDP Port.....	153
13.2	Configuring Media (SRTP) Security	154
14	Services	157
14.1	DHCP Server Functionality	157
14.1.1	Configuring the DHCP Server	157
14.1.2	Configuring the Vendor Class Identifier.....	161
14.1.3	Configuring Additional DHCP Options.....	162
14.1.4	Configuring Static IP Addresses for DHCP Clients	164
14.1.5	Viewing and Deleting DHCP Clients.....	165
14.2	SIP-based Media Recording	167
14.2.1	Enabling SIP-based Media Recording.....	170
14.2.2	Configuring SIP Recording Routing Rules	171
14.2.3	Configuring SIP User Part for SRS.....	173
14.2.4	Interworking SIP-based Media Recording with Third-Party Vendors	173
14.2.4.1	Genesys	173
14.2.4.2	Avaya UCID.....	173
14.3	RADIUS Authentication.....	175
14.3.1	Setting Up a Third-Party RADIUS Server.....	176
14.3.2	Configuring RADIUS Authentication.....	177
14.3.3	Securing RADIUS Communication.....	178

14.3.4	Authenticating RADIUS in the URL	178
14.4	LDAP-based Management and SIP Services	179
14.4.1	Enabling the LDAP Service	180
14.4.2	Enabling LDAP-based Web/CLI User Login Authentication and Authorization.....	181
14.4.3	Configuring LDAP Servers.....	181
14.4.4	Configuring LDAP DN's (Base Paths) per LDAP Server.....	184
14.4.5	Configuring the LDAP Search Filter Attribute	185
14.4.6	Configuring Access Level per Management Groups Attributes	186
14.4.7	Configuring LDAP Search Methods.....	188
14.4.8	Configuring the Device's LDAP Cache.....	189
14.4.9	Configuring Local Database for Management User Authentication	191
14.4.10	LDAP-based Login Authentication Example.....	193
14.4.11	Active Directory-based Routing for Microsoft Lync	197
14.4.11.1	Querying the AD and Routing Priority	197
14.4.11.2	Configuring AD-Based Routing Rules.....	200
14.5	Least Cost Routing.....	202
14.5.1	Overview	202
14.5.2	Configuring LCR	204
14.5.2.1	Enabling the LCR Feature.....	204
14.5.2.2	Configuring Cost Groups.....	205
14.5.2.3	Configuring Time Bands for Cost Groups	206
14.5.2.4	Assigning Cost Groups to Routing Rules.....	208
14.6	Configuring Call Setup Rules	209
14.6.1	Call Setup Rule Examples	213
15	Quality of Experience.....	215
15.1	Reporting Voice Quality of Experience to SEM.....	215
15.1.1	Configuring the SEM Server	215
15.1.2	Configuring Clock Synchronization between Device and SEM	216
15.1.3	Enabling RTCP XR Reporting to SEM	216
15.2	Configuring Quality of Experience Profiles.....	217
15.3	Configuring Bandwidth Profiles	221
15.4	Configuring Media Enhancement Profiles.....	224
16	Control Network	227
16.1	Configuring Media Realms.....	227
16.2	Configuring Remote Media Subnets	231
16.3	Configuring SRDs	233
16.4	Configuring SIP Interfaces	236
16.5	Configuring IP Groups.....	240
16.6	Configuring Proxy Sets	251
17	SIP Definitions	259
17.1	Configuring SIP Parameters	259
17.2	Configuring Registration Accounts.....	259
17.2.1	Regular Registration Mode.....	261
17.2.2	Single Registration for Multiple Phone Numbers using GIN.....	262
17.3	Configuring Proxy and Registration Parameters.....	263
17.3.1	SIP Message Authentication Example	264
17.4	Configuring SIP Message Manipulation	266
17.5	Configuring SIP Message Policy Rules.....	272

18 Coders and Profiles	275
18.1 Configuring IP Profiles	275
Session Border Controller Application.....	291
19 SBC Overview.....	293
19.1 SIP Network Definitions	294
19.2 SIP Dialog Initiation Process.....	294
19.3 User Registration	297
19.3.1 Initial Registration Request Processing.....	297
19.3.2 SBC Users Registration Database	298
19.3.3 Routing using Users Registration Database.....	298
19.3.4 Registration Refreshes	298
19.3.5 Registration Restriction Control.....	299
19.4 SBC Media Handling.....	300
19.4.1 Media Anchoring without Transcoding (Transparent)	301
19.4.2 No Media Anchoring	302
19.4.3 Restricting Coders	303
19.4.4 Prioritizing Coder List in SDP Offer	304
19.4.5 SRTP-RTP and SRTP-SRTP Transcoding	304
19.4.6 Multiple RTP Media Streams per Call Session	305
19.5 Limiting SBC Call Duration.....	305
19.6 SBC Authentication	305
19.6.1 SIP Authentication Server Functionality	305
19.6.2 User Authentication based on RADIUS.....	306
19.7 Interworking SIP Signaling	307
19.7.1 Interworking SIP 3xx Redirect Responses	307
19.7.1.1 Resultant INVITE Traversing Device	307
19.7.1.2 Local Handling of SIP 3xx	308
19.7.2 Interworking SIP Diversion and History-Info Headers	309
19.7.3 Interworking SIP REFER Messages.....	310
19.7.4 Interworking SIP PRACK Messages	310
19.7.5 Interworking SIP Session Timer	311
19.7.6 Interworking SIP Early Media	311
19.7.7 Interworking SIP re-INVITE Messages.....	313
19.7.8 Interworking SIP UPDATE Messages	314
19.7.9 Interworking SIP re-INVITE to UPDATE.....	314
19.7.10 Interworking Delayed Offer	314
19.7.11 Interworking Call Hold.....	314
19.8 Call Survivability.....	315
19.8.1 Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server for Survivability.....	315
19.8.2 BroadSoft's Shared Phone Line Call Appearance for SBC Survivability.....	316
19.8.3 Call Survivability for Call Centers	317
19.8.4 Survivability Mode Display on Aastra IP Phones	319
19.9 Call Forking	320
19.9.1 Initiating SIP Call Forking	320
19.9.2 SIP Forking Initiated by SIP Proxy Server.....	320
19.9.3 Call Forking-based IP-to-IP Routing Rules.....	321
19.10 Alternative Routing on Detection of Failed SIP Response	321

20 SBC Configuration	323
20.1 Enabling the SBC Application	323
20.2 Configuring General Settings	324
20.2.1 Interworking Dialog Information in SIP NOTIFY Messages	324
20.3 Configuring Admission Control.....	326
20.4 Configuring Allowed Audio Coder Groups	329
20.5 Configuring Allowed Video Coder Groups	331
20.6 Routing SBC	332
20.6.1 Configuring Classification Rules.....	332
20.6.1.1 Classification Based on URI of Selected Header Example.....	338
20.6.2 Configuring Message Condition Rules	339
20.6.3 Configuring SBC IP-to-IP Routing	340
20.6.4 Configuring SIP Response Codes for Alternative Routing Reasons.....	349
20.7 SBC Manipulations.....	351
20.7.1 Configuring IP-to-IP Inbound Manipulations.....	353
20.7.2 Configuring IP-to-IP Outbound Manipulations	356
High Availability System	363
21 HA Overview	365
21.1 Connectivity and Synchronization between Devices.....	365
21.2 Device Switchover upon Failure.....	366
21.3 HA Status on the Home Page	367
22 HA Configuration.....	369
22.1 Initial HA Configuration	369
22.1.1 Network Topology Types and Rx/Tx Ethernet Port Group Settings	369
22.1.2 Configuring the HA Devices	370
22.1.2.1 Step 1: Configure the First Device	371
22.1.2.2 Step 2: Configure the Second Device	373
22.1.2.3 Step 3: Initialize HA on the Devices	373
22.2 Configuration while HA is Operational	374
22.3 Configuring Firewall Allowed Rules.....	375
22.4 Monitoring IP Entity and HA Switchover upon Ping Failure	376
23 HA Maintenance	379
23.1 Maintenance of Redundant Device	379
23.2 Replacing a Failed Device	379
23.3 Forcing a Switchover.....	379
23.4 Software Upgrade	379
23.5 Rescue Options.....	380
23.5.1 Taking a Snapshot.....	380
23.5.2 Viewing Available Snapshots	380
23.5.3 Changing the Default Snapshot.....	380
23.5.4 Deleting a Snapshot	381
23.5.5 Manual Recovery.....	381
23.5.5.1 Returning to the Default Snapshot.....	381
23.5.5.2 Fixing the Current Installation	383
23.5.5.3 Returning to an Arbitrary Snapshot.....	383
23.5.5.4 Returning to a Factory Snapshot	383
23.5.6 Automatic Recovery.....	384

Maintenance	385
24 Basic Maintenance	387
24.1 Resetting the Device	387
24.2 Remotely Resetting Device using SIP NOTIFY	389
24.3 Locking and Unlocking the Device	389
24.4 Saving Configuration	390
25 High Availability Maintenance	391
25.1 Initiating an HA Switchover	391
25.2 Resetting the Redundant Unit	392
26 Disconnecting Active Calls	393
27 Software Upgrade	395
27.1 Loading Auxiliary Files	395
27.1.1 Call Progress Tones File	396
27.1.2 Prerecorded Tones File	399
27.1.3 Dial Plan File	399
27.1.3.1 Creating a Dial Plan File	399
27.1.3.2 Dial Plan Prefix Tags for Routing	400
27.1.3.3 Obtaining IP Destination from Dial Plan File	402
27.1.4 User Information File	402
27.1.4.1 Enabling the User Info Table	402
27.1.4.2 User Information File for SBC User Database	402
27.2 Software License Key	406
27.2.1 Obtaining the Software License Key File	406
27.2.2 Installing the Software License Key	408
27.2.2.1 Installing Software License Key using Web Interface	408
27.2.2.2 Installing Software License Key using CLI	409
27.3 Software Upgrade Wizard	410
27.4 Backing Up and Loading Configuration File	415
28 Automatic Update Mechanism	417
28.1 Automatic Configuration Methods	417
28.1.1 DHCP-based Provisioning	417
28.1.2 HTTP-based Provisioning	418
28.1.3 FTP-based Provisioning	419
28.1.4 Provisioning using AudioCodes EMS	419
28.2 HTTP/S-Based Provisioning using the Automatic Update Feature	420
28.2.1 Files Provisioned by Automatic Update	420
28.2.2 File Location for Automatic Update	420
28.2.3 Triggers for Automatic Update	421
28.2.4 Access Authentication with HTTP Server	422
28.2.5 Querying Provisioning Server for Updated Files	422
28.2.6 File Download Sequence	424
28.2.7 Cyclic Redundancy Check on Downloaded Configuration Files	425
28.2.8 MAC Address Automatically Inserted in Configuration File Name	425
28.2.9 Automatic Update Configuration Examples	426
28.2.9.1 Automatic Update for Single Device	426
28.2.9.2 Automatic Update from Remote Servers	427
28.2.9.3 Automatic Update for Mass Deployment	429

29 Restoring Factory Defaults	431
29.1 Restoring Defaults using CLI	431
29.2 Restoring Defaults using an ini File.....	432
30 Saving Current Configuration to a File and Sending it to Remote Destination	433
Status, Performance Monitoring and Reporting	435
31 System Status	437
31.1 Viewing Device Information.....	437
31.2 Viewing Ethernet Port Information	437
32 Carrier-Grade Alarms.....	439
32.1 Viewing Active Alarms.....	439
32.2 Viewing Alarm History	439
33 Performance Monitoring.....	441
33.1 Viewing MOS per Media Realm	441
33.2 Viewing Quality of Experience	442
33.3 Viewing Average Call Duration	443
34 VoIP Status	445
34.1 Viewing Active IP Interfaces.....	445
34.2 Viewing Ethernet Device Status.....	445
34.3 Viewing Static Routes Status	445
34.4 Viewing Registered Users.....	446
34.5 Viewing Registration Status	447
34.6 Viewing Proxy Set Status.....	448
35 Reporting Information to External Party	451
35.1 Configuring RTCP XR.....	451
35.2 Generating Call Detail Records.....	455
35.2.1 Configuring CDR Reporting	455
35.2.2 CDR Field Description	456
35.2.2.1 CDR Fields for SBC Signaling	456
35.2.2.2 CDR Fields for SBC Media	459
35.3 Configuring RADIUS Accounting	461
Diagnostics	467
36 Syslog and Debug Recordings	469
36.1 Syslog Message Format	469
36.1.1 Event Representation in Syslog Messages.....	470
36.1.2 Identifying AudioCodes Syslog Messages using Facility Levels	472
36.1.3 SNMP Alarms in Syslog Messages	472
36.2 Enabling Syslog	473
36.3 Configuring Web Operations to Report to Syslog	474
36.4 Configuring Debug Recording.....	475
36.5 Filtering Syslog Messages and Debug Recordings	476

36.5.1 Filtering IP Network Traces	478
36.6 Viewing Syslog Messages	479
36.7 Collecting Debug Recording Messages	480
36.8 Debug Capturing on Physical VoIP Interfaces	482
37 Creating Core Dump and Debug Files upon Device Crash	485
38 Testing SIP Signaling Calls	487
38.1 Configuring Test Call Endpoints.....	487
38.2 Starting and Stopping Test Calls.....	491
38.3 Viewing Test Call Statistics	492
38.4 Configuring DTMF Tones for Test Calls.....	494
38.5 Configuring SBC Test Call with External Proxy	494
38.6 Test Call Configuration Examples.....	496
Appendix	499
39 Dialing Plan Notation for Routing and Manipulation.....	501
40 Configuration Parameters Reference	505
40.1 Management Parameters.....	505
40.1.1 General Parameters	505
40.1.2 Web Parameters	505
40.1.3 Telnet Parameters	509
40.1.4 ini File Parameters.....	511
40.1.5 SNMP Parameters.....	511
40.1.6 Serial Parameters	516
40.1.7 Auxiliary and Configuration File Name Parameters	517
40.1.8 Automatic Update Parameters	518
40.2 Networking Parameters.....	523
40.2.1 Ethernet Parameters.....	523
40.2.2 Multiple VoIP Network Interfaces and VLAN Parameters	524
40.2.3 Routing Parameters.....	524
40.2.4 Quality of Service Parameters.....	525
40.2.5 NAT Parameters	526
40.2.6 DNS Parameters.....	527
40.2.7 DHCP Parameters	529
40.2.8 NTP and Daylight Saving Time Parameters.....	531
40.3 Debugging and Diagnostics Parameters.....	533
40.3.1 General Parameters	533
40.3.2 SIP Test Call Parameters	533
40.3.3 Syslog, CDR and Debug Parameters.....	534
40.3.4 Resource Allocation Indication Parameters.....	539
40.4 HA Parameters.....	540
40.5 Security Parameters.....	542
40.5.1 General Security Parameters	542
40.5.2 HTTPS Parameters	544
40.5.3 SRTP Parameters.....	545
40.5.4 TLS Parameters.....	548
40.5.5 SSH Parameters	550
40.5.6 IDS Parameters	551
40.6 Quality of Experience Parameters	553
40.7 Control Network Parameters.....	556

40.7.1	IP Group, Proxy, Registration and Authentication Parameters	556
40.7.2	Network Application Parameters	564
40.8	General SIP Parameters	567
40.9	Coders and Profile Parameters	582
40.10	Channel Parameters	584
40.10.1	RTP, RTCP and T.38 Parameters	584
40.11	SBC Parameters	586
40.12	Services	600
40.12.1	SIP-based Media Recording Parameters	600
40.12.2	RADIUS and LDAP Parameters	601
40.12.2.1	General Parameters	601
40.12.2.2	RADIUS Parameters	601
40.12.2.3	LDAP Parameters	604
40.12.3	Least Cost Routing Parameters	607
40.12.4	Call Setup Rules Parameters	608
41	SBC Capacity	609
42	Technical Specifications	611

Notice

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. Updates to this document can be downloaded from <https://www.audiocodes.com/library/technical-documents>.

This document is subject to change without notice.

Date Published: December-11-2018

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 services are provided by AudioCodes or by an authorized AudioCodes Service Partner. For more information on how to buy technical support for AudioCodes products and for contact information, please visit our Web site at <https://www.audiocodes.com/services-support/maintenance-and-support>.

Stay in the Loop with AudioCodes



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 9000 Hardware Installation Manual
CLI Reference Guide
CLI Wizard User's Guide

Note and Warnings



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



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 product.



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



Notes:

- 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 <https://www.audiocodes.com/services-support/open-source/> 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.

Document Revision Record

LTRT	Description
41630	Initial document release for Version 6.8.
41638	PM_EnableThresholdAlarms parameter added; Web Users table CLI commands updated; Web Users table parameter descriptions of Session Limit and Session Timeout updated; IPv6 Feature Key removed; Internal DNS table supports up to 3 (not 4) IP addresses per host name; remote UA behind NAT for media section updated; MgmntLDAPGroups_Level parameter values updated; CRP Normal mode preconfiguration updated; write-and-backup CLI command updated; Ethernet Group section updated. PhysicalPortsTable_GroupStatus parameter updated; EtherGroupTable_Mode (1RX/1TX removed and default changed); SIPRec note bulletin for France added; IPGroup_SIPConnect parameter added; SBCEnableAASTRASurvivabilityNotice replaced by SBCEnableSurvivabilityNotice; R-factor note added; CDR terminator for RADIUS accounting added; WAN parameters removed; dimensions corrected; descriptions of the following parameters were updated - WebSessionTimeout; SRD_EnableUnAuthenticatedRegistrations; DigitalOOSBehaviorForTrunk; DigitalOOSBehavior; IpProfile_SBCRemoteReplacesBehavior; EnableSBCMediaSync
41642	Updated parameters: Version; Action Type; Port Range Start; Number of Media Session Legs; Port Range End; UDP Port; SIP Group Name; Classify By Proxy Set; Inbound Message Manipulation Set; Outbound Message Manipulation Set; Row Role; Rate; Source IP Address; Group Policy; RTP Base UDP Port New parameters: UII Format; Network Node ID Deleted sub-sections: Configuring NFS Settings; Modified sub-sections: Configuring RTP Base UDP Port; Interworking SIP-based Media Recording with Third-Party Vendors; Configuring IP Groups; Configuring SIP Message Manipulation; Configuring Classification Rules; Configuring SBC IP-to-IP Routing; Automatic Update Mechanism; FTP-based Provisioning; HTTP/S-Based Provisioning using the Automatic Update Feature; HA Parameters Deleted sub-sections: NFS parameters;
41646	- Updated sections: CLI-Based Management (user level access); Understanding Configuration Modes (limitation removed); Configuring TLS Certificate Contexts (IPSec removed); Configuring the SEM Server; Configuring Proxy Sets (failure detection); MAC Address Automatically Inserted in Configuration File Name (note); - New sections: Viewing Proxy Set Status. - Updated parameters: IpProfile_DisconnectOnBrokenConnection; IP2IPRouting_Trigger (option 5); IPOutboundManipulation_PrivacyRestrictionMode; CLIPrivPass; EnableCoreDump; QOEPort (removed); PrackMode (removed). - New parameters: ProxySet_SuccessDetectionRetries; ProxySet_SuccessDetectionInterval; ProxySet_FailureDetectionRetransmissions; EnableNonCallCdr; QOEEnableTLS.

LTRT	Description
41649	- Updated sections: Configuring VoIP LAN Interface for OAMP (CLI); Configuring Web User Accounts (typo); Configuring TLS Certificate Contexts; Assigning CSR-based Certificates to TLS Contexts (SHA); Generating Private Keys for TLS Contexts (4096); Configuring SIP Response Codes for Alternative Routing Reasons; Creating Core Dump and Debug Files upon Device Crash (reset) - New sections: Viewing Proxy Set Status - Updated parameters: NATTranslation_SourceStartPort; NATTranslation_SourceEndPort; NATTranslation_TargetStartPort; NATTranslation_TargetEndPort; IpProfile_SBCUseSilenceSupp; DisableSNMP; EnableCoreDump; SessionExpiresDisconnectTime; TLSContexts_TLSVersion; TLSContexts_ServerCipherString; TLSContexts_ClientCipherString; EnableWebAccessFromAllInterfaces; SSHMaxLoginAttempts; ResetWebPassword - New parameters: TLSContexts_DTLSVersion; TLSContexts_DHKeySize
41983	- Updated sections: Configuring Firewall Settings - Updated parameters: AccessList_Source_IP; AccessList_Source_Port; AccessList_Start_Port; AccessList_End_Port; SRD_IntraSRDMediaAnchoring; ProxySet_EnableProxyKeepAlive; ProxySet_IsProxyHotSwap; IpProfile_SBCPlayHeldTone; KeepAliveTrapPort; EnablePChargingVector (removed) - New parameters: CustomerSN
41987	- Updated sections: Configuring NAT Translation per IP Interface; Silence Suppression (removed); SIP-based Media Recording (France URL); DHCP-based Provisioning (note); Viewing Active Alarms (note) - Updated parameters: IpProfile_SCE (removed); IpProfile_SBCSDPPTIMEAnswer (Preferred Value); IpProfile_SBCPreferredPTIME; SyslogOptimization (default); IsCiscoSCEMode; EnableSilenceCompression (removed); UseDisplayNameAsSourceNumber - New parameters: IpProfile_SBCAdaptRFC2833BWToVoiceCoderBW; ActiveAlarmTableMaxSize; NoAlarmForDisabledPort; SBCRemoveSIPSPFromNonSecuredTransport
41996	- New parameters: TimeZoneFormat; SRPTunnelingValidateRTPRxAuthentication; SRPTunnelingValidateRTCPRxAuthentication - Deleted parameters: EnableSIPRemoteReset; IsCiscoSCEMode - Modified parameters: InterfaceTable_PrefixLength; StaticRouteTable_PrefixLength; SBCAdmissionControl_RequestType (default); SetupTime; ConnectTime; ReleaseTime; Test_Call_Play (note); TelnetServerEnable; DisableSNMP - Updated sections: Multiple Interface Table Configuration Summary and Guidelines (64); Configuring Media (SRTP) Security; Configuring SIP Message Manipulation (max); Configuring CDR Reporting (note)

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 <https://online.audiocodes.com/documentation-feedback>.

1 Overview

AudioCodes Mediant 9000 Session Border Controller is a highly scalable Session Border Controller (SBC) designed for deployment in large enterprise and contact center locations and as an access SBC for service provider environments. The Mediant 9000 is a high-capacity SBC, supporting thousands of concurrent sessions and extensive SIP connectivity with wide-ranging interoperability, enhanced perimeter defense against cyber-attacks, and advanced voice quality monitoring.

The device also supports active/standby (1+1) redundancy (High Availability) by employing two devices in the network. The device offers branch survivability during WAN failure, ensuring call service continuity.

The device supports a wide range of local and remote management tools such as an HTTP/S-based Web server and command-line interface (CLI).



Note: For maximum call capacity figures, see "SBC Capacity" on page [609](#).

This page is intentionally left blank.

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 2-1: 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
Interface Name	"O+M+C"

This page is intentionally left blank.

3 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.

The procedure below describes how to change the default IP address using the CLI. The procedure uses the regular CLI commands. Alternatively, you can use the CLI Wizard utility to set up your device with the initial OAMP settings. The utility provides a fast-and-easy method for initial configuration of the device through CLI. For more information, refer to the *CLI Wizard User's Guide*.

➤ **To change the IP address using CLI:**

1. Establish a CLI session with the device, by one of the following connection methods:
 - RS-232 port: Connect the serial port to a PC COM port
 - VGA and USB ports: Connect the VGA port to a VGA monitor and the USB port to a keyboard

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 9000>
```

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. At the prompt, type the following command to complete the network interface configuration:

```
(network-if-0)# activate
(network-if-0)# exit
```

9. If the device is connected to an IP network that uses a VLAN ID, type the following commands to configure it (otherwise, skip this step):

```
(config-voip)# interface network-dev 0
(network-dev-0)# vlan-id 10
(network-dev-0)# exit
(network-dev-0)# activate
```

10. At the prompt, type the following command to complete configuration:

```
(config-voip)# exit
```

11. 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 Ethernet Port Groups" on page 107.

12. At the prompt, type the following to reset the device and activate the new configuration:

```
# reload now
```


Part II

Management Tools

4 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 [29](#)
- Command Line Interface (CLI) - see "CLI-Based Management" on page [57](#)
- Simple Network Management Protocol (SNMP) - see "SNMP-Based Management" on page [69](#)
- Configuration *ini* file - see "INI File-Based Management" on page [77](#)

**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 [505](#).

This page is intentionally left blank.

5 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 406).

5.1 Getting Acquainted with the Web Interface

This section provides a description of the Web interface.

5.1.1 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.

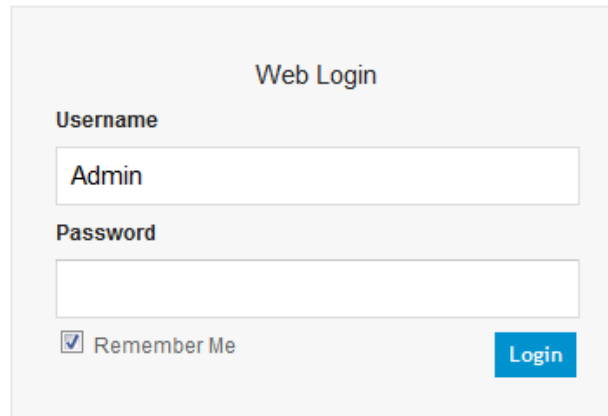
5.1.2 Accessing the Web Interface

The following procedure describes how to access the Web interface.

➤ **To access the Web interface:**

1. Open a standard Web browser (see "Computer Requirements" on page 29).
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 5-1: Web Login Screen



The image shows a web login interface titled "Web Login". It contains two input fields: "Username" with the text "Admin" entered, and "Password" which is empty. Below the password field is a checkbox labeled "Remember Me" which is checked. To the right of the checkbox is a blue button labeled "Login".

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 44.

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 46.
- 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
```



5.1.3 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 5-2: Main Areas of the Web Interface GUI

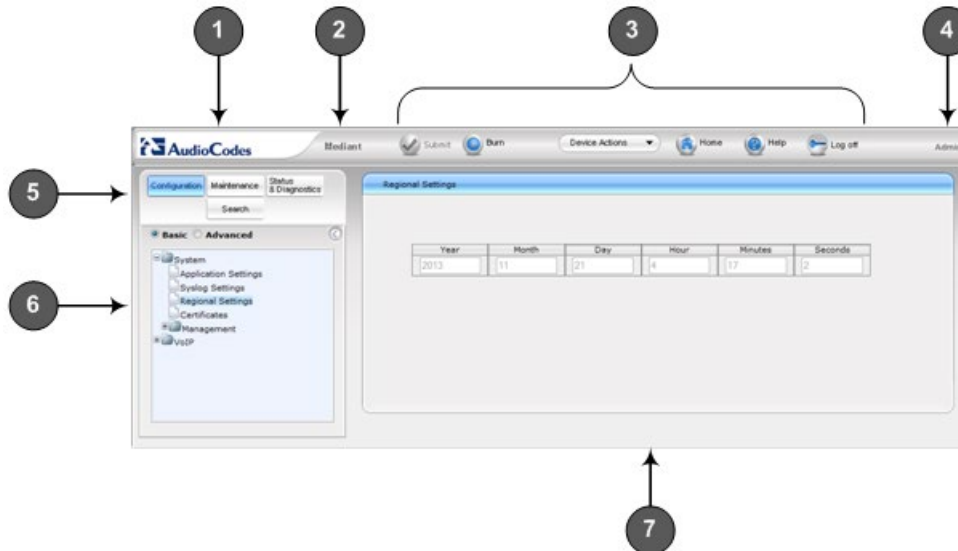


Table 5-1: Description of the Web GUI Areas

Item #	Description
1	AudioCodes company logo.
2	Product name.
3	Toolbar, providing frequently required command buttons. For more information, see "Toolbar Description" on page 32.
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 35) ▪ Search tab: Enables a search engine for searching configuration parameters (see "Searching for Configuration Parameters" on page 39)
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 33.
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 35.

5.1.4 Toolbar Description

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

Table 5-2: Description of Toolbar Buttons

Icon	Button Name	Description
	Submit	Applies parameter settings to the device (see "Saving Configuration" on page 390). Note: This icon is grayed out when not applicable to the currently opened page.
	Burn	Saves parameter settings to flash memory (see "Saving Configuration" on page 390).
	Device Actions	Opens a drop-down list with frequently needed commands: ▪ Load Configuration File: Opens the Configuration File page for loading an <i>ini</i> file to the device (see "Backing Up and Loading Configuration File" on page 415). ▪ Save Configuration File: Opens the Configuration File page for saving the <i>ini</i> file to a folder on your PC (see "Backing Up and Loading Configuration File" on page 415). ▪ Reset: Opens the Maintenance Actions page for performing various maintenance procedures such as resetting the device (see "Resetting the Device" on page 387). ▪ Software Upgrade Wizard: Starts the Software Upgrade Wizard for upgrading the device's software (see "Software Upgrade Wizard" on page 410). ▪ Switch Over: Opens the High Availability Maintenance page for switching between Active and Redundant devices (see High Availability Maintenance on page 391). ▪ Reset Redundant: Opens the High Availability Maintenance page for resetting the Redundant device (see High Availability Maintenance on page 391).
	Home	Opens the Home page (see "Viewing the Home Page" on page 44).
	Help	Opens the Online Help topic of the currently opened configuration page (see "Getting Help" on page 42).
	Log off	Logs off a session with the Web interface (see "Logging Off the Web Interface" on page 43).
-	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.

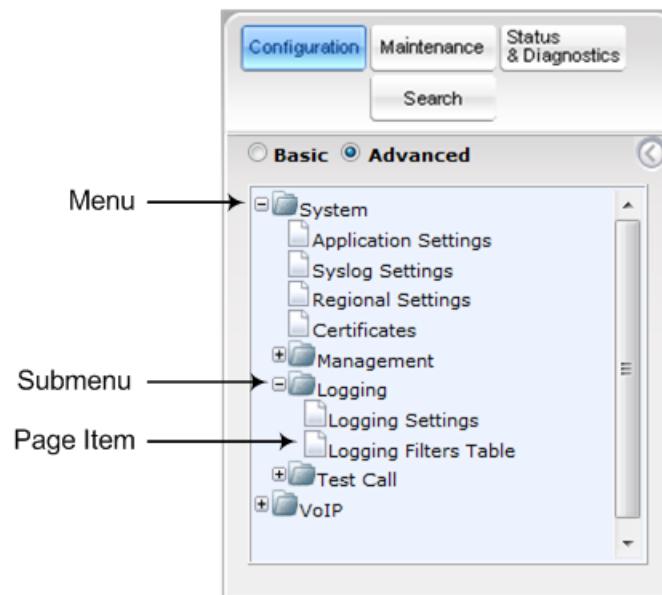
5.1.5 Navigation Tree

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 5-3: 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.

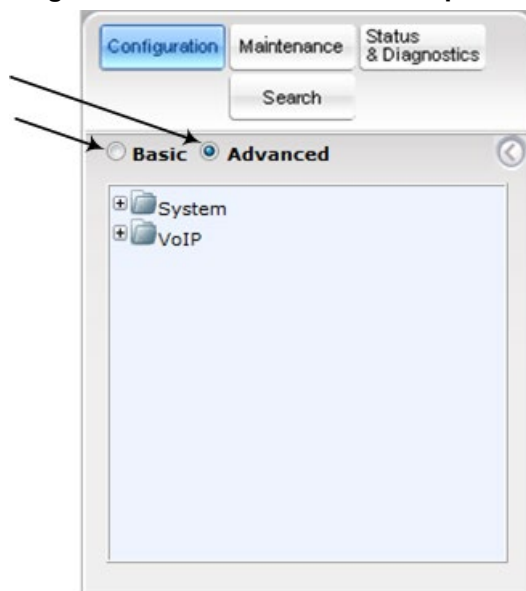
5.1.5.1 Displaying Navigation Tree in Basic and Full View

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 5-4: Basic and Full View Options



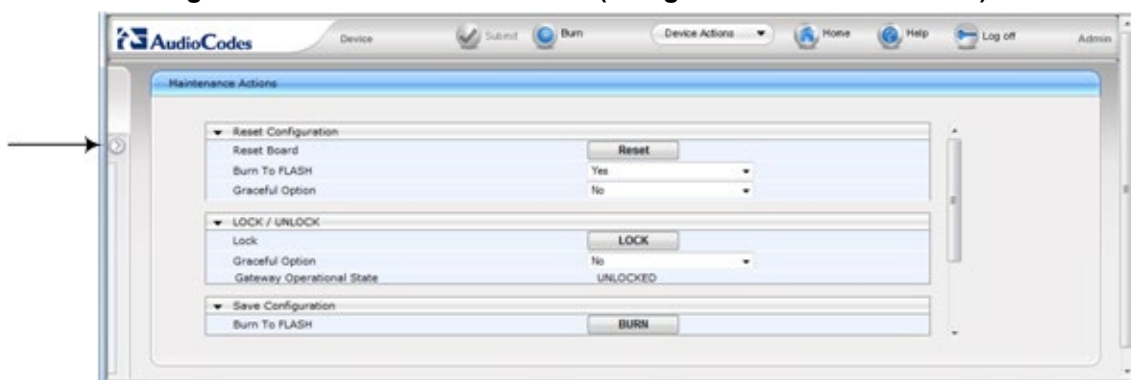
Note: After you reset the device, the Web GUI is displayed in **Basic** view.

5.1.5.2 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 5-5: Show and Hide Button (Navigation Pane in Hide View)



5.1.6 Working with Configuration Pages

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

5.1.6.1 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 46). If a page is read-only, "Read-Only Mode" is displayed at the bottom of the page.

5.1.6.2 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 35
- Displaying parameter groups - see "Showing / Hiding Parameter Groups" on page 36

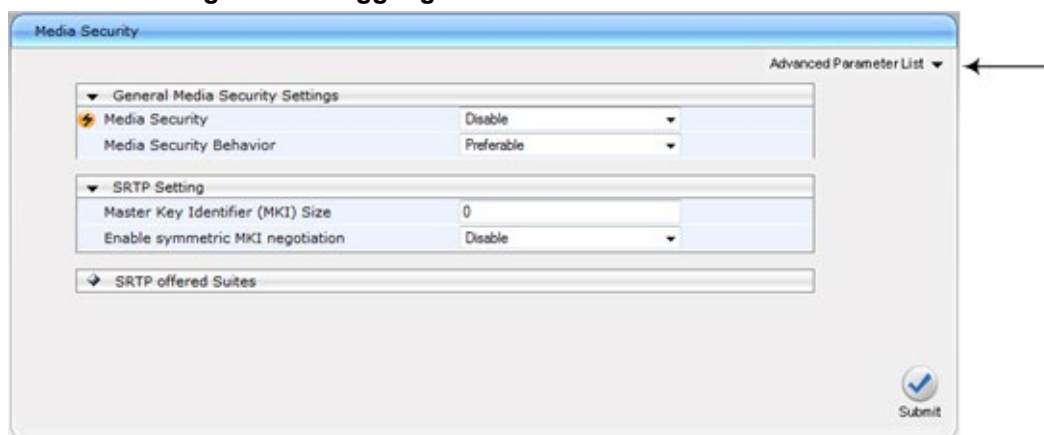
5.1.6.2.1 Displaying Basic and Advanced Parameters

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 5-6: Toggling between Basic and Advanced View



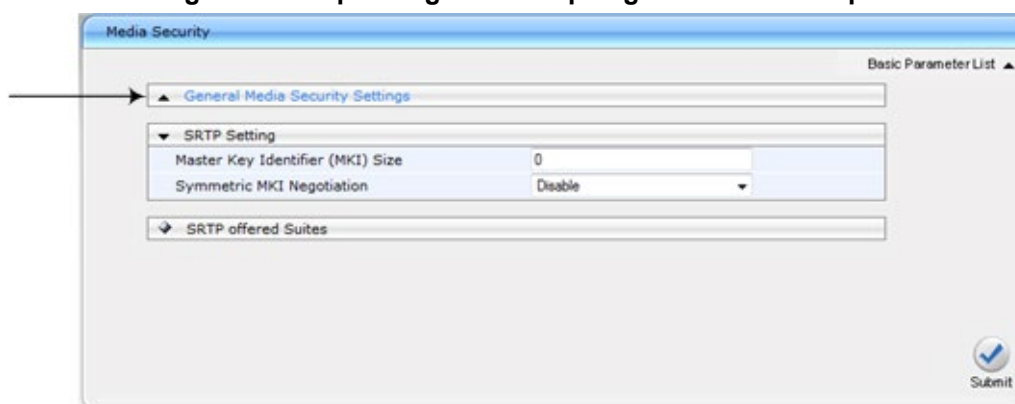
Notes:

- When the Navigation tree is in **Advanced** display mode (see "Navigation Tree" on page 33), 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.

5.1.6.2.2 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.

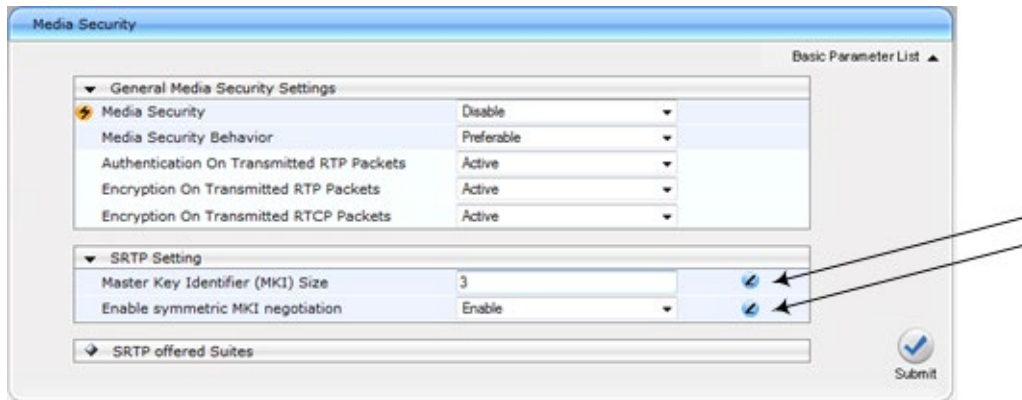
Figure 5-7: Expanding and Collapsing Parameter Groups



5.1.6.3 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 5-8: Edit Symbol after Modifying Parameter Value



- 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.

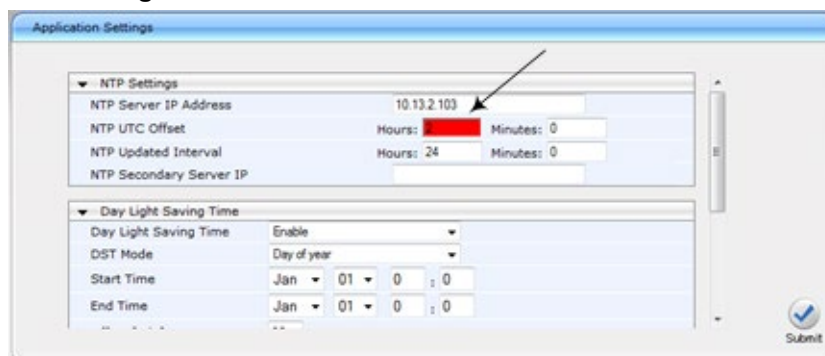
When you click **Submit**, modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect. Parameters displayed on the page with the lightning  icon take effect only after a device reset. For resetting the device, see "Resetting the Device" on page 387.



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 390).

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 5-9: Value Reverts to Previous Valid Value



5.1.6.4 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 5-10: Displayed Details Pane

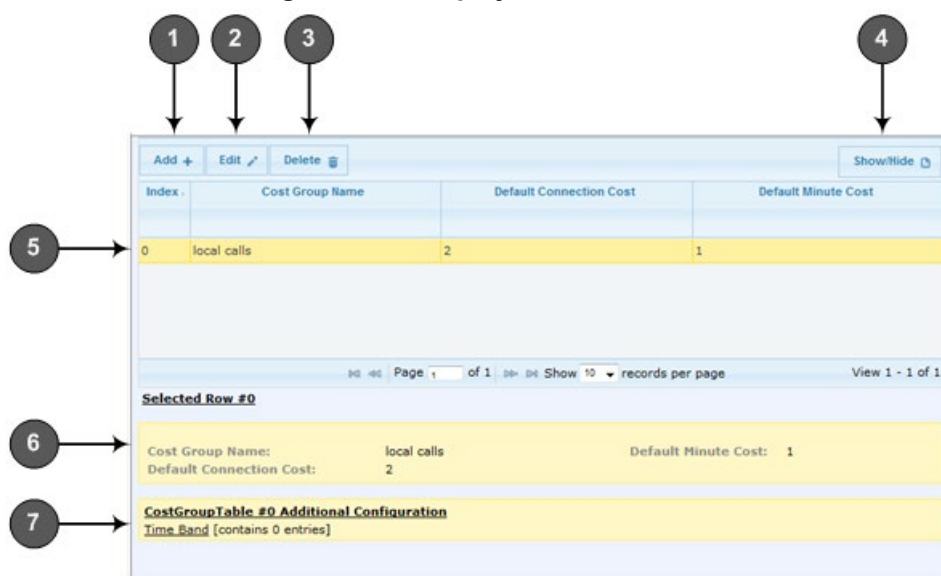


Table 5-3: Enhanced Table Design Description

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.

Item #	Button	
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.
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.

Some tables also provide the **Up** and **Down** buttons for changing the position (index number) of a selected table row. These buttons become available only if the table contains more than one row.

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 5-11: Viewing Table Rows per Page

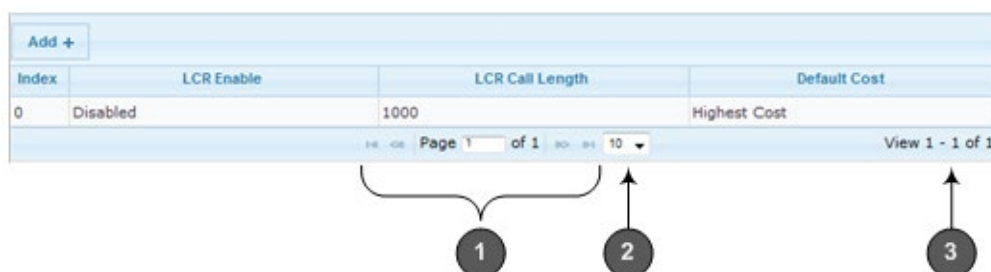


Table 5-4: Row Display and Page Navigation

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.

5.1.7 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 5-12: Searched Result Screen

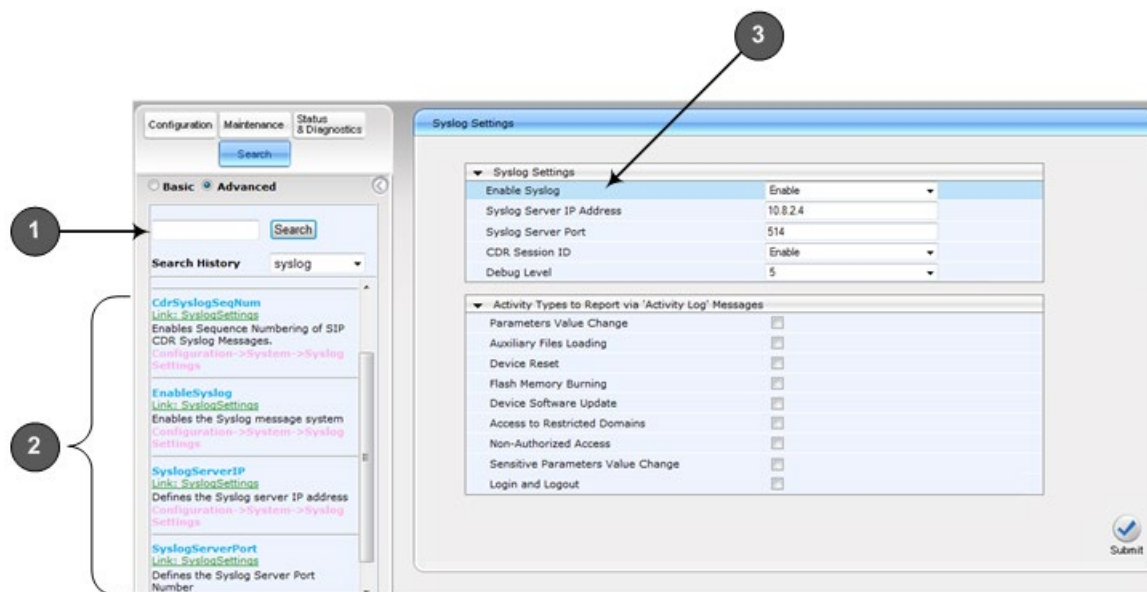


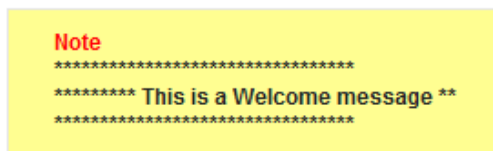
Table 5-5: Search Description

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

5.1.8 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 5-13: User-Defined Web Welcome Message after Login



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 5-6: 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: <pre>[WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; [WelcomeMessage]</pre> For Example: <pre>[WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text; WelcomeMessage 1 = "*****", WelcomeMessage 2 = "***** This is a Welcome message **"; WelcomeMessage 3 = "*****", [WelcomeMessage]</pre> Each index row represents a line of text in the Welcome message box. Up to 20 lines (or rows) of text can be defined.

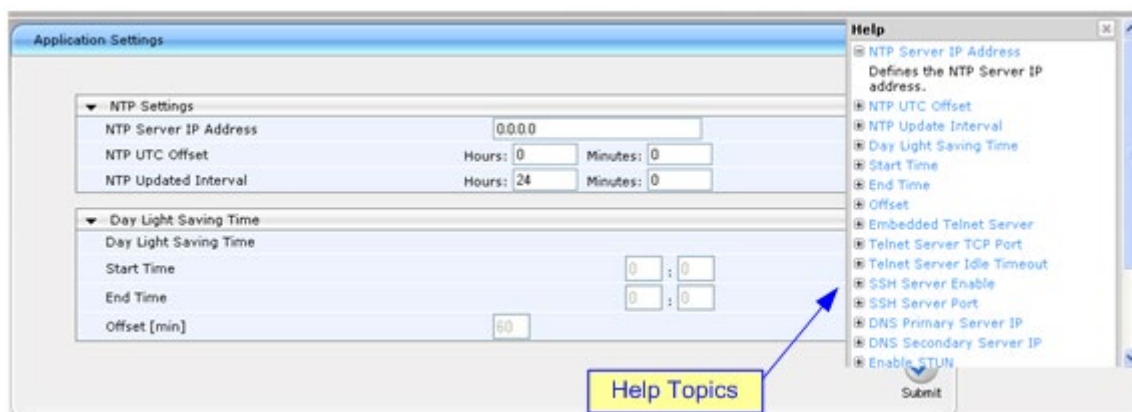
5.1.9 Getting Help

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 5-14: Help Topic for Current Page



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.

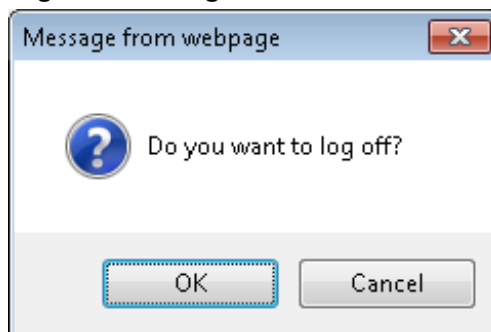
5.1.10 Logging Off the Web Interface

The following procedure 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 5-15: Log Off Confirmation Box



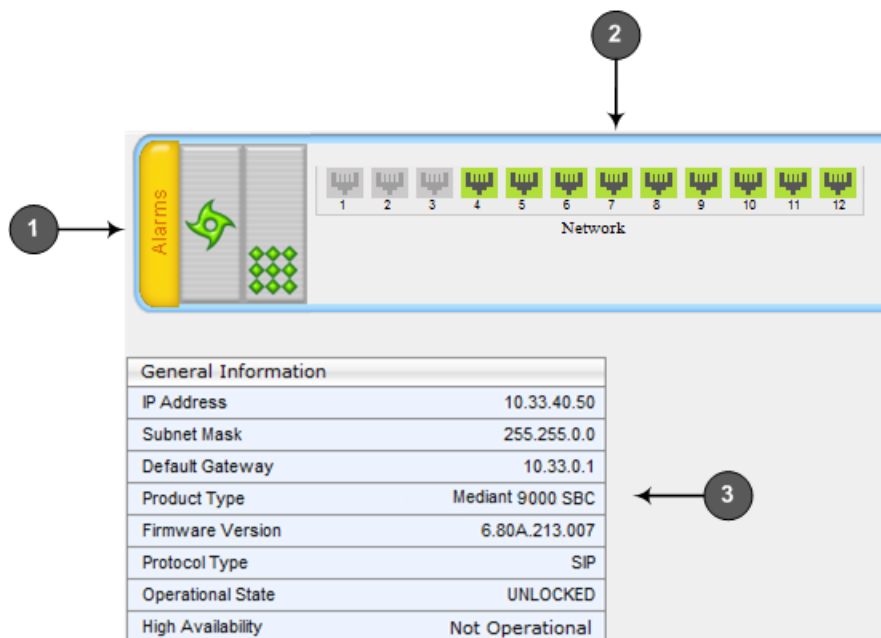
2. Click **OK**; you are logged off the Web session and the Web Login dialog box appears enabling you to re-login, if required.

5.2 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.



The table below describes the areas of the Home page.

Table 5-7: 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 - Yellow = Minor alarm To view active alarms, click this Alarms area to open the Active Alarms page (see Viewing Active Alarms on page 439).
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 437).

Item #	Description
3	General Information pane, displaying the following: ▪ 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 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. For more information, see 'HA Status on the Home Page' on page 367.

5.3 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 5-8: 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 exist.
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.
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 5-9: 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 505).
- You can define additional Web user accounts using a RADIUS server (see "RADIUS Authentication" on page 56).

5.3.1 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 5-10: 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 5-16: 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 Administrator ▼

▼ 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.

5.3.2 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 53).
- 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 53). 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 following procedure describes how to configure Web users in the Web interface. You can also configure this using the CLI command `configure system > create-users-table`.

➤ **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 5-17: 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

Page 1 of 1 View 1 - 2 of 2

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

Figure 5-18: 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

Submit

Cancel

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 5-11: 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	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	Defines the Web user's password.

Parameter	Description
CLI: password	The valid value is a string of 8 to 40 ASCII characters, which must adhere to the following guidelines: ▪ Include at least eight characters. ▪ Include at least two letters that are upper case (e.g., A). ▪ Include at least two letters that are lower case (e.g., a). ▪ Include at least two numbers (e.g., 4). ▪ Include at least two symbols (non-alphanumeric characters) (e.g., \$, #, %). ▪ Must contain no spaces. ▪ Include at least four new characters that were not used in the previous password.
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 53). 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 53). 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: password-age	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 concurrent Web interface sessions allowed for the specific user. For example, if configured to 2, the same user account can be logged into the device's Web interface (i.e., same username-password combination) from two different management stations (i.e., IP addresses) at any one time. Once the user logs in, the session is active until the user logs off (by clicking the Log off icon on the toolbar) or until the session expires if the user is inactive for a user-defined duration (see the 'Session Timeout' parameter below). The valid value is 0 to 5. The default is 2. Note: Up to 5 users can be concurrently logged in to the Web interface.

Parameter	Description
Web: Session Timeout CLI: session-timeout	Defines the duration (in minutes) of inactivity of a logged-in user in the Web interface, after which the user is automatically logged off the Web session. In other words, the session expires when the user has not performed any operations (activities) in the Web interface for the configured timeout duration. The valid value is 0 to 100000. A value of 0 means no timeout. The default value is according to the settings of the WebSessionTimeout global parameter (see "Configuring Web Security Settings" on page 53).
Web: Block Duration CLI: block-duration	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 53). 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 53). Note: The 'Deny Authentication Timer' parameter relates to failed Web logins from specific IP addresses.
Web: User Level CLI: privilege	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.

5.4 Displaying Login Information upon Login

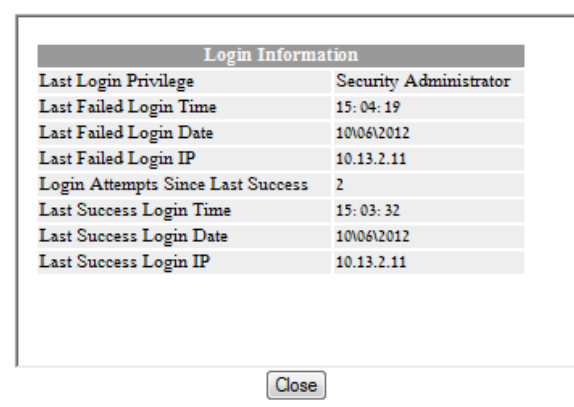
The device can display login information immediately upon Web login.

➤ **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 5-19: Login Information Window



The screenshot shows a window titled "Login Information" containing a table with the following data:

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

At the bottom of the window is a "Close" button.

5.5 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 505.

➤ **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 390.

5.6 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.

5.7 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 505).

➤ **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 5-20: 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 5-21: 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 390.



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.

6 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.
- The CLI can only be accessed by management users with the following user levels:
 - ✓ Administrator
 - ✓ Security Administrator
 - ✓ Master
- For a description of the CLI commands, refer to the CLI Reference Guide.

6.1 Getting Familiar with CLI

This section describes the basic structure of the device's CLI, which you may need to know before configuring the device through CLI.

6.1.1 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 "#".



Note: The default password for accessing the Enable mode is "Admin" (case-sensitive). To change this password, use the CLIPrivPass ini file parameter.

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) #
```

6.1.2 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 6-1: 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.
? (question mark)	- Displays a list of all subcommands in the current mode, for example: <pre>(config-voip) # voip-network ? dns Enter voip-network dns ip-group IP Group table nat-translation NATTranslationtable ...</pre> - Displays a list of available commands beginning with certain letter(s), for example: <pre>(config) # voip-network d? dns Enter voip-network dns</pre> - 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: <pre>(config) # voip-network dns srv2ip ? [0-9] index</pre> 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: <pre>(config) # logging host 10.1.1.1 ? <cr></pre>
<Ctrl + A>	Moves the cursor to the beginning of the command line.
<Ctrl + E>	Moves the cursor to the end of the command line.

Shortcut Key	Description
<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 67.

6.1.3 Common CLI Commands

The following table contains descriptions of common CLI commands.

Table 6-2: 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: <pre>(config)# interface vlan 1 (conf-if-VLAN 1)# do show interfaces GigabitEthernet 0/0</pre>
no	Undoes an issued command or disables a feature. Enter no before the command: <pre># no debug log</pre>
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 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: <pre>(config)# exit # exit (session closed)</pre>
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.

Command	Description
<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 <i>grep</i> common Unix utility options ▪ egrep <options> - filter lines according to <i>egrep</i> 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: <pre># show system version grep Number ;Serial Number: 2239835;Slot Number: 1</pre>

6.1.4 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).

6.1.5 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.

6.2 Enabling CLI

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

6.2.1 Enabling Telnet for CLI

The following procedure 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 41).

➤ 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 509.
4. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

6.2.2 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 6-1: Selecting Public RSA Key in PuTTY



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	AAAAB3NzaC1yc2EAAAABJQAAAIBs4d5kzHckyRml3awmb2b/Csed/d2FDolyGw/bp0YsLl8B9hCtu/GE5AQWbU6FPxmM7ehhUcMptE2NzaZnz7S2/mE3xEwchM51nre n7ZJ3kUcLh2GdoT abG2UpF82ag+y0pnu4wJukD6Kxm1ALMdMQNEyH5k6cw==
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. Configure the other SSH parameters as required. For a description of these parameters, see "SSH Parameters" on page 550.
- e. 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.

6.3 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 following procedure describes how to establish a CLI session with the device.

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



Note: 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 46.

➤ **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

6.4 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**.

6.5 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.

6.6 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 66).



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

6.7 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.

This page is intentionally left blank.

7 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 69.
- 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 406.
- For more information on using the EMS tool, refer to the *EMS User's Manual* and *EMS Server IOM Manual*.

7.1 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 511.

➤ 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

Trap Community String

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 7-1: SNMP Community String Parameter Descriptions

Parameter	Description
Community String - Read Only configure system > snmp > ro-community-string [SNMPReadOnlyCommunityString_x]	Defines a read-only SNMP community string. Up to five read-only community strings can be configured. The valid value is a string of up to 19 characters that can include only the following: ▪ Upper- and lower-case letters (a to z, and A to Z) ▪ Numbers (0 to 9) ▪ Hyphen (-) ▪ Underline (_) For example, "Public-comm_string1". The default is "public".

Parameter	Description
Community String - Read / Write configure system > snmp > rw- community-string [SNMPReadWriteCommunityString_x]	Defines a read-write SNMP community string. Up to five read-write community strings can be configured. The valid value is a string of up to 19 characters that can include only the following: ▪ Upper- and lower-case letters (a to z, and A to Z) ▪ Numbers (0 to 9) ▪ Hyphen (-) ▪ Underline (_) For example, "Private-comm_string1". The default is "private".
Trap Community String configure system > snmp trap > community-string [SNMPTrapCommunityString]	Defines the community string for SNMP traps. The valid value is a string of up to 19 characters that can include only the following: ▪ Upper- and lower-case letters (a to z, and A to Z) ▪ Numbers (0 to 9) ▪ Hyphen (-) ▪ Underline (_) For example, "Trap-comm_string1". The default is "trapuser".

7.2 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 7-1: 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 7-2: 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.

Parameter	Description
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 74
Trap Enable [SNMPManagerTrapSendingEnable_x]	Activates the sending of traps to the SNMP Manager. [0] Disable [1] Enable (Default)

7.3 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 following procedure 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 7-2: 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.

7.4 Configuring SNMP V3 Users

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

The following procedure 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 7-3: SNMP V3 Setting Page - Add Record Dialog Box

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 72), the configured trap destination becomes disabled and the trap user reverts to default (i.e., SNMPv2).

Table 7-3: 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	Privacy protocol of the SNMP v3 user.

Parameter	Description
[SNMPUsers_PrivProtocol]	▪ [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.

This page is intentionally left blank.

8 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 [505](#).
- To restore the device to default settings using the *ini* file, see "Restoring Factory Defaults" on page [431](#).

8.1 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 [77](#)
- Table parameters - see "Configuring Table ini File Parameters" on page [78](#)

8.1.1 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 [79](#).

8.1.2 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 79.

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.

8.1.3 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.

8.2 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 390).
 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 81.



Note: To restore the device to default settings using the *ini* file, see "Restoring Factory Defaults" on page 431.

8.3 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 395): 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 415): 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*.

8.4 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).

8.5 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>`).

8.6 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 this <https://www.audiocodes.com/library/firmware/>, 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

9 Configuring SSL/TLS Certificates

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



Notes:

- The device is shipped with an active, default TLS setup. Thus, configure certificates only if required.
- Since X.509 certificates have an expiration date and time, you must configure the device to use Network Time Protocol (NTP) to obtain the current date and time from an NTP server. Without the correct date and time, client certificates cannot work. For configuring NTP, see [Configuring Automatic Date and Time using SNTP](#) on page 99.
- Only **Base64 (PEM)** encoded X.509 certificates can be loaded to the device.

9.1.1 Configuring TLS Certificate Contexts

The TLS Contexts table lets you configure up to 100 TLS certificates, referred to as *TLS Contexts*. The Transport Layer Security (TLS), also known as Secure Socket Layer (SSL), is used to secure the device's SIP signaling connections, Web interface, and Telnet server. The TLS/SSL protocol provides confidentiality, integrity, and authenticity between two communicating applications over TCP/IP.

The device is shipped with a default TLS Context (ID 0 and string name "default"), which includes a self-generated random private key and a self-signed server certificate. The subject name for the default certificate is "ACL_nnnnnnn", where *nnnnnnn* denotes the serial number of the device. The default TLS Context can be used for SIP over TLS (SIPS) or any other supported application such as Web (HTTPS), Telnet, and SSH. The default TLS Context cannot be deleted.

The user-defined TLS Contexts are used **only** for SIP over TLS (SIPS). This enables you to use different TLS certificates for your IP Groups (SIP entities). This is done by assigning a specific TLS Context to the Proxy Set and/or SIP Interface associated with the IP Group.

Each TLS Context can be configured with the following:

- Context ID and name
- TLS version - SSL 2.0 (only for TLS handshake), SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2)
- Encryption ciphers for server and client - DES, RC4 compatible, Advanced Encryption Standard (AES)
- Online Certificate Status Protocol (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 OCSP. When OCSP is enabled, the device queries the OCSP server for revocation information whenever a peer certificate is received (TLS client mode, or TLS server mode with mutual authentication).
- Private key - externally created and then uploaded to device
- X.509 certificates - self-signed certificates or signed as a result of a certificate signing request (CSR)
- Trusted root certificate authority (CA) store (for validating certificates)

When the device establishes a TLS connection (handshake) with a SIP user agent (UA), the TLS Context is determined as follows:

■ **Incoming calls:**

1. Proxy Set: If the incoming call is successfully classified to an IP Group based on Proxy Set (i.e., IP address of calling party) and the Proxy Set is configured for TLS ('Transport Type' parameter is set to **TLS**), the TLS Context assigned to the Proxy Set is used. For configuring Proxy Sets, see Configuring Proxy Sets on page 251.
2. SIP Interface: If the Proxy Set is either not configured for TLS (i.e., the 'Transport Type' parameter is set to **UDP**) or not assigned a TLS Context, and/or classification to a Proxy Set fails, the device uses the TLS Context assigned to the SIP Interface used for the call. For configuring SIP Interfaces, see Configuring SIP Interfaces on page 236.
3. Default TLS Context (ID 0): If the SIP Interface is not assigned a TLS Context or no SIP Interface is used for the call, the device uses the default TLS Context.

■ **Outgoing calls:**

1. Proxy Set: If the outgoing call is sent to an IP Group associated with a Proxy Set that is assigned a TLS Context and the Proxy Set is configured for TLS (i.e., 'Transport Type' parameter is set to **TLS**), the TLS Context is used. If the 'Transport Type' parameter is set to **UDP**, the device uses UDP to communicate with the proxy and no TLS Context is used.
2. SIP Interface: If the Proxy Set is not assigned a TLS Context, the device uses the TLS Context assigned to the SIP Interface used for the call.
3. Default TLS Context (ID 0): If the SIP Interface is not assigned a TLS Context or no SIP Interface is used for the call, the device uses the default TLS Context.



Notes:

- If the TLS Context used for an existing TLS connection is changed during the call by the user agent, the device ends the connection.
- The device does not query OCSP for its own certificate.
- Some PKIs do not support OCSP, but generate Certificate Revocation Lists (CRLs). For such scenarios, set up an OCSP server such as OCSPD.

TLS Context certification also enables employing different levels of security strength (key size) per certificate. This feature also enables the display of the list of all trusted certificates currently installed on the device. For each certificate, detailed information such as issuer and expiration date is shown. Certificates can be deleted or added from/to the Trusted Root Certificate Store.

You can also configure TLS certificate expiry check, whereby the device periodically checks the validation date of the installed TLS server certificates and sends an SNMP trap event if a certificate is nearing expiry. This feature is configured globally for all TLS Contexts. For configuring TLS certificate expiry check, see 'Configuring TLS Server Certificate Expiry Check' on page 97.

The following procedure describes how to configure a TLS Context in the Web interface. You can also configure this using the table ini file parameter, TLSContexts or CLI command, configure system > tls <ID>.

➤ **To configure a TLS Context:**

1. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
2. Click **Add**; the following dialog box appears:

Figure 9-1: TLS Contexts Table - Add Record Dialog Box

Add Record	
Index	1
Name	
Version	0
Ciphers Server	AES:RC4
Ciphers Client	ALL:!ADH
Ocsf Server	0
Ocsf Server Primary	0.0.0.0
Ocsf Server Secondary	0.0.0.0
Ocsf Port	2560
Ocsf Default Response	0
Submit Cancel	

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

TLS Context Parameter Descriptions

Parameter	Description
Web: Index CLI: tls <ID> [TLSContexts_Index]	Defines an index number for the new table record. Note: Each table row must be configured with a unique index.
Web: Name CLI: name [TLSContexts_Name]	Defines an arbitrary name to easily identify the TLS Context. The valid value is a string of up to 31 characters.

Parameter	Description
Web: TLS Version CLI: tls-version [TLSContexts_TLSEVersion]	Defines the supported SSL/TLS protocol version. Clients attempting to communicate with the device using a TLS version that is not configured are rejected. [0] Any including SSLv3 = (Default) SSL 3.0 and all TLS versions are supported. SSL/TLS handshakes always start with an SSL 2.0-compatible handshake and then switch to the highest TLS version supported by both peers. [1] TLSv1.0 = TLS 1.0 only. [2] TLSv1.1 = TLS 1.1 only. [3] TLSv1.0 and TLSv1.1 = TLS 1.1 and TLS 1.0 only. [4] TLSv1.2 = TLS 1.2 only. [5] TLSv1.0 and TLSv1.2 = TLS 1.2 and TLS 1.0 only. [6] TLSv1.1 and TLSv1.2 = TLS 1.2 and TLS 1.1 only [7] TLSv1.0 TLSv1.1 and TLSv1.2 = TLS 1.2, TLS 1.1 and TLS 1.0 only (excludes SSL 3.0).
DTLS Version [TLSContexts_DTLSEVersion]	Defines the Datagram Transport Layer Security (DTLS) version, which is used to negotiate keys for WebRTC calls. [0] Any (default) [1] DTLSv1.0 [2] DTLSv1.2 Note: The parameter is applicable only to the SBC application.
Web: Cipher Server CLI: ciphers-server [TLSContexts_ServerCipherString]	Defines the supported cipher suite for the TLS server (in OpenSSL cipher list format). The default is AES:RC4. For valid values, visit the OpenSSL website at https://www.openssl.org/docs/man1.0.2/apps/ciphers.html.
Web: Cipher Client CLI: ciphers-client [TLSContexts_ClientCipherString]	Defines the supported cipher suite for TLS clients. The default is DEFAULT. For possible values and additional details, visit the OpenSSL website at https://www.openssl.org/docs/man1.0.2/apps/ciphers.html.
Web: OCSP Server CLI: ocsf-server [TLSContexts_OcsfEnable]	Enables or disables certificate checking using OCSP. [0] Disable (default) [1] Enable
Web: Primary OCSP Server CLI: ocsf-server-primary [TLSContexts_OcsfServerPrimary]	Defines the IP address (in dotted-decimal notation) of the primary OCSP server. The default IP address is 0.0.0.0.
Web: Secondary OCSP Server CLI: ocsf-server-secondary [TLSContexts_OcsfServerSecondary]	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
Web: OCSP Port CLI: ocsf-port [TLSContexts_OcsfServerPort]	Defines the OCSP server's TCP port number. The default port number is 2560.

Parameter	Description
Web: OCSP Default Response CLI: ocsdp-default-response [TLSContexts_OcspDefaultResponse]	Determines whether the device allows or rejects peer certificates if it cannot connect to the OCSP server. ▪ [0] Reject (default) ▪ [1] Allow
DH Key Size [TLSContexts_DHKeySize]	Defines the Diffie-Hellman (DH) key size (in bits). DH is an algorithm used chiefly for exchanging cryptography keys used in symmetric encryption algorithms such as AES. ▪ [1024] 1024 (default) ▪ [2048] 2048

9.1.2 Assigning CSR-based Certificates to TLS Contexts

The following procedure describes how to request a digitally signed certificate from a Certification Authority (CA) for a TLS Context. This process is referred to as a certificate signing request (CSR) and is required if your organization employs a Public Key Infrastructure (PKI) system. The CSR contains information identifying the device (such as a distinguished name in the case of an X.509 certificate).

➤ **To assign a CSR-based certificate to a TLS Context:**

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. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Certificates**  button, located at the bottom of the TLS Contexts page; the Context Certificates page appears.
4. Under the **Certificate Signing Request** group, do the following:
 - a. In the 'Subject Name [CN]' field, enter the DNS name.
 - b. From the 'Signature Algorithm' drop-down list, select the hash function algorithm (SHA-1, SHA-256, or SHA-512) with which to sign the certificate.
 - c. Fill in the rest of the request fields according to your security provider's instructions.
 - d. Click the **Create CSR** button; a textual certificate signing request is displayed in the area below the button:

Figure 9-2: Certificate Signing Request Group

▼ Certificate Signing Request

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

Create CSR

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-----
MIIBtjCCAR8CAQAwZjESMBAGA1UEAxMJYXVkaW8uY29tMRUwEwYDVQQLZwI2ZWFK
cXVhcnRlcnMxZjAQBgNVBAOTCUNvbnBvcnF0ZTEVMBMGAA1UEBxMMUG91Z2hrZWVw
c21lMREwDwYDVQQIEWhvZ2VhZjEwYzE1MAkGA1UEBhMCVVMwZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBAPhpf2t4OLy3FRk5Bw7Fl2FWCXQ7nvuocHtu7Nns071M
xL7Of8YoL63eeIK2eDo8nm6rJO677z/AHWJmF65pAK1CboIPgOZNS0g6+5JAmJAA
1LNunoqjEsK7CF32uvolH//gFkhy5zleNvObi+25Pn38aJzEXc8DkGwZ19rROqRZ
AgMBAAGgADANBgkqhkiG9w0BAQQFAAQBggQDihdgbc1zkHdLFr+5BRuScKyGUXBM6
q7FGjFXAfzK1MmgnBMc/MYf3GTbawrQF7p6dNJ60DivmuCPf6Gzz5m2uqC6LqoTi
nLnQpVCmbdva/B1QyEpPbQhZqpULJ8CSeSrrY3ru23AZeDUBvYyho90IkRbAp//+3
ZvnZZe5M5CBSLg==
-----END CERTIFICATE REQUEST-----

```

5. Copy the text and send it to your security provider (CA) to sign this request.
6. When the CA sends you a server certificate, 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/MQswCQYDVQQGEw
JGUjETMBEGA1UEChMKQ2VydG1wb3N0ZTEbMBkGA1UEAxMSQ2VydG1wb3N0ZSBT
ZXJ2ZXVvMB4XDTE4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1
UEBhMCRLlxEzARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9z
dGUxGzU2VydMlVlcjCCASEwDQYJKoZIhvcNAQEBBQADggEoADCCAQkCggEAPqd4Mz
iR4spWldGRx8bQrhZkonWnNm`+Yhb7+4Q67ecf1janH7GcN/SXsfx7jJpreWUL
f7v7Cvpr4R7qIJcmDHIntmf7JPM5n6cDBv17uSW63er7NkVnMFHwK1QaGFLMyb
FkzaeGrvFm4k3lRefiXDmuOe+FhJgHYezYHf44LvPRPwhSrzi9+Ag3o8pWDguJ
uZDIUP1F1jMa+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.
9. Open the TLS Contexts page again, select the TLS Context index row, and then verify that under the **Certificate Information** group, the 'Private key' field displays "OK"; otherwise, consult your security administrator:

Figure 9-3: 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



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.

9.1.3 Assigning Externally Created Private Keys to TLS Contexts

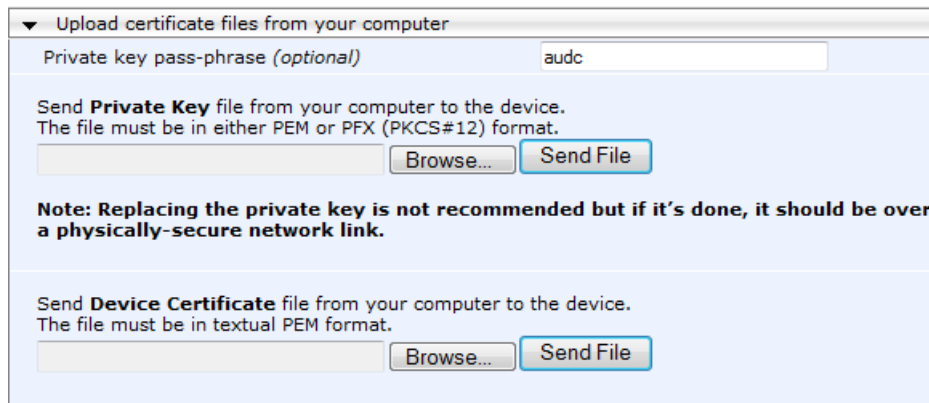
The following procedure describes how to assign an externally created private key to a TLS Context.

- **To assign an externally created private key to a TLS Context:**
 1. Obtain a private key in either textual PEM (PKCS #7) or PFX (PKCS #12) format (typically provided by your security administrator). The file may be encrypted with a short pass-phrase.
 2. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
 3. In the TLS Contexts table, select the required TLS Context index row, and then click

the **Context Certificates**  button, located at the bottom of the TLS Contexts page; the Context Certificates page appears.

4. Scroll down to the **Upload certificate files from your computer** group.

Figure 9-4: Upload Certificate Files from your Computer Group



▼ Upload certificate files from your computer

Private key pass-phrase (optional)

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.

5. Fill in the 'Private key pass-phrase' field, if required.
6. Click the **Browse** button corresponding to the 'Send Private Key' field, navigate to the private key file (Step 1), and then click **Send File**.
7. If the security administrator has provided you with a device certificate file, load it using the 'Send Device Certificate' field.
8. After the files successfully load to the device, save the configuration with a device reset.
9. Open the TLS Contexts page again, select the TLS Context index row, and then verify that under the **Certificate Information** group, the 'Private key' field displays "OK"; otherwise, consult your security administrator.

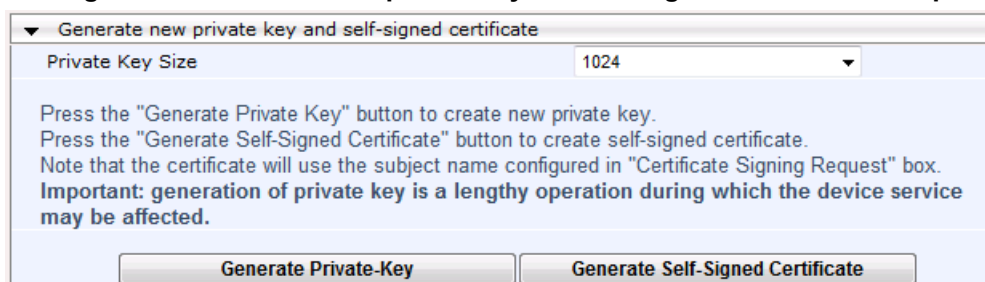
9.1.4 Generating Private Keys for TLS Contexts

The device can generate the private key for a TLS Context, as described in the procedure below. The private key can be generated for CSR or self-signed certificates.

➤ To generate a new private key for a TLS Context:

1. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
2. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Certificates**  button, located at the bottom of the TLS Contexts page; the Context Certificates page appears.
3. Scroll down to the **Generate new private key and self-signed certificate** group:

Figure 9-5: Generate new private key and self-signed certificate Group



▼ Generate new private key and self-signed certificate

Private Key Size

Press the "Generate Private Key" button to create new private key.
Press the "Generate Self-Signed Certificate" button to create self-signed certificate.
Note that the certificate will use the subject name configured in "Certificate Signing Request" box.
Important: generation of private key is a lengthy operation during which the device service may be affected.

4. From the 'Private Key Size' drop-down list, select the desired private key size (in bits) for RSA public-key encryption for newly self-signed generated keys:
 - 512
 - 1024 (default)
 - 2048
 - 4096
5. Click **Generate Private Key**; a message appears requesting you to confirm key generation.
6. Click **OK** to confirm key generation; the device generates a new private key, indicated by a message in the **Certificate Signing Request** group.

Figure 9-6: Indication of Newly Generated Private Key

▼ Certificate Signing Request

Subject Name [CN]	John Doe
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

Create CSR

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

A new 1024-bits Private-Key was generated for Context-ID=1
Please save the configuration.

7. Continue with the certificate configuration, by either creating a CSR or generating a new self-signed certificate.
8. Save the configuration with a device reset for the new certificate to take effect.

9.1.5 Creating Self-Signed Certificates for TLS Contexts

The following procedure describes how to assign a certificate that is digitally signed by the device itself to a TLS Context. In other words, the device acts as a CA.

➤ **To assign a self-signed certificate to a TLS Context:**

1. Before you begin, make sure that:
 - 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 therefore, must be listed in the server certificate.
 - No traffic is running on the device. The certificate generation process is disruptive to traffic and should be done during maintenance time.
2. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Certificates**  button, located at the bottom of the TLS Contexts page; the Context Certificates page appears.
4. Under the **Certificate Signing Request** group, in the 'Subject Name [CN]' field, enter the fully-qualified DNS name (FQDN) as the certificate subject.
5. Scroll down the page to the **Generate new private key and self-signed certificate**

group:

Figure 9-7: Generate new private key and self-signed certificate Group

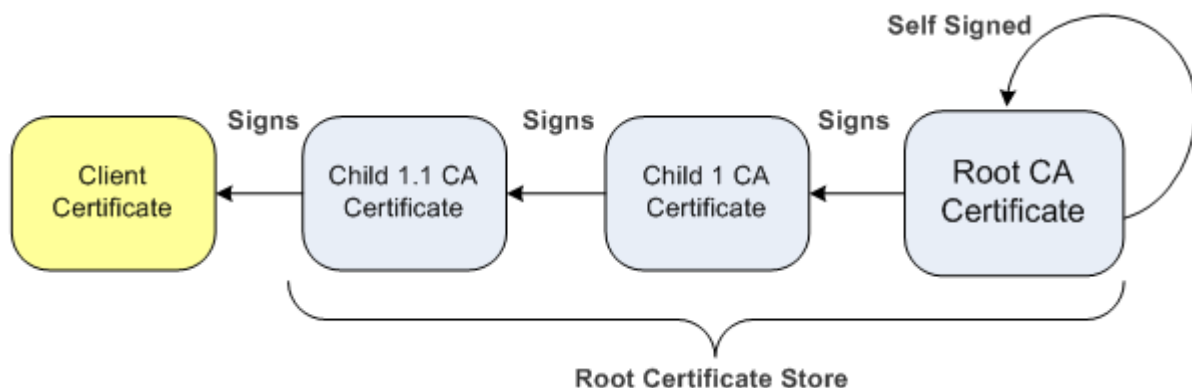
6. Click **Generate Self-Signed Certificate**; a message appears (after a few seconds) displaying the new subject name.
7. Save the configuration with a device reset for the new certificate to take effect.

9.1.6 Importing Certificates and Certificate Chain into Trusted Certificate Store

The device provides its own Trusted Root Certificate Store. This lets you manage certificate trust. You can add up to 20 certificates to the store per TLS Context (but this may be less depending on certificate file size).

The trusted store can also be used for certificate chains. 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 9-8: Certificate Chain Hierarchy



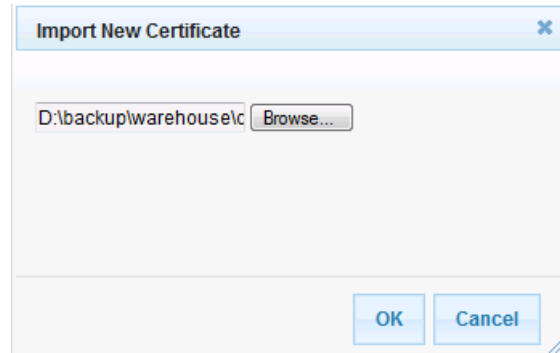
For the device to trust a whole chain of certificates per TLS Context, you need to add them to the device's Trusted Certificates Store, as described below.

➤ To import certificates into device's Trusted Root Certificate Store:

1. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
2. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Trusted-Roots**  button, located at the bottom of the TLS Contexts page; the Trusted Certificates page appears.

3. Click the **Import** button, and then select the certificate file to load.

Figure 9-9: Importing Certificate into Trusted Certificates Store



4. Click **OK**; the certificate is loaded to the device and listed in the Trusted Certificates store.

You can also do the following with certificates that are in the Trusted Certificates store:

- Delete certificates: Select the required certificate, click **Remove**, and then in the Remove Certificate dialog box, click **Remove**.
- Save certificates to a file on your PC: Select the required certificate, click **Export**, and then in the Export Certificate dialog box, browse to the folder on your PC where you want to save the file and click **Export**.

9.1.7 Configuring Mutual TLS Authentication

9.1.7.1 TLS for SIP Clients

When Secure SIP (SIPS) is implemented using TLS, it is sometimes required to use two-way (mutual) authentication between the device and a SIP user agent (client). When the device acts as the TLS server in a specific connection, the device demands the authentication of the SIP client's certificate. Both the device and the client use certificates from a CA to authenticate each other, sending their X.509 certificates to one another during the TLS handshake. Once the sender is verified, the receiver sends its' certificate to the sender for verification. SIP signaling starts when authentication of both sides completes successfully.

TLS mutual authentication can be configured for specific calls by enabling mutual authentication on the SIP Interface used by the call. The TLS Context associated with the SIP Interface or Proxy Set belonging to these calls are used.



Note: SIP mutual authentication can also be configured globally for all calls, using the 'TLS Mutual Authentication' parameter (SIPSRequireClientCertificate) in the General Security Settings page (**Configuration** tab > **VoIP** menu > **Security** > **General Security Settings**).

➤ To configure mutual TLS authentication for SIP messaging:

1. Enable two-way authentication on the specific SIP Interface:
 - a. In the SIP Interface Table page (see Configuring SIP Interfaces on page 236), set the 'TLS Mutual Authentication' parameter to **Enable** for the specific SIP Interface.
 - b. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

2. Configure a TLS Context with the following certificates:
 - Import the certificate of the CA that signed the certificate of the SIP client, into the Trusted Root Store so that the device can authenticate the client (see 'Importing Certificates and Certificate Chain into Trusted Certificate Store' on page 94).
 - Make sure that the TLS certificate is signed by a CA that the SIP client trusts so that the client can authenticate the device.

9.1.7.2 TLS for Remote Device Management

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 management 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.

➤ To enable mutual TLS authentication for HTTPS:

1. Set the 'Secured Web Connection (HTTPS)' field to **HTTPS Only** in the Web Security Settings page (see Configuring Web Security Settings on page 53) 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 TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Trusted-Roots**  button, located at the bottom of the TLS Contexts page; the Trusted Certificates page appears.
4. Click the **Import** button, and then select the certificate file.
5. When the operation is complete, set the 'Requires Client Certificates for HTTPS connection' field to **Enable** in the Web Security Settings page.
6. Save the configuration with a device reset (see Saving Configuration).

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 OCSP server, per TLS Context (see 'Configuring TLS Certificate Contexts' on page 85).

9.1.8 Configuring TLS Server Certificate Expiry Check

You can also configure the TLS Server Certificate Expiry Check feature, whereby the device periodically checks the validation date of the installed TLS server certificates. You can also configure the device to send a notification SNMP trap event (acCertificateExpiryNotification) at a user-defined number of days before the installed TLS server certificate is to expire. This trap event indicates the TLS Context to which the certificate belongs.



Note: TLS certificate expiry check is configured globally for all TLS Contexts.

➤ **To configure TLS certificate expiry checks and notification:**

1. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
2. Scroll down the page to the **TLS Expiry Settings** group:

Figure 9-10: TLS Expiry Settings Group

▼ TLS Expiry Settings	
TLS Expiry Check Start (days)	60
TLS Expiry Check Period (days)	7
Submit TLS Expiry Settings	

3. In the 'TLS Expiry Check Start' field, enter the number of days before the installed TLS server certificate is to expire at which time the device sends an SNMP trap event to notify of this.
4. 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.
5. Click the **Submit TLS Expiry Settings** button.

This page is intentionally left blank.

10 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.

10.1 Configuring Date and Time Manually

You can manually configure the date and time of the device (instead of using an NTP server), as described in the procedure below. You can also configure the following with your manually configured date and time:

- UTC time offset (e.g., GMT +1). To configure the offset, use the 'NTP UTC Offset' (NTPServerUTCOffset) parameter (see 'Configuring Automatic Date and Time using SNTP' on page 99)
- Daylight Saving Time (DST) - see 'Configuring Daylight Saving Time' on page 101

➤ **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 10-1: 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.

10.2 Configuring Automatic Date and Time using SNTP

The device's 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 device, as an 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. 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 an NTP server within the network. The client requests a time update from the user-defined NTP server (IP address or FQDN) at a user-defined update interval. Typically, this update interval is every 24 hours based on when the system was restarted.

You can also configure a time offset for the time received from the NTP server, according to your region. For example, Germany Berlin region is UTC/GMT +1 hours and therefore, you would configure the offset to "1". For USA New York, the UTC/GMT offset is -5 hours and therefore, the offset is a minus value and configured as "-5". To configure Daylight Saving Time (DST), see 'Configuring Daylight Saving Time' on page 101.

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, the device ignores NTP messages received without authentication.

The following procedure describes how to configure SNTP. For detailed descriptions of the configuration parameters, see NTP and Daylight Saving Time Parameters on page 531.

➤ **To configure SNTP using the Web interface:**

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).
2. Scroll down to the 'NTP Settings' group:

Figure 10-2: 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	

3. Configure the NTP server address:
 - In the 'NTP Server Address' (NTPServerIP) field, configure the primary NTP server's address (IP or FQDN).
 - In the 'NTP Secondary Server Address' (NTPSecondaryServerIP) field, configure the secondary NTP server.
4. In the 'NTP UTC Offset' (NTPServerUTCOffset) field, configure the time offset in relation to the UTC. For example, if your region is GMT +1 (an hour ahead), enter "1".
5. In the 'NTP Updated Interval' (NTPUpdateInterval) field, configure the period after which the date and time of the device is updated.
6. Configure NTP message authentication:
 - In the 'NTP Authentication Key Identifier' field, configure the NTP authentication key identifier.
 - In the 'NTP Authentication Secret Key' field, configure the secret authentication key shared between the device and the NTP server.
7. Verify that the device has received the correct date and time from the NTP server. You can do this by viewing the date and time in the Regional Settings page (see 'Configuring Date and Time Manually' on page 99).



Note: If the device receives no response from the NTP server, it polls the NTP server for 10 minutes. If there is still no response after this duration, the device declares the NTP server as unavailable, by sending an SNMP alarm (acNTPServerStatusAlarm). The failed response could be due to incorrect configuration.

10.3 Configuring Daylight Saving Time

You can apply daylight saving time (DST) to the date and time of the device. DST defines a date range in the year (summer) where the time is brought forward so that people can experience more daylight. DST applies an offset of up to 60 minutes (default) to the local time. For example, Germany Berlin has DST from 30 March to 26 October, where the time is brought forward by an hour (e.g., 02:00 to 03:00 on 30 March). Therefore, you would configure the DST offset to 60 minutes (one hour).

➤ **To configure DST using the Web interface:**

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).
2. Scroll down to the 'Day Light Saving Time' group:

Figure 10-3: Configuring DST

▼ Day Light Saving Time									
Day Light Saving Time	Enable ▼								
DST Mode	Day of year ▼								
Start Time	Mar ▼	30 ▼	2	:	0				
End Time	Oct ▼	26 ▼	3	:	0				
Offset [min]	60								
Day of Month Start	Mar ▼	Sunday ▼	First ▼	2	:	0			
Day of Month End	Oct ▼	Sunday ▼	First ▼	3	:	0			

3. From the 'Day Light Saving Time' (DayLightSavingTimeEnable) drop-down list, select **Enable**.
4. From the 'DST Mode' drop-down list, select the range type for configuring the start and end dates for DST:
 - **Day of year:** The range is configured by exact date (day number of month), for example, from March 30 to October 30. If 'DST Mode' is set to **Day of year**, in the 'Start Time' (DayLightSavingTimeStart) and 'End Time' (DayLightSavingTimeEnd) drop-down lists, configure the period for which DST is relevant.
 - **Day of month:** The range is configured by month and day type, for example, from the last Sunday of March to the last Sunday of October. If 'DST Mode' is set to **Day of month**, in the 'Day of Month Start' and 'Day of Month End' drop-down lists, configure the period for which DST is relevant.
5. In the 'Offset' (DayLightSavingTimeOffset) field, configure the DST offset in minutes.
6. If the current date falls within the DST period, verify that it has been successfully applied to the device's current date and time. You can view the device's date and time in the Regional Settings page (see 'Configuring Date and Time Manually' on page 99).

This page is intentionally left blank.

Part IV

General VoIP Configuration

11 Network

This section describes the network-related configuration.

11.1 Configuring Physical Ethernet Ports

The Physical Ports Settings table lets you configure the device's Ethernet ports. This includes port speed and duplex mode, Native VLAN (PVID), and a brief description.

The device's management tools (e.g., Web interface) use hard-coded strings to represent the physical ports. To view the mapping of the physical ports to these logical ports (strings) as well as view port status, 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
1	GE_2	00:1e:67:11:7c:29	100Mbps	FULL	DOWN	1

The following procedure 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:

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 11-1: Physical Port Settings Parameter Descriptions

Parameter	Description
Port CLI: port [PhysicalPortsTable_Port]	(Read-only) Displays the port number.

Parameter	Description
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" – Active port. When the Ethernet Group includes two ports and their transmit/receive mode is configured to 2RX 1TX or 2RX 2TX (see Configuring Ethernet Port Groups), both ports show "Active".

11.2 Configuring Ethernet Port Groups

The Ethernet Group Settings table lets you configure Ethernet Groups. An Ethernet Group represents a physical Ethernet port(s) on the device. You can assign an Ethernet Group with one, two, or no ports (*members*). When two ports are assigned to an Ethernet Group, 1+1 Ethernet port redundancy can be implemented in your network. This provides port redundancy within the Ethernet Group, whereby if an active port is disconnected, the device switches over to the other port in the Ethernet Group. If you configure an Ethernet Group with only one port, the Ethernet Group operates as a single port, without redundancy. You can also configure a combination of Ethernet Group types, where some contain one port and others two ports.

The Ethernet Group Settings table also lets you configure the transmit (Tx) and receive (Rx) settings for the Ethernet ports per Ethernet Group. The Tx/Rx setting applies only to Ethernet Groups that contain two ports. This setting determines whether both ports or only one of the ports can receive and/or transmit traffic.

The maximum number of Ethernet Groups that can be configured is the same as the number of Ethernet ports provided by the device. Thus, the device supports up to 12 Ethernet Groups, each containing 1 port, or up to 6 Ethernet Groups, each containing 2 ports. By default, each Ethernet Group is assigned one port.

You can assign Ethernet ports to IP network interfaces. This is done by first configuring an Ethernet Device with the required Ethernet Group containing the port or ports (see 'Configuring Underlying Ethernet Devices' on page 110). Then by assigning the Ethernet Device to the IP network interface in the Interface table (see 'Configuring IP Network Interfaces' on page 112). 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).

The following procedure 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.



Note: Before you can re-assign a port to a different Ethernet Group, you must first remove the port from its current Ethernet Group. To remove the port, either set the 'Member' field to **None** or to a different port.

➤ To configure Ethernet Groups:

1. Open the Ethernet Group Settings page (**Configuration** tab > **VoIP** menu > **Network** > **Ethernet Groups Table**).
2. If the port that you want to assign to a specific Ethernet Group is already associated with another Ethernet Group, you must first remove the port from the currently associated Ethernet Group before you can associate it with the desired Ethernet Group:
 - a. Select the Ethernet Group to which the port is currently associated, and then click **Edit**; the following dialog box appears:

Figure 11-1: Ethernet Group Settings Page

The screenshot shows a web-based configuration form for an Ethernet Group. The form has a title bar 'Edit Record #0' with a close button. Below the title bar are five rows of configuration fields: 'Index' with a text input containing '0', 'Group' with a dropdown menu showing 'GROUP_1', 'Mode' with a dropdown menu showing '2RX 1TX', 'Member 1' with a dropdown menu showing 'GE_1', and 'Member 2' with a dropdown menu showing 'None'. At the bottom right of the form are two buttons: 'Submit' and 'Cancel'.

- b. Set the 'Member 1' or 'Member 2' field (depending on where the port appears) to **None** (or to a different port).
- c. Click **Submit**; the port is removed from the Ethernet Group.
3. Select the Ethernet Group that you want to configure and associate a port(s), and then click **Edit**.
4. Configure the Ethernet Group according to the parameters described in the table below.
5. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 11-2: 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 Ethernet Group. This applies only to Ethernet Groups containing two ports. [3] 2RX/1TX = Both ports in the Ethernet 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 Ethernet Group can receive and transmit packets. [5] Single = (Default) If the Ethernet Group contains only one port, use this option. [6] None = If no port is assigned to the Ethernet Group, use this option. 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 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 369.

Parameter	Description
Member 1 CLI: member1 [EtherGroupTable_Member1]	Assigns the first port to the Ethernet Group. To assign no port, set this field to None . Note: Before you can re-assign a port to a different Ethernet Group, you must first remove the port from its current Ethernet Group. To remove the port, either set this field to None or to a different port.
Member 2 CLI: member2 [EtherGroupTable_Member2]	Assigns the second port to the Ethernet Group. To assign no port, set this field to None . Note: Before you can re-assign a port to a different Ethernet Group, you must first remove the port from its current Ethernet Group. To remove the port, either set this field to None or to a different port.

11.3 Configuring Underlying Ethernet Devices

The Ethernet Device table lets you configure up to 100 *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 multihoming (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 445).



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 following procedure 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:

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 11-3: Ethernet Device Table Parameter Descriptions

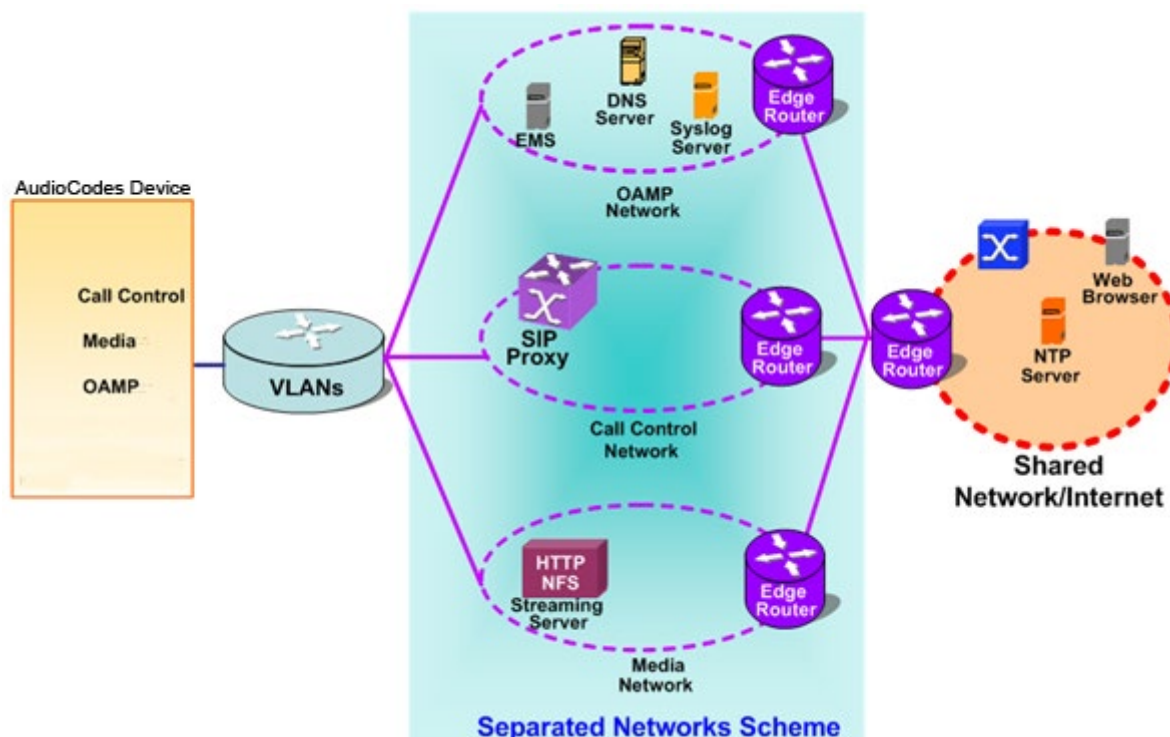
Parameter	Description
Index [DeviceTable_Index]	Defines an index number for the new table record. Note: Each table row must be configured with a unique index.

Parameter	Description
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 Ethernet Port Groups on page 107.
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 112) and/or with a static route in the Static Route table ('Device Name' field - see "Configuring Static IP Routing" on page 119). By default, the device automatically assigns a name using the following syntax: "dev <next available table row index>" (e.g., "dev 3").

11.4 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 11-2: Multiple Network Interfaces



The device is shipped with a default OAMP interface. For more information, see "Default OAMP IP Address" on page 21. 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 364.
- IP address (IPv4 and IPv6) and subnet mask (prefix length)

- For configuring Quality of Service (QoS), see "Configuring the QoS Settings" on page 123.
- 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 119.



Note: Before configuring IP interfaces, it is recommended that you read the IP interface configuration guidelines in "Interface Table Configuration Guidelines" on page 115.

The following procedure 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

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 445.

Table 11-4: 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 valid value of the prefix length depends on the IP address

Parameter	Description
	version: - IPv4: 0 to 30 (default is 16) - IPv6: 64 (default is 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.
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 Device CLI: underlying-dev [InterfaceTable_UnderlyingDevice]	Assigns an Ethernet Device to the IP interface. An Ethernet Device is a VLAN ID associated with a physical Ethernet port (Ethernet Group). To configure Ethernet Devices, see Configuring Underlying Ethernet Devices on page 110.

11.4.1 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.

11.4.2 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 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.

11.4.3 Networking Configuration Examples

This section provides configuration examples of networking interfaces.

11.4.3.1 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 11-5: Example of Single VoIP Interface in Interface Table

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Device	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 11-6: 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.

11.4.3.2 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 11-7: Example of VoIP Interfaces per Application Type in Interface Table

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Device	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 11-8: 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.

11.4.3.3 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 11-9: Example of VoIP Interfaces of Combined Application Types in Interface Table

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Device	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	MediaCntrl 1
2	Media & Control	IPv4 Manual	200.200.86.14	24	200.200.86.1	202	MediaCntrl 2

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Device	Interface Name
3	Media & Control	IPv6 Manual	2000::1:200:20 0:86:14	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 11-10: 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 11-3: Example of Layer-2 QoS in DiffServ Table

Diff Serv Table

Add + Edit Delete Show/Hide

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

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

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

Submit

11.4.3.4 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 11-11: Configured Default Gateway Example

Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underlying Device	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 11-12: Separate Static Route Table Example

Destination	Prefix Length	Gateway	Device Name
17.17.0.0	16	192.168.10.1	100
171.79.39.0	24	200.200.85.10	200

11.5 Configuring Static IP Routes

The Static Route table lets you configure up to 30 static IP routing rules. Using static routes lets you communicate with LAN networks that are not located behind the Default Gateway specified for the IP network interface, configured in the Interface table, from which the packets are sent.

Before sending an IP packet, the device searches the Static Route table for an entry that matches the requested destination host/network. If an entry is found, the device sends the packet to the gateway that is configured for the static route. If no explicit entry is found, the packet is sent to the Default Gateway configured for the IP network interface.

You can view the status of the configured static routes 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 445).

The following procedure 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	

Submit
Cancel

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



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

Static Route Table Parameter Descriptions

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]	Assigns an IP network interface through which the static route's Gateway is reached. The Device Name (or underlying device) represents the IP network interface, including VLAN ID and associated physical port(s). The value must be identical to the value in the 'Underlying Device' parameter of the required IP network interface in the Interface table (see Configuring IP Network Interfaces on page 112). For configuring Ethernet Devices, see Configuring Underlying Ethernet Devices on page 110.
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, the value 16 represents subnet 255.255.0.0. The valid value of the prefix length depends on the IP address version: - Pv4: 0 to 32 - IPv6: 0 to 128

Parameter	Description
	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 Gateway (next hop) used for traffic destined to the subnet/host defined in the 'Destination' / 'Prefix Length' field. Notes: ▪ The Gateway's address must be in the same subnet as the IP address of the network interface that is associated with the static route (using the 'Device Name' parameter - see above). ▪ The IP network interface associated with the static route must be of the same IP address family (IPv4 or IPv6).
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.

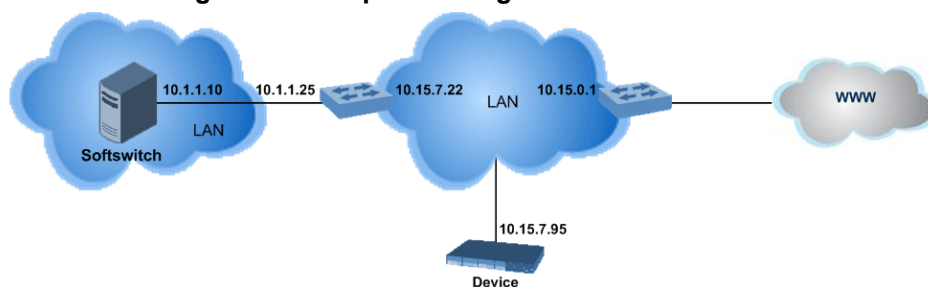
11.5.1 Configuration Example of Static IP Routes

An example of the use for static routes is shown in the figure below. In the example scenario, the device needs to communicate with a softswitch at IP address 10.1.1.10. However, the IP network interface from which packets destined for 10.1.1.10 is sent, is configured to send the packets to a Default Gateway at 10.15.0.1. Therefore, the packets do not reach the softswitch. To resolve this problem, a static route is configured to specify the correct gateway (10.15.7.22) in order to reach the softswitch.

Note the following configuration:

- The static route is configured with a subnet mask of 24 (255.255.255.0), enabling the device to use the static route to send all packets destined for 10.1.1.x to this gateway and therefore, to the network in which the softswitch resides.
- The static route in the Static Route table is associated with the IP network interface in the Interface table, using the 'Device Name' and 'Underlying Device' parameters, respectively.
- The static route's Gateway address in the Static Route table is in the same subnet as the IP address of the IP network interface in the Interface table.

Figure 4: Example of using Static Route



No Static Route:

The device sends packets to 10.15.0.1, which is the Default Gateway defined for this IP network interface in the Interface table. Therefore, the device will not succeed in reaching the softswitch.

Interface Table									
Add + Edit ✎ Delete 🗑 Show/Hide ⌵									
Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Interface Name	Primary DNS	Secondary DNS	Underlying Device
0	OAMP + Mec IPv4 Manual		10.15.7.95	16	10.15.0.1	Voice	0.0.0.0	0.0.0.0	vlan 1

Static Route Configured:

A static route with the correct gateway is needed for routing to the softswitch. The device communicates with the softswitch (10.1.1.0/24) using the gateway 10.15.7.22.

Note: The device first searches for a matching route in the Static Route table. If not found, it uses the default gateway defined in the Interface table.

Static Route Table					
Add + Edit ✎ Delete 🗑 Show/Hide ⌵					
Index	Device Name	Destination	Prefix Length	Gateway	Description
0	vlan 1	10.1.1.0	24	10.15.7.22	Softswitch

11.5.2 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 IP network interface associated with the static route.
- 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.

11.6 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 11-13: 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

Application	Traffic / Network Types	Class-of-Service (Priority)
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 115): - OAMP - Control	Depends on traffic type: - Control: Premium control - Management: Bronze

The following procedure 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 11-5: DiffServ Table Page - Add Record

- 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 11-14: 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.

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

Figure 11-6: QoS Settings Page - Differentiated Services

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

11.7 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 following procedure 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:

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

Figure 11-7: Configuring ICMP Messaging in Network Settings Page

Send and Receive ICMP Redirect Messages	Enable
Send ICMP Unreachable Messages	Disable

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

11.8 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 126
- Internal SRV table - see "Configuring the Internal SRV Table" on page 127

11.8.1 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 three 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 112).

The following procedure 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 11-8: 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 11-15: 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. Note: Currently, this parameter is not supported.

11.8.2 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 112).

The following procedure 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 11-9: Internal SRV Table Page

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 11-16: Internal SRV Table Parameter Descriptions

Parameter	Description
Domain Name CLI: domain-name [Srv2lp_InternalDomain]	Defines the host name to be translated. The valid value is a string of up to 31 characters.
Transport Type CLI: transport-type [Srv2lp_TransportType]	Defines the transport type. ▪ [0] UDP (default) ▪ [1] TCP ▪ [2] TLS
DNS Name (1-3) CLI: dns-name-1 2 3 [Srv2lp_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 [Srv2lp_Priority1/2/3]	Defines the priority of the target host. A lower value means that it is more preferred.
Weight (1-3) CLI: weight-1 2 3 [Srv2lp_Weight1/2/3]	Defines a relative weight for records with the same priority.
Port (1-3) CLI: port-1 2 3 [Srv2lp_Port1/2/3]	Defines the TCP or UDP port on which the service is to be found.

11.9 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.

11.9.1 Device Located behind NAT

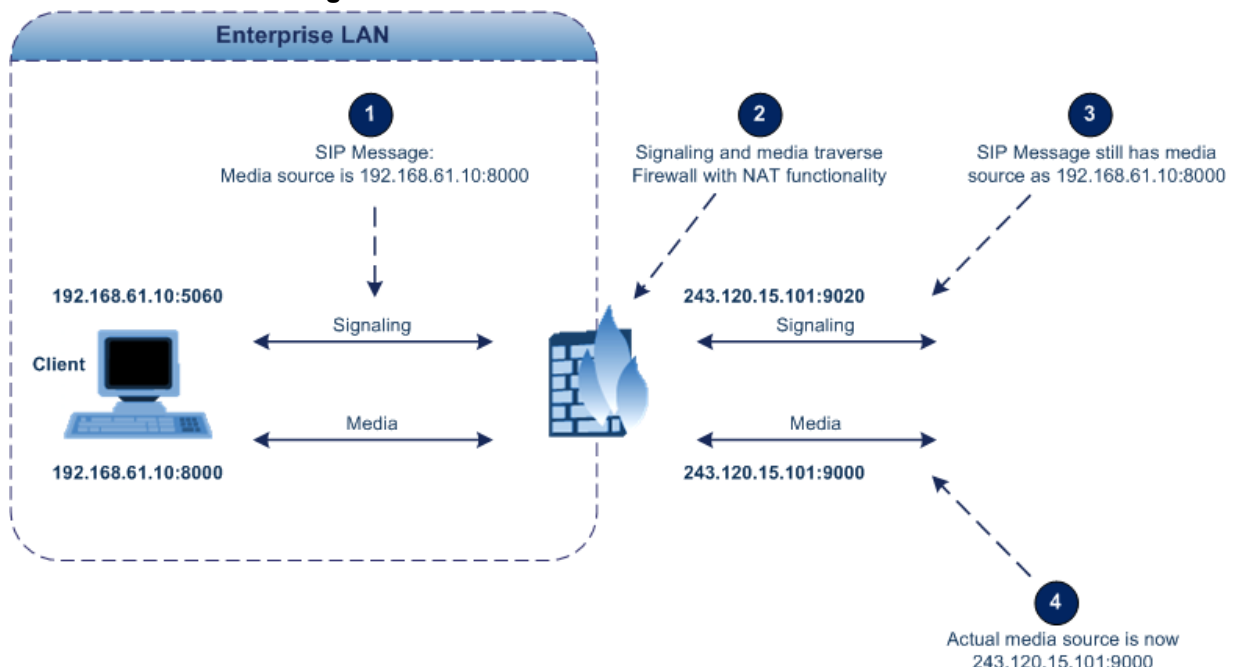
Two different streams traverse through NAT - signaling and media. A device located behind a NAT that initiates a signaling path has problems receiving incoming signaling responses as they are blocked by the NAT server. Therefore, the initiating device must inform the receiving device where to send the 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 130.
- b. If configured, uses the NAT Translation table which configures NAT per interface - see Configuring NAT Translation per IP Interface on page 130.

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 11-10: Device behind NAT and NAT Issues



11.9.1.1 Configuring a Static NAT IP Address for All Interfaces

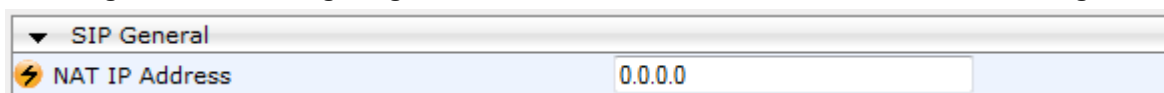
You can configure a global (public) IP address of the router to enable static NAT between the device and the Internet for all network interfaces. Thus, the device replaces the source IP address for media of all outgoing SIP messages sent on any of its network interfaces to this public IP address.

The following procedure 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 11-11: Configuring Static NAT IP Address in SIP General Parameters Page



The screenshot shows a web interface for 'SIP General' parameters. Under the 'SIP General' section, there is a field labeled 'NAT IP Address' with a lightning bolt icon to its left. The field contains the value '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.

11.9.1.2 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 into NAT IP addresses (*global - public*) when the device is located behind NAT. The device's NAT traversal mechanism replaces the source IP address of SIP messages sent from a specific VoIP interface (Control and/or Media) in the IP Interfaces table to a public IP address. 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 following procedure 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**).

- Click **Add**; the following dialog box appears:

Figure 11-12: NAT Translation Table Page

Field	Value
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

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

Table 11-17: 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.
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 (0-65535) 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 (0-65535) 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 (0-65535) 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.

Parameter	Description
Target End Port CLI: TargetEndPort [NATTranslation_TargetEndPort]	Defines the optional, ending port range (0-65535) 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.

11.9.2 Remote UA behind NAT

11.9.2.1 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 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 240). If this feature is disabled, the device's NAT detection is according to the settings of the global parameter, 'SIP NAT Detection' parameter.

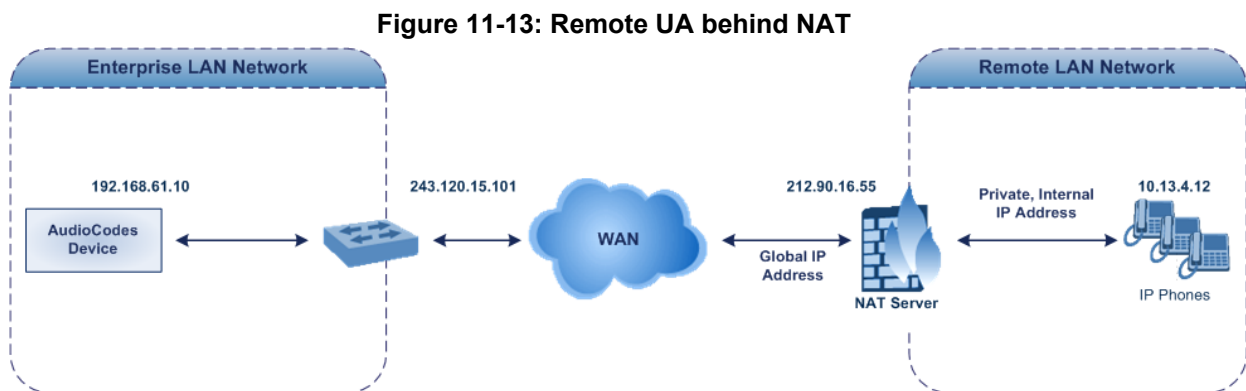
11.9.2.2 Media (RTP/RTCP/T.38)

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:port (UDP) indicated in the SDP body of the SIP message received from the UA. 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 133
- RTP No-Op packets according to the avt-rtp-noop draft - see "No-Op Packets" on page 134

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



11.9.2.2.1 First Incoming Packet Mechanism

In scenarios where the remote user agent (UA) resides behind a NAT server, it's possible that the device, if not configured for NAT traversal, will send the media (RTP, RTCP and T.38) streams to an invalid IP address / UDP port (i.e., private IP address:port of UA and not the public address). When the UA is located behind a NAT, although the UA sends its private IP address:port in the original SIP message (INVITE), the device receives the subsequent media packets with a source address of a public IP address:port (i.e., allocated by the NAT server). Therefore, to ensure that the media reaches the UA, the device must send it to the public address.

The device identifies whether the UA is located behind NAT, by comparing the source IP address of the first received media packet, with the IP address and UDP port of the first received SIP message (INVITE) when the SIP session was started. This is done for each media type--RTP, RTCP and T.38--and therefore, they can have different destination IP addresses and UDP ports than one another.

You can configure the device's NAT feature to operate in one of the following modes:

- **Auto-Detect:** NAT is performed only if necessary. If the UA is identified as being located behind NAT, the device sends the media packets to the public IP address:port obtained from the source address of the first media packet received from the UA. Otherwise, the packets are sent using the IP address:port obtained from the first received SIP message. Note that if the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA, does it determine whether the UA is behind NAT.

- **NAT Is Not Used:** (Default) NAT feature is disabled. The device considers the UA as not located behind NAT and always sends the media packets to the UA using the IP address:port obtained from the first received SIP message.
 - **NAT Is Used:** NAT is always performed. The device considers the UA as located behind NAT and always sends the media packets to the UA using the source address obtained from the first media packet received from the UA. In this mode, the device does not send any packets until it receives the first packet from the UA (in order to obtain the IP address).
- **To enable NAT resolution using the First Incoming Packet mechanism:**
1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **Media** > **General Media Settings**).
 2. Set the 'NAT Mode' parameter to one of the following:
 - [0] Auto-Detect
 - [1] NAT Is Not Used
 - [2] NAT Is Used
 3. Click **Submit**.

11.9.2.2.2 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 523.

- **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 523). 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.

11.10 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 11-14: 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 [542](#).

11.11 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.

This page is intentionally left blank.

12 Security

This section describes the VoIP security-related configuration.

12.1 Configuring Firewall Settings

The Firewall Settings table lets you configure the device's Firewall, which defines network traffic filtering rules (*access list*) for incoming traffic. 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 55), 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 375.



The following procedure 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 12-1: Firewall Settings Page - Add Record

Index	0
Source IP	0.0.0.0
Source Port	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	

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 12-1: 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 from where the device receives the incoming packet.
Source Port CLI: src-port [AccessList_Source_Port]	Defines the source UDP/TCP ports of the remote host from where the device receives the incoming packet. 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 first UDP/TCP port in the range of ports on the device on which the incoming packet is received. From the perspective of the remote IP entity, this is the destination port. To configure the last port in the range, see the 'End Port' parameter (below). 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 last UDP/TCP port in the range of ports on the device on which the incoming packet is received. From the perspective of the remote IP entity, this is the destination port. To configure the first port in the range, see the 'Start Port' parameter (above). 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

Parameter	Description
	"Configuring IP Network Interfaces" on page 112.
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 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 12-2: 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.

12.2 Configuring General Security Settings

The device uses TLS over TCP to encrypt and optionally, authenticate SIP messages. This is referred to as Secure SIP (SIPS). SIPS uses the X.509 certificate exchange process, as described in 'Configuring SSL/TLS Certificates' on page 85, where you need to configure certificates (TLS Context).



Notes:

- When a TLS connection with the device is initiated by a SIP client, the device also responds using TLS, regardless of whether or not TLS was configured.
- For backward compatibility, the following parameters can be used:
 - ✓ SIPTransportType to enable TLS.
 - ✓ TLSLocalSIPPort to configure the device's port used for TLS traffic.

➤ To configure SIPS:

1. Configure a TLS Context as required.
2. Assign the TLS Context to a Proxy Set or SIP Interface (see Configuring Proxy Sets on page 251 and Configuring SIP Interfaces on page 236, respectively).
3. Configure a SIP Interface with a TLS port number.
4. Configure various SIPS parameters in the General Security Settings page (**Configuration** tab > **VoIP** menu > **Security** > **General Security Settings**).

For a description of the TLS parameters, see TLS Parameters on page 548.

5. By default, 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 the 'Enable SIPS' (EnableSIPS) parameter to **Enable** in the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**).

12.3 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.
- 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 150.

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).

12.3.1 Enabling IDS

The following procedure 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 12-2: 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.

12.3.2 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 12-3: IDS Policy Table with Default Rules

Add + Edit ✎ Delete -			Show/Hide ☏
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			

- Click **Add**; the following dialog box appears:

Figure 12-4: IDS Policy Table - Add Record

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

Table 12-3: 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.

- 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 12-5: 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 12-6: IDS Rule Table - 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 12-4: 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. ▪ [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 332) ✓ Routing failure ✓ Other local rejects (prior to SIP 180 response) ✓ Remote rejects (prior to SIP 180 response)

Parameter	Description
	▪ [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-alm-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-alm-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-alm-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 [IDSRule_DenyPeriod]	Defines the duration (in sec) to keep the attacker on the blacklist. The valid range is 0 to 1,000,000. The default is -1 (i.e., not configured).

12.3.3 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 236.
- **Proxy Sets:** For detection of malicious attacks from specified Proxy Set(s). For configuring Proxy Sets, see "Configuring Proxy Sets" on page 251.
- **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 12-7: IDS Match Table - Add Record

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 12-5: 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 CLI: proxy-set	Defines the Proxy Set(s) to which the IDS Policy is assigned. This indicates the Proxy Sets from where the attacks are coming from. The

Parameter	Description
[IDSMatch_ProxySet]	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 145).

12.3.4 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, IDSAAlarmClearPeriod) 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 145). For example, if you set IDSAAlarmClearPeriod to 20 sec

and 'Threshold Window' to 15 sec, the IDSArmClearPeriod 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 439). In this example, a Minor threshold alarm is cleared and replaced by a Major threshold alarm:

Figure 12-8: IDS Alarms in Active Alarms Table

17	Minor	Board#1/IDSMATCH#2/IDSRULE#0	Policy 2 (Proxy): minor threshold (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 threshold (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 threshold (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 <IDS Match Policy ID> rule
<IDS Rule ID>
```

The IP address is displayed only if the 'Threshold Scope' parameter is set to **IP** or **IP+Port**; otherwise, only the alarm is displayed.

- 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 and alarms in Syslog messages to a Syslog server. This only occurs if you have configured Syslog (see "Enabling Syslog" on page 473). An example of a Syslog message with IDS alarms and notifications is shown below:

Figure 12-9: Syslog Message Example with IDS Alarms and Notifications

```
[S=92159] [SID:438286865] ( lgr_ids(97420 ) IDS Event: reason=establish-fail,event=14003(establish-classify-fail),ip=10.13.45.200:5060(SII),transport=udp
[S=92160] [SID:438286865] ( lgr_ids(97421 ) IDS Counter (0,19995): IDSMATCH#0/IDSRULE#0,policy=3(TEST),reason=establish-fail,scope=ip,scope-val=10.13.45.200(SII),value=6
[S=92161] [SID:438286865] ( lgr_ids(97422 ) ?? [WARNING] IDS Rule (0): Threshold cross. IDSMATCH#0/IDSRULE#0,policy=3(TEST),value=6,severity=2(major)
[S=92162] [SID:438286865] ( lgr_ids(97423 ) ?? [WARNING] IDS Rule (0): Threshold cross. IDSMATCH#0/IDSRULE#0,policy=3(TEST),value=6,severity=4(blacklist)
[S=92163] [SID:438286865] ( lgr_ids(97424 ) ?? [WARNING] IDS Blacklist: Added IP 10.13.45.200(NI:0) to blacklist
[S=92164] [SID:438286865] ( lgr_psrdrif(97425 ) SNMP EVENT: IDS_BLACKLIST_NOTIFY "Added IP 10.13.45.200(NI:0) to blacklist"
[S=92165] RAISE-ALARM:acIDSBlacklistNotification; Textual Description: Added IP 10.13.45.200(NI:0) to blacklist; Severity:indeterminate; Source; Unique ID:30;
[S=92166] [SID:438286865] ( lgr_psrdrif(97426 ) InsertBoardEvent- event ADD BLACKLIST EV inserted channel -100
```


The table below lists the Syslog text messages per malicious event:

Table 12-6: 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

13 Media

This section describes the media-related configuration.

13.1 Configuring RTP/RTCP Settings

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

13.1.1 Configuring RTP Base UDP Port

You can configure the range of local UDP ports for RTP, RTCP, and T.38 media streams. The range of possible UDP ports that can be used, depending on configuration, is 6,000 through to 65,535. The device assigns ports randomly to the traffic within the configured port range.

For RTCP and T.38 traffic, the port offset from the RTP port used for the voice session is one and two, respectively. For example, if the voice session uses RTP port 6000, the RTCP port and T.38 port for the session is 6001 and 6002, respectively. However, you can configure the device to use the same port for RTP and T.38 packets, by setting the T38UseRTPPort parameter to 1.

Within the port range, the device allocates the UDP ports in "jumps" (spacing) of 5 (default) or 10, configured by the UdpPortSpacing parameter. For example, if the port range starts at 6000 and the UDP port spacing is 10, the available ports include 6000, 6010, 6020, 6030, and so on.

The port range is calculated using the following equation:

$$\text{BaseUDPPort to 65,535}$$

Where, *BaseUDPPort* is a parameter for configuring the lower boundary of the UDP port range (default is 6000).

For example, if the base UDP port is set to 6000, the port range is 6000 to 65,535.

You can also configure specific port ranges for specific SIP entities, using Media Realms (see Configuring Media Realms on page 227). You can configure each Media Realm with a different UDP port range and then associate the Media Realm with a specific IP Group, for example. However, the port range of the Media Realm must be within the range configured by the BaseUDPPort parameter.

The following procedure describes how to configure the RTP base UDP port in 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 13-1: RTP Based UDP Port in RTP/RTCP Settings Page



The screenshot shows a web interface for configuring RTP/RTCP settings. It features a tabbed interface with 'Configuration' selected. Under the 'VoIP' menu, the 'Media' section is active, displaying the 'RTP/RTCP Settings' page. A group of settings titled 'General Settings' is visible, containing a parameter labeled 'RTP Base UDP Port' with a lightning bolt icon. The value '6000' is entered in the adjacent text input field.

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.



Note:

- The RTP port must be different from ports configured for SIP signaling traffic (i.e., ports configured for SIP Interfaces). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.
- The base UDP port number (BaseUDPPort parameter) must be greater than the highest UDP port configured for a SIP Interface (see Configuring SIP Interfaces on page 236). For example, if your highest configured UDP port for a SIP Interface is 6060, you must configure the BaseUDPPort parameter to any value greater than 6060.

13.2 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.

You can also enable the device to validate the authentication of packets for SRTP tunneling for RTP and RTCP. This applies only to SRTP-to-SRTP SBC calls and where the

endpoints use the same key. This is configured using the 'SRTP Tunneling Authentication for RTP' and 'SRTP Tunneling Authentication for RTCP' parameters.



Notes:

- For a detailed description of the SRTP parameters, see "SRTP Parameters" on page 545.
- 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 390.

This page is intentionally left blank.

14 Services

This section describes configuration for various supported services.

14.1 DHCP Server Functionality

The device can serve as a Dynamic Host Configuration Protocol (DHCP) server that assigns and manages IP addresses from a user-defined address pool for DHCP clients. The DHCP server can also be configured to supply additional information to the requesting client such as the IP address of the TFTP server, DNS server, NTP server, and default router (gateway). The DHCP server functionality complies with IETF RFC 2131 and RFC 2132.

The DHCP server can service up to 25,000 DHCP clients. The DHCP clients are typically IP phones that are connected to the device's LAN port.

The DHCP server is activated when you configure a valid entry in the DHCP Servers table (see "Configuring the DHCP Server" on page 157) and associate it with an active IP network interface (listed in the Interface table). When an IP phone on the LAN requests an IP address, the DHCP server allocates one from the address pool. In scenarios of duplicated IP addresses on the LAN (i.e., an unauthorized network device using one of the IP addresses of the DHCP address pool), the DHCP server detects this condition using an Address Resolution Protocol (ARP) request and temporarily blacklists the duplicated address.

You can also configure the DHCP server to respond only to DHCPDiscover requests from DHCP clients that contain a specific value for Option 60 (Vendor Class Identification). For more information, see "Configuring the Vendor Class Identifier" on page 161.

14.1.1 Configuring the DHCP Server

The DHCP Servers table lets you configure the device's DHCP server. The DHCP Server table configures the DHCP server implementation. This includes configuring the DHCP IP address pool from where IP addresses are allocated to requesting DHCP clients, as well as configuring other information such as IP addresses of the DNS server, NTP server, default router (gateway), and SIP proxy server. The DHCP server sends the information in DHCP Options. The table below lists the DHCP Options that the DHCP server sends to the DHCP client and which are configurable in the DHCP Servers table.

Table 14-1: Configurable DHCP Options in DHCP Servers Table

DHCP Option Code	DHCP Option Name
Option 53	DHCP Message Type
Option 54	DHCP Server Identifier
Option 51	IP Address Lease Time
Option 1	Subnet Mask
Option 3	Router
Option 6	Domain Name Server
Option 44	NetBIOS Name Server
Option 46	NetBIOS Node Type
Option 42	Network Time Protocol Server
Option 2	Time Offset

DHCP Option Code	DHCP Option Name
Option 66	TFTP Server Name
Option 67	Boot file Name
Option 120	SIP Server

Once you have configured the DHCP server, you can configure the following:

- DHCP Vendor Class Identifier names (DHCP Option 60) - see "Configuring the Vendor Class Identifier" on page 161
- Additional DHCP Options - see "Configuring Additional DHCP Options" on page 162
- Static IP addresses for DHCP clients - see "Configuring Static IP Addresses for DHCP Clients" on page 164



Note: If you configure additional DHCP Options in the DHCP Option table, they override the default ones, which are configured in the DHCP Servers table. For example, if you configure Option 67 in the DHCP Option table, the device uses the value configured in the DHCP Option table instead of the value configured in the DHCP Servers table.

To view and delete currently serviced DHCP clients, see "Viewing and Deleting DHCP Clients" on page 165.

The following procedure describes how to configure the DHCP server in the Web interface. You can also configure this using the table ini file parameter, DhcpServer or CLI command, configure voip > dhcp server <index>.

➤ **To configure the device's DHCP server:**

1. Open the DHCP Servers page (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. Click **Add**; the following dialog box appears:

Figure 14-1: DHCP Servers Table - Add Record Dialog Box

The 'Add Record' dialog box for DHCP Servers contains the following fields and values:

Index	0
Interface Name	None
Start IP address	100.0.168.192
End IP address	149.0.168.192
Subnet Mask	0.255.255.255
Lease time	1440
DNS server 1	0.0.0.0
DNS server 2	0.0.0.0
NetBIOS name server	0.0.0.0
NetBIOS note type	Broadcast
NTP server 1	0.0.0.0
NTP server 2	0.0.0.0
Time offset	0
TFTP server name	
Boot file name	
Expand boot-file name	Yes
Override router	0.0.0.0
SIP server	
SIP server type	DNS name

Buttons: Submit, Cancel

3. Configure a DHCP server according to the parameters described in the table below.
4. Click **Submit**.

Table 14-2: DHCP Servers Table Parameter Descriptions

Parameter	Description
Web: Index CLI: dhcp server <index>	Defines an index number for the new table record. Notes: - Each table row must be configured with a unique index. - Currently, only one index row can be configured.
Web: Interface Name CLI: network-if [DhcpServer_InterfaceName]	Associates an IP interface on which the DHCP server operates. The IP interfaces are configured in the Interface table (see Configuring IP Network Interfaces). By default, no value is defined.
Web: Start IP Address CLI: start-address [DhcpServer_StartIPAddress]	Defines the starting IP address (IPv4 address in dotted-decimal format) of the IP address pool range used by the DHCP server to allocate addresses. The default value is 192.168.0.100. Note: The IP address must belong to the same subnet as the associated interface's IP address.
Web: End IP Address CLI: end-address [DhcpServer_EndIPAddress]	Defines the ending IP address (IPv4 address in dotted-decimal format) of the IP address pool range used by the DHCP server to allocate addresses. The default value is 192.168.0.149. Note: The IP address must belong to the same subnet as the associated interface's IP address and must be "greater or equal" to the starting IP address defined in 'Start IP Address'.
Web: Subnet Mask CLI: subnet-mask [DhcpServer_SubnetMask]	Defines the subnet mask (for IPv4 addresses) for the DHCP client. The value is sent in DHCP Option 1 (Subnet Mask). The default value is 0.0.0.0. Note: The value must be "narrower" or equal to the subnet mask of the associated interface's IP address. If set to "0.0.0.0", the subnet mask of the associated interface is used.
Web: Lease Time CLI: lease-time [DhcpServer_LeaseTime]	Defines the duration (in minutes) of the lease time to a DHCP client for using an assigned IP address. The client needs to request a new address before this time expires. The value is sent in DHCP Option 51 (IP Address Lease Time). The valid value range is 0 to 214,7483,647. The default is 1440. When set to 0, the lease time is infinite.
Web: DNS Server 1 CLI: dns-server-1 [DhcpServer_DNSServer1]	Defines the IP address (IPv4) of the primary DNS server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 6 (Domain Name Server). The default value is 0.0.0.0.
Web: DNS Server 2 CLI: dns-server-2 [DhcpServer_DNSServer2]	Defines the IP address (IPv4) of the secondary DNS server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 6 (Domain Name Server).

Parameter	Description
Web: NetBIOS Name Server CLI: netbios-server [DhcpServer_NetbiosNameServer]	Defines the IP address (IPv4) of the NetBIOS WINS server that is available to a Microsoft DHCP client. The value is sent in DHCP Option 44 (NetBIOS Name Server). The default value is 0.0.0.0.
Web: NetBIOS Node Type CLI: netbios-node-type [DhcpServer_NetbiosNodeType]	Defines the node type of the NetBIOS WINS server for a Microsoft DHCP client. The value is sent in DHCP Option 46 (NetBIOS Node Type). [0] Broadcast (default) [1] peer-to-peer [4] Mixed [8] Hybrid
Web: NTP Server 1 CLI: ntp-server-1 [DhcpServer_NTPServer1]	Defines the IP address (IPv4) of the primary NTP server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 42 (Network Time Protocol Server). The default value is 0.0.0.0.
Web: NTP Server 2 CLI: ntp-server-2 [DhcpServer_NTPServer2]	Defines the IP address (IPv4) of the secondary NTP server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 42 (Network Time Protocol Server). The default value is 0.0.0.0.
Web: Time Offset CLI: time-offset [DhcpServer_TimeOffset]	Defines the Greenwich Mean Time (GMT) offset (in seconds) that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 2 (Time Offset). The valid range is -43200 to 43200. The default is 0.
Web: TFTP Server CLI: tftp-server-name [DhcpServer_TftpServer]	Defines the IP address or name of the TFTP server that the DHCP server assigns to the DHCP client. The TFTP server typically stores the boot file image, defined in the 'Boot file name' parameter (see below). The value is sent in DHCP Option 66 (TFTP Server Name). The valid value is a string of up to 80 characters. By default, no value is defined.
Web: Boot file name CLI: boot-file-name [DhcpServer_BootFileName]	Defines the name of the boot file image for the DHCP client. The boot file stores the boot image for the client. The boot image is typically the operating system the client uses to load (downloaded from a boot server). The value is sent in DHCP Option 67 (Bootfile Name). To define the server storing the file, use the 'TFTP Server' parameter (see above). The valid value is a string of up to 256 characters. By default, no value is defined. The name can also include the following case-sensitive placeholder strings that are replaced with actual values if the 'Expand Boot-file Name' parameter is set to Yes : <MAC>: Replaced by the MAC address of the client (e.g., <i>boot_<MAC>.ini</i>). The MAC address is obtained in the client's DHCP request. <IP>: Replaced by the IP address assigned by the DHCP server to the client.

Parameter	Description
Web: Expand Boot-file Name CLI: expand-boot-file-name [DhcpServer_ExpandBootfileName]	Enables the use of the placeholders in the boot file name, defined in the 'Boot file name' parameter. [0] No [1] Yes (default)
Web: Override Router CLI: override-router-address [DhcpServer_OverrideRouter]	Defines the IP address (IPv4 in dotted-decimal notation) of the default router that the DHCP server assigns the DHCP client. The value is sent in DHCP Option 3 (Router). The default value is 0.0.0.0. If not specified (empty or "0.0.0.0"), the IP address of the default gateway configured in the Interface table for the IP network interface that you associated with the DHCP server (see the 'Interface Name' parameter above) is used.
Web: SIP Server CLI: sip-server [DhcpServer_SipServer]	Defines the IP address or DNS name of the SIP server that the DHCP server assigns the DHCP client. The client uses this SIP server for its outbound SIP requests. The value is sent in DHCP Option 120 (SIP Server). After defining this parameter, use the 'SIP server type' parameter (see below) to define the type of address (FQDN or IP address). The valid value is a string of up to 256 characters. The default is 0.0.0.0.
Web: SIP server type CLI: sip-server-type [DhcpServer_SipServerType]	Defines the type of SIP server address. The actual address is defined in the 'SIP server' parameter (see above). Encoding is done per SIP Server Type, as defined in RFC 3361. [0] DNS names = (Default) The 'SIP server' parameter is configured with an FQDN of the SIP server. [1] IP address = The 'SIP server' parameter is configured with an IP address of the SIP server.

14.1.2 Configuring the Vendor Class Identifier

The DHCP Vendor Class table lets you configure up to 10 Vendor Class Identifier (VCI) names (DHCP Option 60). When the table is configured, the device's DHCP server responds only to DHCPDiscover requests that contain Option 60 and that match one of the DHCP VCIs configured in the table. If you have not configured any entries in the table, the DHCP server responds to all DHCPDiscover requests, regardless of the VCI.

The VCI is a string that identifies the vendor and functionality of a DHCP client to the DHCP server. For example, Option 60 can show the unique type of hardware (e.g., "AudioCodes 440HD IP Phone") or firmware of the DHCP client. The DHCP server can then differentiate between DHCP clients and process their requests accordingly.

The following procedure describes how to configure the DHCP VCIs in the Web interface. You can also configure this using the table ini file parameter, DhcpVendorClass or CLI command, configure voip > dhcp vendor-class.

➤ **To configure DHCP Vendor Class Identifiers:**

1. Open the DHCP Servers page (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the DHCP Servers table, select the row of the desired DHCP server for which you want to configure VCIs, and then click the **DHCP Vendor Class Table** link located at the bottom of the page; the DHCP Vendor Class Table page opens.

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

Figure 14-2: DHCP Vendor Class Table - Add Record Dialog Box

4. Configure a VCI for the DHCP server according to the parameters described in the table below.
5. Click **Submit**.

Table 14-3: DHCP Vendor Class Table Parameter Descriptions

Parameter	Description
Web: Index CLI: dhcp vendor-class <index> [DhcpVendorClass_Index]	Defines an index number for the new table record. Note: Each table row must be configured with a unique index.
Web: DHCP Server Index CLI: dhcp-server-number [DhcpVendorClass_DhcpServerIndex]	Associates the VCI table entry with a DHCP server that you configured in "Configuring the DHCP Server" on page 157. Note: Currently, only one DHCP server (Index 0) can be configured and therefore, this parameter is always set at 0.
Web: Vendor Class Identifier CLI: vendor-class [DhcpVendorClass_VendorClassId]	Defines the value of the VCI DHCP Option 60. The valid value is a string of up to 80 characters. By default, no value is defined.

14.1.3 Configuring Additional DHCP Options

The DHCP Option table lets you configure up to 10 additional DHCP Options that the DHCP server can use to service the DHCP client. These DHCP Options are included in the DHCP Offer response sent by the DHCP server.

The following procedure describes how to configure DHCP Options in the Web interface. You can also configure this using the table ini file parameter, DhcpOption or CLI command, configure voip > dhcp option.



Note: The additional DHCP Options configured in the DHCP Option table override the default ones, which are configured in the DHCP Servers table. In other words, if you configure Option 67 in the DHCP Option table, the device uses the value configured in the DHCP Option table instead of the value configured in the DHCP Servers table.

➤ To configure DHCP Options:

1. Open the DHCP Servers page (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the DHCP Servers table, select the row of the desired DHCP server for which you want to configure additional DHCP Options, and then click the **DHCP Option Table** link located at the bottom of the page; the DHCP Option Table page opens.

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

Figure 14-3: DHCP Option Table - Add Record Dialog Box

4. Configure additional DHCP Options for the DHCP server according to the parameters described in the table below.
5. Click **Submit**.

Table 14-4: DHCP Option Table Parameter Descriptions

Parameter	Description
Web: Index CLI: dhcp option [DhcpOption_Index]	Defines an index number for the new table record. Note: Each table row must be configured with a unique index.
Web: DHCP Server Index CLI: dhcp-server-number [DhcpOption_DhcpServerIndex]	Associates the DHCP Option table entry with a DHCP server that you configured in "Configuring the DHCP Server" on page 157. Note: Currently, only one DHCP server (Index 0) can be configured and therefore, this parameter is always set at 0.
Web: Option CLI: option [DhcpOption_Option]	Defines the code of the DHCP Option. The valid value is 1 to 254. The default is 159. For example, for DHCP Option 150 (Cisco proprietary for defining multiple TFTP server IP addresses), enter the value 150.
Web: Type CLI: type [DhcpOption_Type]	Defines the format (type) of the DHCP Option value that is configured in the 'Value' parameter (see below). [0] ASCII = (Default) Plain-text string (e.g., when the value is a domain name). [1] IP address = IPv4 address. [2] Hexadecimal = Hexadecimal-encoded string. For example, if you set the 'Value' parameter to "company.com", you need to set the 'Type' parameter to ASCII .

Parameter	Description
Web: Value CLI: value [DhcpOption_Value]	Defines the value of the DHCP Option. For example, if you are using Option 66, this parameter is used for specifying the TFTP provisioning server (e.g., <code>http://192.168.3.155:5000/provisioning/</code>). The valid value is a string of up to 256 characters. By default, no value is defined. For IP addresses, the value can be one or more IPv4 addresses, each separated by a comma (e.g., <code>192.168.10.5,192.168.10.20</code>). For hexadecimal values, the value is a hexadecimal string (e.g., <code>c0a80a05</code>). You can also configure the parameter with case-sensitive placeholder strings that are replaced with actual values if the 'Expand Value' parameter (see below) is set to Yes: ▪ <MAC>: Replaced by the MAC address of the client. The MAC address is obtained from the client's DHCP request. For example, the parameter can be set to: <code>http://192.168.3.155:5000/provisioning/cfg_<MAC>.txt</code> ▪ <IP>: Replaced by the IP address assigned by the DHCP server to the client. For example, the parameter can be set to: <code>http://192.168.3.155:5000/provisioning/cfg_<IP>.txt</code>
Web: Expand Value CLI: expand-value [DhcpOption_ExpandValue]	Enables the use of the special placeholder strings, "<MAC>" and "<IP>" for configuring the 'Value' parameter (see above). ▪ [0] No ▪ [1] Yes (default) Note: This parameter is applicable only to values of type ASCII (see the 'Type' parameter above).

14.1.4 Configuring Static IP Addresses for DHCP Clients

The DHCP Static IP table lets you configure up to 100 DHCP clients with static IP addresses. The static IP address is a "reserved" IP address for a specified DHCP client defined by MAC address. In other words, instead of assigning the DHCP client with a different IP address upon each IP address lease renewal request, the DHCP server assigns the client the same IP address. For DHCP clients that are not listed in the table, the DHCP server assigns a random IP address from its address pool, as in normal operation.

The following procedure describes how to configure static IP addresses for DHCP clients in the Web interface. You can also configure this using the table ini file parameter, `DhcpStaticIP` or CLI command, `configure voip > dhcp static-ip <index>`.

➤ To configure static IP addresses for DHCP clients:

1. Open the DHCP Servers page (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the DHCP Servers table, select the row of the desired DHCP server for which you want to configure static IP addresses for DHCP clients, and then click the **DHCP Static IP Table** link located at the bottom of the page; the DHCP Static IP Table page opens.

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

Figure 14-4: DHCP Static IP Table - Add Record

4. Configure a static IP address for a specific DHCP client according to the parameters described in the table below.
5. Click **Submit**.

Table 14-5: DHCP Static IP Table Parameter Descriptions

Parameter	Description
Web: Index CLI: dhcp static-ip <index> [DhcpStaticIP_Index]	Defines an index number for the new table record. Note: Each table row must be configured with a unique index.
Web: DHCP Server Index CLI: dhcp-server-number [DhcpStaticIP_DhcpServerIndex]	Associates the DHCP Static IP table entry with a DHCP server that you configured in "Configuring the DHCP Server" on page 157. Note: Currently, only one DHCP server (Index 0) can be configured and therefore, this parameter is always set at 0.
Web: IP Address CLI: ip-address [DhcpStaticIP_IPAddress]	Defines the "reserved", static IP address (IPv4) to assign the DHCP client. The default is 0.0.0.0.
Web: MAC Address CLI: mac-address [DhcpStaticIP_MACAddress]	Defines the DHCP client by MAC address (in hexadecimal format). The valid value is a string of up to 20 characters. The format includes six groups of two hexadecimal digits, each separated by a colon. The default MAC address is 00:90:8f:00:00:00.

14.1.5 Viewing and Deleting DHCP Clients

The DHCP Clients table lets you view all currently serviced DHCP clients by the DHCP server. The table also lets you delete DHCP clients. If you delete a client, the DHCP server ends the lease of the IP address to the client and the IP address becomes available for allocation by the DHCP server to another client.

The following procedure describes how to view DHCP clients in the Web interface. You can also view this using the following CLI commands:

- To view DHCP clients:

```
# show voip dhcp clients
```

- To view DHCP clients according to IP address:

```
# show voip dhcp ip
```

- To view DHCP clients according to MAC address:

```
# show voip dhcp mac
```

- To view DHCP clients that have been blacklisted from DHCP implementation (due to duplicated IP addresses in the network, where another device is using the same IP address as the one assigned to the client):

```
# show voip dhcp black-list
```

➤ **To view or delete DHCP clients:**

1. Open the DHCP Servers page (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the DHCP Servers table, select the row of the desired DHCP server for which you want to view DHCP clients, and then click the **DHCP Clients Table** link located at the bottom of the page; the DHCP Clients Table page opens:

Figure 14-5: DHCP Clients Table

▼ DHCP Clients Table				
Action ▼				Show/Hide ☐
Index	DHCP Server Index	IP Address	MAC Address	Lease Expiration
0	0	192.168.0.100	00:90:8f:28:3d:e9	Mon Apr 5 16:47:00 2010
1	0	193.168.0.100	cc:c3:ea:d1:aa:a6	Mon Apr 5 22:18:10 2010
2	0	194.168.0.100	00:90:8f:1e:d2:7e	Mon Apr 5 21:59:26 2010
3	0	195.168.0.100	00:15:60:58:25:ab	Mon Apr 5 17:56:46 2010
4	0	196.168.0.100	00:24:7e:0a:4c:52	Mon Apr 5 18:39:32 2010
⏪ ⏩ Page 1 of 2 ⏪ ⏩ Show 10 records per page View 1 - 10 of 13				

The table displays the following per client:

- **Index:** Table index number.
 - **DHCP Server Index:** The index number of the configured DHCP server scope in the DHCP Server table (see "Configuring the DHCP Server" on page 157) with which the client is associated.
 - **IP Address:** IP address assigned to the DHCP client by the DHCP server.
 - **MAC Address:** MAC address of the DHCP client.
 - **Lease Expiration:** Date on which the lease of the DHCP client's IP address obtained from the DHCP server expires.
3. To delete a client:
 - a. Select the table row index of the DHCP client that you want to delete.
 - b. Click the **Action** button, and then from the drop-down menu, choose **Delete**; a confirmation message appears.
 - c. Click **OK** to confirm deletion.

14.2 SIP-based Media Recording

The device can record 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).



Warning for Deployments in France: The device supports SIP-based Media Recording (SIPREC) according to RFC 6341. As such, you must adhere to the Commission Nationale Informatique et Liberté's (CNIL) directive (<https://www.cnil.fr/en/rights-and-obligations>) and be aware that article R226-15 applies penalties to the malicious interception, diversion, use or disclosure of correspondence sent, transmitted or received by means of telecommunication, or the setting up of a device designed to produce such interceptions.

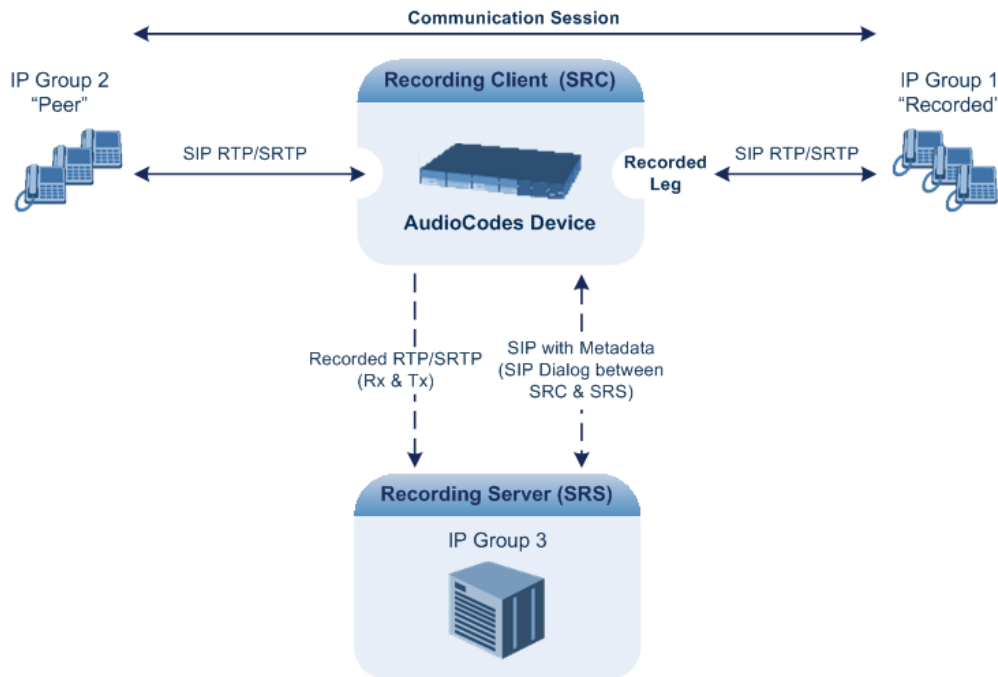


Notes:

- 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 406. 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. Specifying the leg is important as it determines the various call media attributes of the recorded RTP (or SRTP), for example, coder type.

The device can also record SRTP calls and send it to the SRS in SRTP. In such scenarios, the SRTP is used on one of the IP legs. For an SBC RTP-SRTP session, the recorded IP Group in the SIP Recording Routing table must be set to the RTP leg if recording is required to be RTP, or set to the SRTP leg if recording is required to be SRTP.

For SBC calls, the device can also be located between an SRS and an SRC and act as an RTP-SRTP translator. In such a setup, the device receives SIP recording sessions (as a server) from the SRC and translates SRTP media to RTP, or vice versa, and then forwards the recording to the SRS in the translated media format.

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/SRTP 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/SRTP streams in hex - bits 0-31 are the same as group, bits 32-47 are the RTP/SRTP 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/SRTP 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 to the SRS with the updated RTP/SRTP data. If the recorded leg uses SRTP, the device can send the media streams to the SRS as SRTP; otherwise, the media streams are sent as RTP to the SRS.

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,SUB
SCRIBE,UPDATE
Require: siprec
User-Agent: Mediant /v.6.80A.014
Content-Type: multipart/mixed;boundary=boundary_ac1ffffff85b
Content-Length: 1832

--boundary_ac1ffffff85b
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_ac1ffffff85b
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_ac1ffffff85b-

```

14.2.1 Enabling SIP-based Media Recording

The following procedure 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 171) 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.

14.2.2 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 167.

The following procedure 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 14-6: SIP Recording Routing Table - Add Record

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 14-6: 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. Note: For an SBC RTP-SRTP session, the recorded IP Group must be set to the RTP leg if recording is required to be RTP, or set to the SRTP leg if recording is required to be SRTP.
Recorded Source Prefix CLI: recorded-src-prefix [SIPRecRouting_RecordedSourcePrefix]	Defines calls to record based on source number or URI.

Parameter	Description
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).

14.2.3 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.

14.2.4 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.

14.2.4.1 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">4BOKLLA3VH66JF112M1CC9VHKS1  
4F0KP</ac:GenesysUUID>
```

No configuration is required for this support.

14.2.4.2 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 equipment, 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.

14.3 RADIUS Authentication

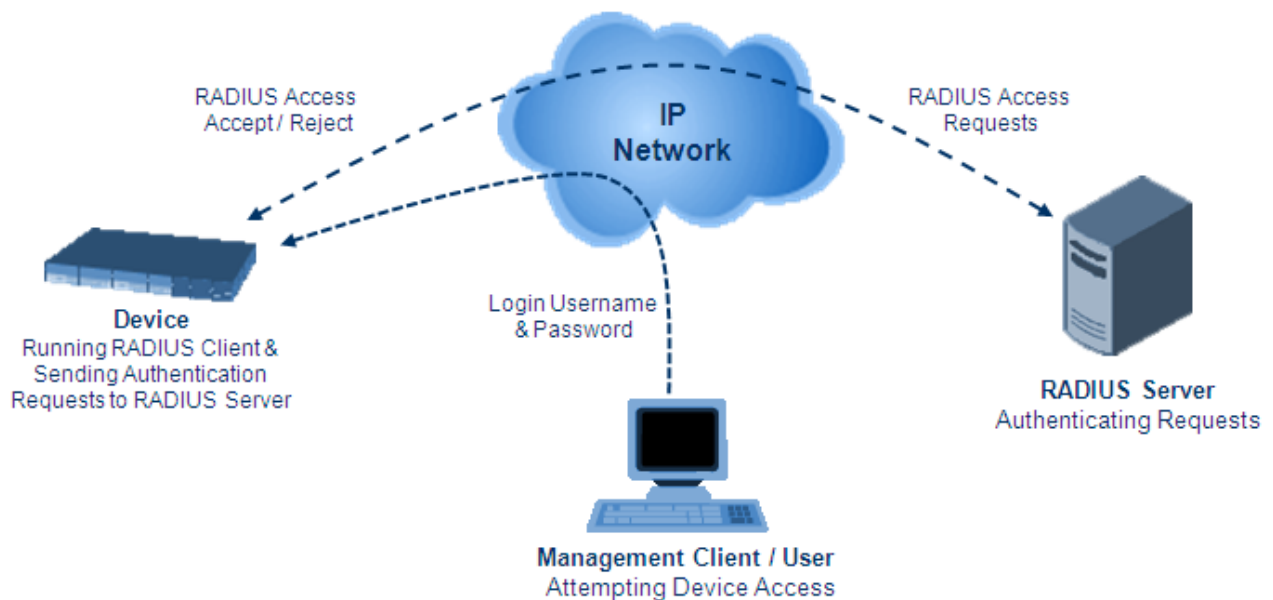
You can enhance security for your device by implementing Remote Authentication Dial-In User Service (RADIUS - RFC 2865) for authenticating multiple management user accounts of the device's embedded Web and Telnet (CLI) servers. Thus, RADIUS also prevents unauthorized access to your device.

When RADIUS authentication is not used, the user's login username and password are locally authenticated by the device in its Web Users table (database). However, the Web Users table can be used as a fallback mechanism in case the RADIUS server does not respond. For configuring local user accounts, see [Configuring Web User Accounts](#).

When RADIUS authentication is used, the RADIUS server stores the user accounts - usernames, passwords, and access levels (authorization). When a management user (client) tries to access the device, the device sends the RADIUS server the user'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.

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 7: 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 [176](#)
- Configure the device as a RADIUS client for communication with the RADIUS server - see ['Configuring RADIUS Authentication'](#) on page [177](#)

14.3.1 Setting Up a Third-Party RADIUS Server

The following procedure 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.

```
#
# 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.

14.3.2 Configuring RADIUS Authentication

The following procedure describes how to configure the RADIUS feature. For a detailed description of the RADIUS parameters, see 'RADIUS Parameters' on page 552.

➤ **To configure RADIUS:**

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

Figure 8: Authentication Settings Page - RADIUS Configuration

General Login Authentication Settings	
Use Local Users Database	When No Auth Server Defined ▾
Behavior upon Authentication Server Timeout	Verify Access Locally ▾
Password Local Cache Mode	Reset Timer Upon Access ▾
Password Local Cache Timeout (sec)	300
Default Access Level	200
LDAP settings	
Use LDAP for Web/Telnet Login	Disable ▾
RADIUS Settings	
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	••••••••
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 used to authenticate the device to the RADIUS server.
5. In the 'RADIUS VSA Vendor ID' field, enter the same vendor ID number as set on 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 176.
 - **If the RADIUS server response does not include 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. Configure RADIUS timeout handling:
 - a. From the 'Behavior upon Authentication Server Timeout' drop-down list, select the option if the RADIUS server does not respond within five seconds:
 - ♦ **Deny Access:** device denies user login access.
 - ♦ **Verify Access Locally:** device checks the username and password configured locally for the user (in the Web User Accounts page or Web Users table), and if correct, allows access.
 - b. In the 'Password Local Cache Timeout' field, enter a time limit (in seconds) after which the username and password verified by the RADIUS server becomes invalid and a username and password needs to be re-validated with the RADIUS server.
 - c. From the 'Password Local 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. Configure when the Web Users table must be used to authenticate login users. From the 'Use Local Users Database' drop-down list, select one of the following:
 - **When No Auth Server Defined (default):** When no RADIUS server is configured (or as fallback if the server is inaccessible).
 - **Always:** Always, but if not found, use the RADIUS server to authenticate the user.
9. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

14.3.3 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**).

14.3.4 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.

14.4 LDAP-based Management and SIP Services

The device supports the Lightweight Directory Access Protocol (LDAP) application protocol and can operate with third-party, LDAP-compliant servers such as Microsoft Active Directory (AD).

You can use LDAP for the following LDAP services:

- **SIP-related (Control) LDAP Queries:** This can be used for routing or manipulation (e.g., calling name and destination address). The device connects and binds to the remote LDAP server (IP address or DNS/FQDN) during the service's initialization (at device start-up) or whenever you change the LDAP server's IP address and port. Binding to the LDAP server is based on username and password (Bind DN and Password). Service makes 10 attempts to connect and bind to the remote LDAP server, with a timeout of 20 seconds between attempts. If connection fails, the service remains in disconnected state until the LDAP server's IP address or port is changed. If connection to the LDAP server later fails, the service attempts to reconnect.

For the device to run a search, the path to the directory's subtree, known as the distinguished name (DN), where the search is to be done must be configured (see 'Configuring LDAP DNs (Base Paths) per LDAP Server' on page 184). The search key (filter), which defines the exact DN to search, and one or more attributes whose values must be returned to the device must also be configured. For more information on configuring these attributes and search filters, see 'Active Directory-based Routing for Microsoft Lync' on page 197.

The device can store recent LDAP queries and responses in its local cache. The cache is used for subsequent queries and/or in case of LDAP server failure. For more information, see 'Configuring the Device's LDAP Cache' on page 189.

If connection with the LDAP server disconnects (broken), the device sends the SNMP alarm, `acLDAPLostConnection`. Upon successful reconnection, the alarm clears. If connection with 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.

- **Management-related LDAP Queries:** This is used for authenticating and authorizing management users (Web and CLI) and is based on the user's login username and password (credentials) when attempting login to one of the device's management platforms. When configuring the login username (LDAP Bind DN) and password (LDAP Password) to send to the LDAP server, you can use templates based on the dollar (\$) sign, which the device replaces with the actual username and password entered by the user during the login attempt. You can also configure the device to send the username and password in clear-text format or encrypted using TLS (SSL).

The device connects to the LDAP server (i.e., an LDAP session is created) only when a login attempt occurs. The LDAP Bind operation establishes the authentication of the user based on the username-password combination. The server typically checks the password against the `userPassword` attribute in the named entry. A successful Bind operation indicates that the username-password combination is correct; a failed Bind operation indicates that the username-password combination is incorrect.

Once the user is successfully authenticated, the established LDAP session may be used for further LDAP queries to determine the user's management access level and privileges (Operator, Admin, or Security Admin). This is known as the user authorization stage. To determine the access level, the device searches the LDAP directory for groups of which the user is a member, for example:

```
CN=# Support Dept,OU=R&D
Groups,OU=Groups,OU=APC,OU=Japan,OU=ABC,DC=corp,DC=abc,DC=com
CN=#AllCellular,OU=Groups,OU=APC,OU=Japan,OU=ABC,DC=corp,DC=a
bc,DC=com
```

The device then assigns the user the access level configured for that group (in 'Configuring Access Level per Management Groups Attributes' on page 186). The location in the directory where you want to search for the user's member group(s) is configured using the following:

- Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"), which defines the location in the directory from where the LDAP search begins, and is configured in 'Configuring LDAP DN's (Base Paths) per LDAP Server' on page 184.
- Search filter, for example, (&(objectClass=person)(sAMAccountName=JohnD)), which filters the search in the subtree to include only the specific username. The search filter can be configured with the dollar (\$) sign to represent the username, for example, (sAMAccountName=\$). For configuring the search filter, see 'Configuring the LDAP Search Filter Attribute' on page 185.
- Management attribute (e.g., memberOf), from where objects that match the search filter criteria are returned. This shows the user's member groups. The attribute is configured in the LDAP Configuration table (see 'Configuring LDAP Servers' on page 181).

If the device finds a group, it assigns the user the corresponding access level and permits login; otherwise, login is denied. Once the LDAP response has been received (success or failure), the device ends the LDAP session.

For both of the previously discussed LDAP services, the following additional LDAP functionality is supported:

- Search method for searching DN object records between LDAP servers and within each LDAP server (see 'Configuring LDAP Search Methods' on page 188).
- Default access level that is assigned to the user if the queried response does not contain an access level.
- Local users database (Web Users table) for authenticating users instead of the LDAP server (for example, when a communication problem occurs with the server). For more information, see 'Configuring Local Database for Management User Authentication' on page 191.

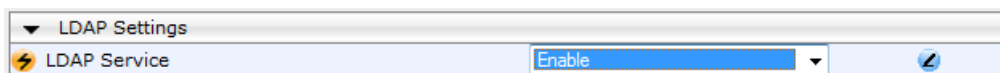
14.4.1 Enabling the LDAP Service

Before you can configure LDAP support, 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 9: Enabling LDAP on the LDAP Settings Page



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.

14.4.2 Enabling LDAP-based Web/CLI User Login Authentication and Authorization

The LDAP service can be used for authenticating and authorizing device management users (Web and CLI), based on the user's login username and password (credentials). At the same, it can also be used to determine users' management access levels (privileges). Before you can configure LDAP-based login authentication, you must enable this type of LDAP service, as described in the following procedure.

➤ **To enable LDAP-based login authentication:**

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

Figure 10: Authentication Settings Page - Enabling LDAP-based Login

The screenshot shows a web interface for 'LDAP settings'. There is a dropdown menu labeled 'Use LDAP for Web/Telnet Login' with a lightning bolt icon on the left. The dropdown is currently set to 'Enable'.

2. Under LDAP Settings, from the 'Use LDAP for Web/Telnet Login' drop-down list, select **Enable**.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

14.4.3 Configuring LDAP Servers

The LDAP Configuration table lets you configure up to four LDAP servers. This table defines the address and connectivity settings of the LDAP server. The LDAP server can be configured for SIP-related queries (e.g., routing and manipulation) or LDAP-based management user login authentication and authorization (username-password).

The following procedure describes how to configure an LDAP server in the Web interface. You can also configure this using the table ini file parameter, LdapConfiguration or CLI command, configure voip/ldap/ldap-configuration.

➤ **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 11: LDAP Configuration Table - Add Record

The screenshot shows a 'Add Record' dialog box with the following fields and values:

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 Network Interface	Control Interface
Connection Status	
Type	Control
Use TLS	No
Management Attribute	

At the bottom right, there are 'Submit' and 'Cancel' buttons.

3. Configure an LDAP server according to the parameters described in the table below.
4. Click **Submit**.

Table 14-7: 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. Note: If you want to use an FQDN for the LDAP server, leave this parameter undefined and configure the FQDN in the 'LDAP Server Domain Name' parameter (see below).
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 duration (in msec) that the device waits for LDAP server responses. The valid value range is 0 to 86400. The default is 3000. Note: If the response time expires, you can configure the device to use its local database (Web Users table) for authenticating the user. For more information, see 'Configuring Local Database for Management User Authentication' on page 191.
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, leave the 'LDAP Server IP' parameter undefined.
LDAP Password CLI: password [LdapConfiguration_LdapConfPassword]	Defines the user password for accessing the LDAP server during connection and binding operations. - LDAP-based SIP queries: The parameter is the password used by the device to authenticate itself, as a client, to obtain LDAP service from the LDAP server. - LDAP-based user login authentication: The parameter represents the login password entered by the user during a login attempt. You can use the \$ (dollar) sign in this value to enable the device to automatically replace the \$ sign with the user's login password in the search filter, which it sends to the LDAP server for authenticating the user's username-password combination. For example, \$. Note: By default, the device sends the password in clear-text format. You can enable the device to encrypt the password using TLS (see the 'Use SSL' parameter below).

Parameter	Description
LDAP Bind DN CLI: bind-dn [LdapConfiguration_LdapConfBindDn]	Defines the LDAP server's bind Distinguished Name (DN) or username. - LDAP-based SIP queries: The DN is used as the username during connection and binding to the LDAP server. The DN is used to uniquely name an AD object. Below are example parameter settings: ✓ cn=administrator,cn=Users,dc=domain,dc=com ✓ administrator@domain.com ✓ domain\administrator - LDAP-based user login authentication: This parameter represents the login username entered by the user during a login attempt. You can use the \$ (dollar) sign in this value to enable the device to automatically replace the \$ sign with the user's login username in the search filter, which it sends to the LDAP server for authenticating the user's username-password combination. An example configuration for this parameter is \$@sales.local, where the device replaces the \$ with the entered username, for example, JohnD@sales.local. The username can also be configured with the domain name of the LDAP server. Note: By default, the device sends the username in clear-text format. You can enable the device to encrypt the username using TLS (see the 'Use SSL' parameter below).
LDAP Network Interface CLI: interface-type [LdapConfiguration_LdapConfInterfaceType]	Assigns one of the device's IP network interfaces for communicating with the LDAP server. [0] Control Interface (default) = The top-most IP network interface row in the IP Interfaces table that is configured for a Control application (may be combined with other applications such as OAMP and Media) is used. [1] OAM Interface = The OAMP interface (may be combined with other applications such as Control and Media) in the IP Interfaces table is used. For configuring IP network interfaces, see Configuring IP Network Interfaces.
Type CLI: type [LdapConfiguration_Type]	Defines whether the LDAP server is used for SIP-related queries or management login authentication-related queries. [0] Control (Default) [1] Management Note: If you use the same LDAP server for both management and SIP (Control) related applications, the device establishes different LDAP sessions for each application.
Management Attribute CLI: mgmt-attr [LdapConfiguration_MngmAuthAtt]	Defines the LDAP attribute name to query, which contains a list of groups to which the user is a member. For Active Directory, this attribute is typically "memberOf". The attribute's values (groups) are used to determine the user's management access level; the group's corresponding access level is configured in 'Configuring Access Level per Management Groups Attributes' on page 186. Notes: - This parameter is applicable only to LDAP-based login authentication and authorization (i.e., the 'Type' parameter is set to Management). - If this functionality is not used, the device assigns the user the configured default access level. For more information, see 'Configuring Access Level per Management Groups Attributes' on page 186.

Parameter	Description
Use SSL CLI: [LdapConfiguration_useTLS]	Enables the device to encrypt the username and password (for Control and Management related queries) using TLS when sending them to the LDAP server. [0] No = (Default) Username and password are sent in clear-text format. [1] Yes
Connection Status CLI: connection-status [LdapConfiguration_ConnectionStatus]	(Read-only) Displays the connection status with the LDAP server. "Not Applicable" "LDAP Connection Broken" "Connecting" "Connected" Note: For more information about a disconnected LDAP connection, see your Syslog messages generated by the device.

14.4.4 Configuring LDAP DN (Base Paths) per LDAP Server

The LDAP Search DN Table lets you configure LDAP base paths. The table is a "child" of the LDAP Configuration table (see 'Configuring LDAP Servers' on page 181) and configuration is done per LDAP server. For the device to run a search using the LDAP service, the base path to the directory's subtree, referred to as the distinguished name object (or DN), where the search is to be done must be configured. For each LDAP server, you can configure up to three base paths.

The following procedure describes how to configure DN per LDAP server in the Web interface. You can also configure this using the table ini file parameter, LdapServersSearchDNs or CLI command, configure voip/ldap/ldap-servers-search-dns.

➤ **To configure an LDAP base path per LDAP server:**

1. Open the LDAP Configuration Table page (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Configuration Table**).
2. In the LDAP Configuration table, select the row of the LDAP server for which you want to configure DN base paths, and then click the **Search DN** link (located at the bottom of the page); the LDAP Search DN Table page opens.
3. Click **Add**; the following dialog box appears:

Figure 12: LDAP Search DN Table - Add Record

4. Configure an LDAP DN base path according to the parameters described in the table below.
5. Click **Submit**, and then save ("burn") your settings to flash memory.

LDAP Search DN Table Parameter Descriptions

Parameter	Description
Index CLI: set internal-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 CLI: set base-path [LdapServersSearchDNs_Base_Path]	Defines the full path (DN) to the objects in the AD where the query is done. The valid value is a string of up to 256 characters. 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).

14.4.5 Configuring the LDAP Search Filter Attribute

When the LDAP-based login username-password authentication succeeds, the device searches the LDAP server for all groups of which the user is a member. The LDAP query is based on the following LDAP data structure:

- **Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"):** The DN defines the location in the directory from which the LDAP search begins and is configured in 'Configuring LDAP DNs (Base Paths) per LDAP Server' on page 184.
- **Filter (e.g., "(&(objectClass=person)(sAMAccountName=johnd))"):** This filters the search in the subtree to include only the login username (and excludes others). This is configured by the 'LDAP Authentication Filter' parameter, as described in the following procedure. You can use the dollar (\$) sign to represent the username. For example, the filter can be configured as "(sAMAccountName=\$)", where if the user attempts to log in with the username "SueM", the LDAP search is done only for the attribute sAMAccountName that equals "SueM".
- **Attribute (e.g., "memberOf") to return from objects that match the filter criteria:** The attribute is configured by the 'Management Attribute' parameter in the LDAP Configuration table (see 'Configuring LDAP Servers' on page 181).

Therefore, the LDAP response includes only the groups of which the specific user is a member.



Notes:

- The search filter is applicable only to LDAP-based login authentication and authorization queries.
- The search filter is a global setting that applies to all LDAP-based login authentication and authorization queries, across all configured LDAP servers.

➤ To configure the LDAP search filter for management users:

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

Figure 13: LDAP Settings Page - LDAP Search Filter

LDAP Settings	
LDAP Service	Enable
LDAP Authentication Filter	(sAMAccountName=)

2. Under LDAP Settings, in the 'LDAP Authentication Filter' parameter, enter the LDAP search filter attribute for searching the login username for user authentication.
3. Click **Submit**.

14.4.6 Configuring Access Level per Management Groups Attributes

The Management LDAP Groups table lets you configure LDAP group objects and their corresponding management user access level. The table is a "child" of the LDAP Configuration table (see 'Configuring LDAP Servers' on page 181) and configuration is done per LDAP server. For each LDAP server, you can configure up to three table row entries of LDAP group(s) and their corresponding access level.



Notes:

- The Management LDAP Groups table is applicable only to LDAP-based login authentication and authorization queries.
- If the LDAP response received by the device includes multiple groups of which the user is a member and you have configured different access levels for some of these groups, the device assigns the user the highest access level. For example, if the user is a member of two groups where one has access level "Monitor" and the other "Administrator", the device assigns the user the "Administrator" access level.
- When the access level is unknown, the device assigns the default access level to the user, configured by the 'Default Access Level' parameter in the Authentication Settings page (**Configuration** tab > **System** menu > **Management** > **Authentication Settings**). This can occur in the following scenarios:
 - ✓ The user is not a member of any group.
 - ✓ The group of which the user is a member is not configured on the device (as described in this section).
 - ✓ The device is not configured to query the LDAP server for a management attribute (see 'Configuring LDAP Servers' on page 181).

Group objects represent groups in the LDAP server of which the user is a member. The access level represents the user account's permissions and rights in the device's management interface (e.g., Web and CLI). The access level can either be Monitor, Administrator, or Security Administrator. For an explanation on the privileges of each level, see Configuring Web User Accounts.

When the username-password authentication with the LDAP server succeeds, the device searches the LDAP server for all groups of which the user is a member. The LDAP query is based on the following LDAP data structure:

- Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"), which defines the location in the directory from which the LDAP search begins. This is configured in 'Configuring LDAP DN's (Base Paths) per LDAP Server' on page 184.

- Filter (e.g., "(&(objectClass=person)(sAMAccountName=johnd))"), which filters the search in the subtree to include only the login username (and excludes others). This is configured by the 'LDAP Authentication Filter' parameter.
- Attribute (e.g., "memberOf") to return from objects that match the filter criteria. This attribute is configured by the 'Management Attribute' parameter in the LDAP Configuration table.

The LDAP response includes all the groups of which the specific user is a member, for example:

```
CN=\# Support Dept,OU=R&D
Groups,OU=Groups,OU=APC,OU=Japan,OU=ABC,DC=corp,DC=abc,DC=com
CN=\#AllCellular,OU=Groups,OU=APC,OU=Japan,OU=ABC,DC=corp,DC=abc,DC=com
```

The device searches this LDAP response for the group names that you configured in the Management LDAP Groups table in order to determine the user's access level. If the device finds a group name, the user is assigned the corresponding access level and login is permitted; otherwise, login is denied. Once the LDAP response has been received (success or failure), the LDAP session terminates.

The following procedure describes how to configure an access level per management groups in the Web interface. You can also configure this using the table ini file parameter, MgmntLDAPGroups or CLI command, configure voip > ldap > mgmt-ldap-groups.

➤ **To configure management groups and corresponding access level:**

1. Open the LDAP Configuration Table page (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Configuration Table**).
2. In the LDAP Configuration table, select the row of the LDAP server for which you want to configure management groups with a corresponding access level, and then click the **Management LDAP Groups Table** link (located at the bottom of the page); the Management LDAP Groups Table page opens.
3. Click **Add**; the following dialog box appears:

Figure 14: Management LDAP Groups Table - Add Record

4. Configure a group name(s) with a corresponding access level according to the parameters described in the table below.
5. Click **Submit**, and then save ("burn") your settings to flash memory.

Management LDAP Groups Table Parameter Descriptions

Parameter	Description
Index [MgmntLDAPGroups_GroupIndex]	Defines an index number for the new table record. Note: Each table row must be configured with a unique index.
Level [MgmntLDAPGroups_Level]	Defines the access level of the group(s). ■ [0] Operator (Default) ■ [1] Admin ■ [2] Security Admin

Parameter	Description
Groups [MgmtLDAPGroups_ Group]	Defines the attribute names of the groups in the LDAP server. The valid value is a string of up to 256 characters. To define multiple groups, separate each group name with a semicolon (;).

14.4.7 Configuring LDAP Search Methods

You can configure the device's method for searching the LDAP server(s) for the configured DN objects:

- **DN Search Method between Two LDAP Servers:** When two LDAP servers are implemented, the device runs an LDAP query to search for DN object records on both LDAP servers. You can configure how the device queries the DN object record between the two LDAP servers:
 - **Parallel Search:** The device queries the LDAP servers simultaneously.
 - **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.
- **DN Search Method within an LDAP Server:** You can configure how the device queries the DN object record within each LDAP server:
 - **Parallel Search:** The device queries all DN objects simultaneously. 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 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 searches in "Administration", and so on.

➤ To configure LDAP search methods:

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

Figure 15: LDAP Settings Page - Search Methods

LDAP Search Server Method	LDAP_SEARCH_IN_PARALLE
search dns in parallel	Enable

2. Under LDAP Settings, configure the following:
 - Search method for DN objects between two LDAP servers, using the 'LDAP Search Server Method' parameter (LDAPSearchServerMethod).
 - Search method for DN objects within an LDAP server, using the 'search dns in parallel' parameter (LdapSearchDnsInParallel).
3. Click **Submit**.

14.4.8 Configuring the Device's LDAP Cache

The device can optionally store 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.



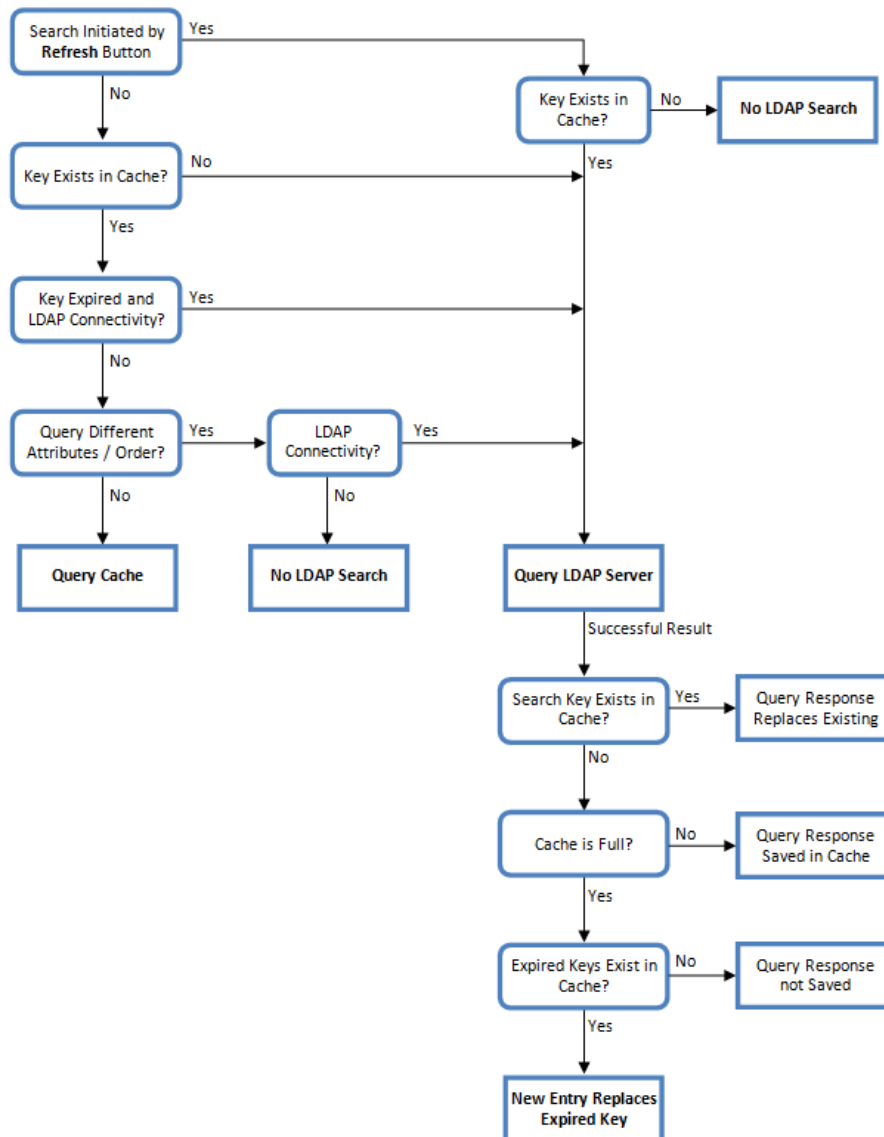
Note: The LDAP Cache feature is applicable only to LDAP-based SIP queries (Control).

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 16: LDAP Query Process with Local LDAP Cache



Note: If for 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 following procedure 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 606.

➤ **To configure the LDAP cache:**

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

Figure 17: 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. Under LDAP Cache, do the following:
 - a. From the 'LDAP Cache Service' drop-down list, select **Enable** to enable LDAP cache.
 - b. In the 'LDAP Cache Entry Timeout' field, enter the duration (in minutes) for which an entry in the LDAP cache is valid.
 - c. In the 'LDAP Cache Entry Removal Timeout' field, enter the duration (in hours) after which the device removes the LDAP entry from the cache.
3. 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.

14.4.9 Configuring Local Database for Management User Authentication

You can configure the device to use its local database (Web Users table) to authenticate management users based on the username-password combination. You can configure the device to use the Web Users table upon the following scenarios:

- LDAP or RADIUS server is not configured (or broken connection), or always use the Web Users table and only if the user is not found, to use the server.
- Connection with the LDAP or RADIUS server fails due to a timeout. In such a scenario, the device can deny access or verify the user's credentials (username-password) locally in the Web Users table.

If user authentication using the Web Users table succeeds, the device grants management access to the user; otherwise access is denied. The access level assigned to the user is also determined by the Web Users table. To configure local Web/CLI users in the Web Users table, see Configuring Web User Accounts.



Notes:

- This feature is applicable to LDAP and RADIUS servers.
- This feature is applicable only to user management authentication.

➤ To use the Web Users table for authenticating management users:

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

Figure 18: Authentication Settings Page - Local Database for Login Authentication

General Login Authentication Settings	
Use Local Users Database	Always
Behavior upon Authentication Server Timeout	Verify Access Locally

2. Under General Login Authentication Settings:
 - Configure when the Web Users table must be used to authenticate login users. From the 'Use Local Users Database' drop-down list, select one of the following:
 - ◆ **When No Auth Server Defined (default):** When no LDAP/RADIUS server is configured (or as fallback if the server is inaccessible).
 - ◆ **Always:** Always, but if not found, use the LDAP/RADIUS server to authenticate the user.
 - Configure whether the Web Users table must be used to authenticate login users upon connection timeout with the server. From the 'Behavior upon Authentication Server Timeout' drop-down list, select one of the following:
 - ◆ **Deny Access:** User is denied access to the management platform.
 - ◆ **Verify Access Locally (default):** The device verifies the user's credentials in the Web Users table.
3. Click **Submit**.

14.4.10 LDAP-based Login Authentication Example

To facilitate your understanding on LDAP entry data structure and how to configure the device to use and obtain information from this LDAP directory, a brief configuration example is described in this section. The example applies to LDAP-based user login authentication and authorization (access level), and assumes that you are familiar with other aspects of LDAP configuration (e.g., LDAP server's address).

The LDAP server's entry data structure schema in the example is as follows:

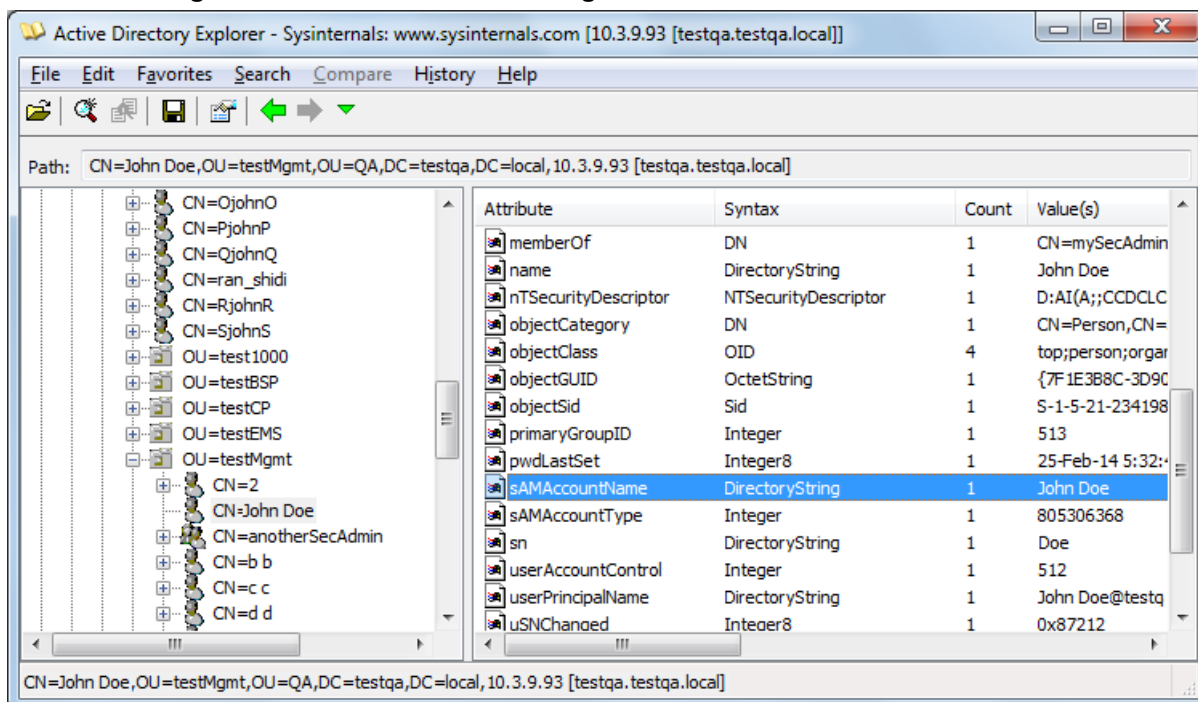
- **DN (base path):** OU=testMgmt,OU=QA,DC=testqa,DC=local. The DN path to search for the username in the directory is shown below:

Figure 19: Base Path (DN) in LDAP Server

Attribute	Syntax	Count	Value(s)
accountExpires	Integer8	1	0x7FFFFFFFFFFFFFFF
badPasswordTime	Integer8	1	06-Mar-14 10:03:18 AM
badPwdCount	Integer	1	0
cn	DirectoryString	1	John Doe
codePage	Integer	1	0
countryCode	Integer	1	0
description	DirectoryString	1	10600
displayName	DirectoryString	1	John Doe
distinguishedName	DN	1	CN=John Doe,OU=testMgm
givenName	DirectoryString	1	John
instanceType	Integer	1	4
lastLogoff	Integer8	1	0x0
lastLogon	Integer8	1	06-Mar-14 10:03:41 AM
logonCount	Integer	1	0
memberOf	DN	1	CN=mySecAdmin,OU=testM
name	DirectoryString	1	John Doe
nTSecurityDescriptor	NTSecurityDescriptor	1	D:AI(A;;CCDCLCSWRPWPDT
objectCategory	DN	1	CN=Person,CN=Schema,CN
objectClass	OID	4	top;person;organizationalPe
objectGUID	OctetString	1	{7F1E3B8C-3D90-47BC-A9E
objectSid	Sid	1	S-1-5-21-2341986137-2970
primaryGroupID	Integer	1	513
pwdLastSet	Integer8	1	25-Feb-14 5:32:45 PM
sAMAccountName	DirectoryString	1	John Doe
sAMAccountType	Integer	1	805306368
sn	DirectoryString	1	Doe
userAccountControl	Integer	1	512
userPrincipalName	DirectoryString	1	John.Doe@testqa.local
uSNCreated	Integer8	1	0x87212
uSNChanged	Integer8	1	0x8311F
whenChanged	GeneralizedTime	1	25-Feb-14 5:32:45 PM
whenCreated	GeneralizedTime	1	06-Oct-02 5:27:51 AM

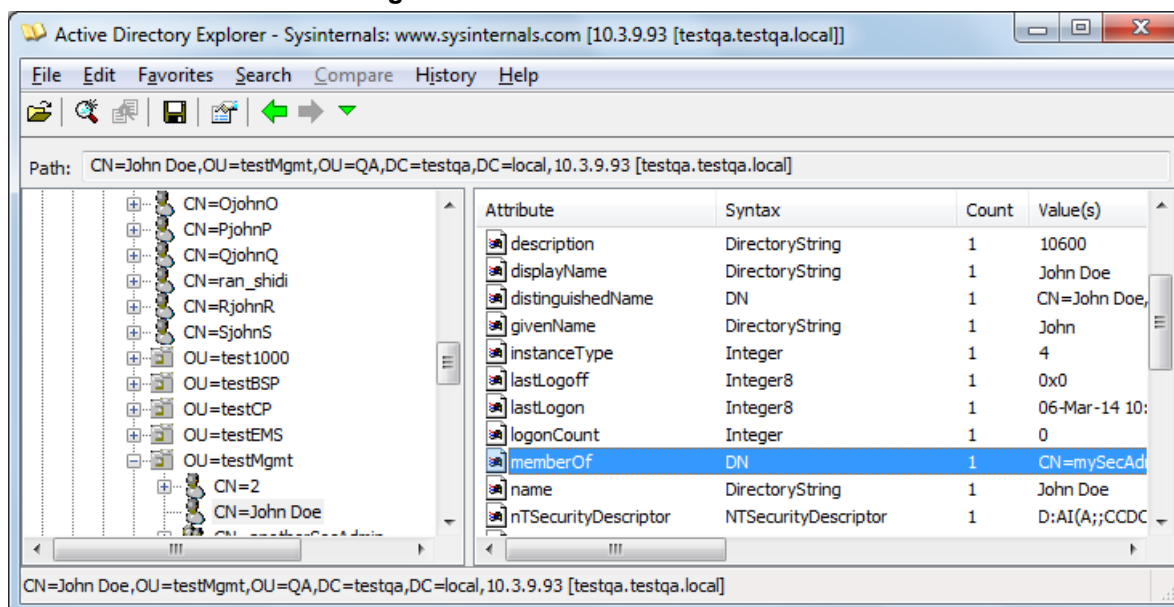
- **Search Attribute Filter:** (sAMAccountName=\$). The login username is found based on this attribute (where the attribute's value equals the username):

Figure 20: Username Found using sAMAccount Attribute Search Filter



- **Management Attribute:** memberOf. The attribute contains the member groups of the user:

Figure 21: User's memberof Attribute



- **Management Group:** mySecAdmin. The group to which the user belongs, as listed under the memberOf attribute:

Figure 22: User's mySecAdmin Group in memberOf Management Attribute

The configuration to match the above LDAP data structure schema is as follows:

- The DN is configured in the LDAP Configuration table (see 'Configuring LDAP Servers' on page 181):

Figure 23: Configuring DN

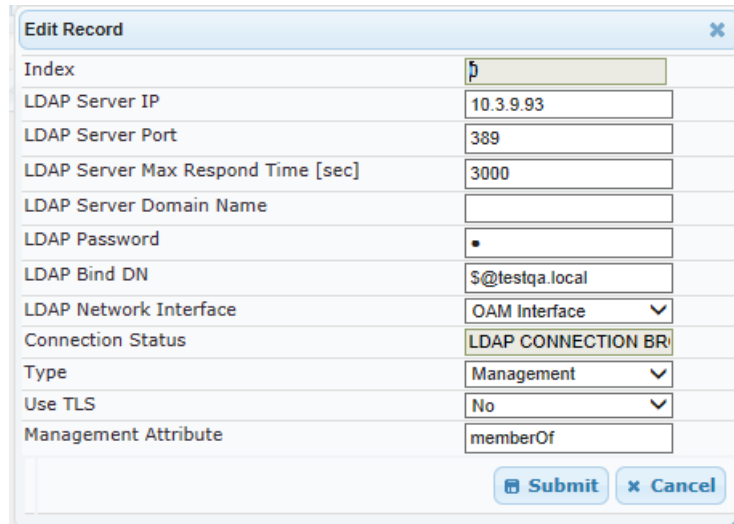
- The search attribute filter based on username is configured by the 'LDAP Authentication Filter' parameter in the LDAP Settings page (see 'Configuring the LDAP Search Filter Attribute' on page 185):

Figure 24: Configuring Search Attribute Filter

LDAP Settings		
LDAP Service	Enable	▼
LDAP Authentication Filter	(sAMAccountName=)	
LDAP Search Server Method	LDAP Search Sequentially	▼
Search DN's in Parallel	Disable	▼

- The group management attribute is configured by the 'Management Attribute' parameter in the LDAP Configuration table:

Figure 25: Configuring Management Attribute

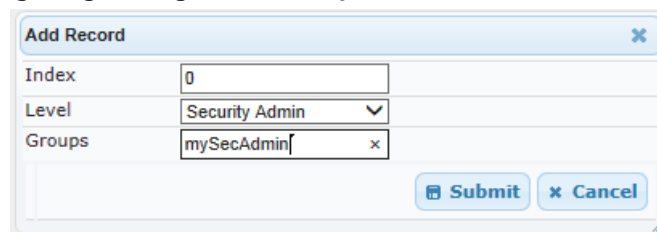


Index	5
LDAP Server IP	10.3.9.93
LDAP Server Port	389
LDAP Server Max Respond Time [sec]	3000
LDAP Server Domain Name	
LDAP Password	•
LDAP Bind DN	S@testqa.local
LDAP Network Interface	OAM Interface
Connection Status	LDAP CONNECTION BR
Type	Management
Use TLS	No
Management Attribute	memberOf

Submit Cancel

- The management group and its corresponding access level is configured in the Management LDAP Groups table (see 'Configuring Access Level per Management Groups Attributes' on page 186):

Figure 26: Configuring Management Group Attributes for Determining Access Level



Index	0
Level	Security Admin
Groups	mySecAdmin

Submit Cancel

14.4.11 Active Directory-based Routing for Microsoft Lync

Typically, enterprises wishing to deploy the Microsoft® Lync™ Server are faced with a complex, call routing dial plan when migrating users from their existing PBX or IP PBX to the Lync Server platform. As more and more end-users migrate to the new voice system, dialing plan management and PBX link capacity can be adversely impacted. To resolve this issue, enterprises can employ Microsoft's Active Directory (AD), which provides a central database to manage and maintain information regarding user's availability, presence, and location.

The device supports outbound IP call routing decisions based on information stored on the AD. Based on queries sent to the AD, the device can route the call to one of the following IP domains:

- Lync client - users connected to Lync Server through the Mediation Server
- PBX or IP PBX - users not yet migrated to Lync Server
- Mobile - mobile number
- Private - private telephone line for Lync users (in addition to the primary telephone line)

14.4.11.1 Querying the AD and Routing Priority

The device queries the AD using the initial destination number (i.e., called number). The query can return up to four user phone numbers, each pertaining to one of the IP domains (i.e., private number, Lync number, PBX / IP PBX number, and mobile number). The configuration parameters listed in the table below are used to configure the query attribute keys that defines the AD attribute that you wish to query in the AD:

Parameters for Configuring Query Attribute Key

Parameter	Queried User Domain (Attribute) in AD	Query or Query Result Example
MSLDAPPBXNumAttributeName	PBX or IP PBX number (e.g., "telephoneNumber" - default)	telephoneNumber=+3233554447
MSLDAPOCSNumAttributeName	Mediation Server / Lync client number (e.g., "msRTCSIP-line")	msRTCSIP-line=john.smith@company.com
MSLDAPMobileNumAttributeName	Mobile number (e.g., "mobile")	mobile=+3247647156
MSLDAPPrivateNumAttribute	Any attribute (e.g., "msRTCSIP-PrivateLine") Note: Used only if set to same value as Primary or Secondary key.	msRTCSIP-PrivateLine=+3233554480
MSLDAPPrimaryKey	Primary Key query search instead of PBX key - can be any AD attribute	msRTCSIP-PrivateLine=+3233554480
MSLDAPSecondaryKey	Secondary Key query key search if Primary Key fails - can be any attribute	-

The process for querying the AD and subsequent routing based on the query results is as follows:

1. If the Primary Key is configured, it uses the defined string as a primary key instead of the one defined in MSLDAPPBXNumAttributeName. It requests the attributes which are described below.
2. If the primary query is not found in the AD and the Secondary Key is configured, it does a second query for the destination number using a second AD attribute key name, configured by the MSLDAPSecondaryKey parameter.
3. If none of the queries are successful, it routes the call to the original dialed destination number according to the routing rule matching the "LDAP_ERR" destination prefix number value, or rejects the call with a SIP 404 "Not Found" response.
4. For each query (primary or secondary), it queries the following attributes (if configured):
 - MSLDAPPBXNumAttributeName
 - MSLDAPOCSNumAttributeName
 - MSLDAPMobileNumAttributeName

In addition, it queries the special attribute defined in MSLDAPPrivateNumAttributeName, only if the query key (primary or secondary) is equal to its value.
5. If the query is found: The AD returns up to four attributes - Lync, PBX / IP PBX, private (only if it equals Primary or Secondary key), and mobile.
6. The device adds unique prefix keywords to the query results in order to identify the query type (i.e., IP domain). These prefixes are used as the prefix destination number value in the Outbound IP Routing table to denote the IP domains:
 - "PRIVATE" (PRIVATE:<private_number>): used to match a routing rule based on query results of the private number (MSLDAPPrivateNumAttributeName)
 - "OCS" (OCS:<Lync_number>): used to match a routing rule based on query results of the Lync client number (MSLDAPOCSNumAttributeName)
 - "PBX" (PBX:<PBX_number>): used to match a routing rule based on query results of the PBX / IP PBX number (MSLDAPPBXNumAttributeName)
 - "MOBILE" (MOBILE:<mobile_number>): used to match a routing rule based on query results of the mobile number (MSLDAPMobileNumAttributeName)
 - "LDAP_ERR": used to match a routing rule based on a failed query result when no attribute is found in the AD



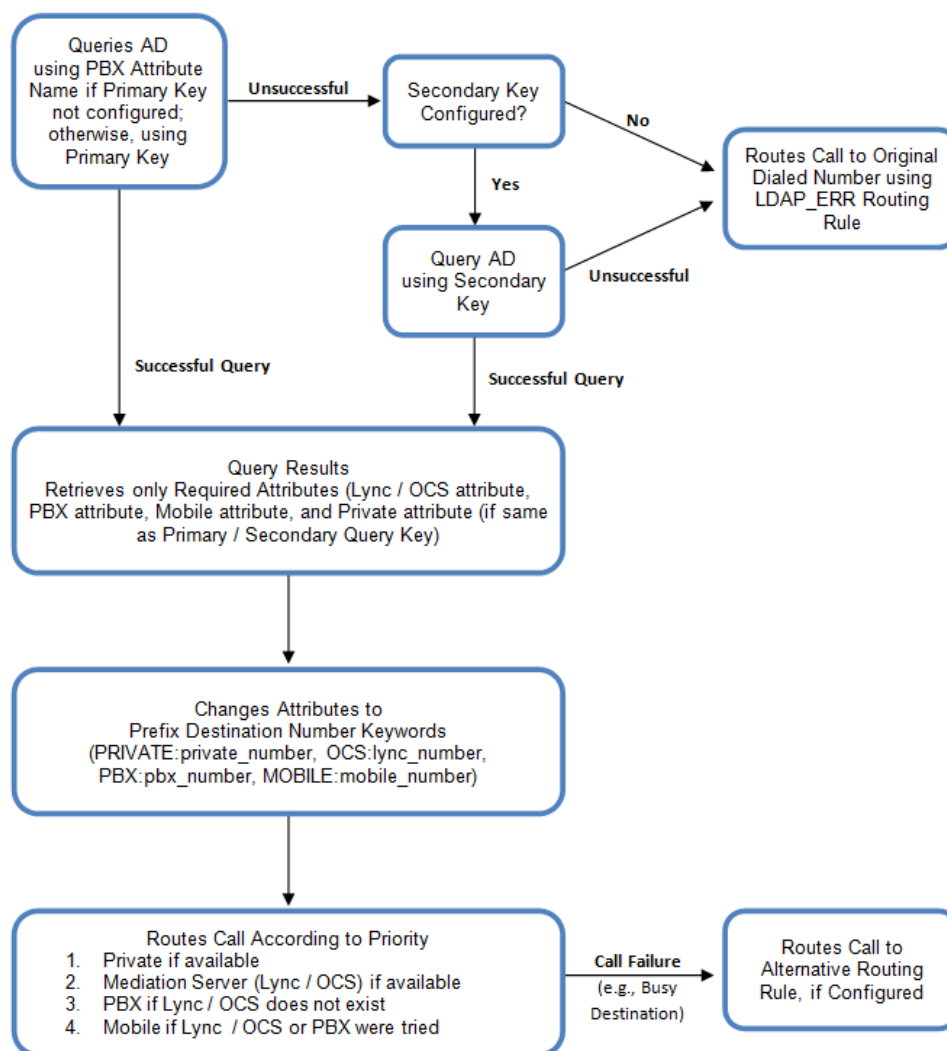
Note: These prefixes are involved only in the routing and manipulation processes; they are not used as the final destination number.

7. The device uses the Outbound IP Routing table to route the call based on the LDAP query result. The device routes the call according to the following priority:
 1. **Private line:** If the query is done for the private attribute and it's found, the device routes the call according to this attribute.
 2. **Mediation Server SIP address (Lync):** If the private attribute does not exist or is not queried, the device routes the call to the Mediation Server (which then routes the call to the Lync client).
 3. **PBX / IP PBX:** If the Lync client is not found in the AD, it routes the call to the PBX / IP PBX.

4. **Mobile number:** If the Lync client (or Mediation Server) is unavailable (e.g., SIP response 404 "Not Found" upon INVITE sent to Lync client), and the PBX / IP PBX is also unavailable, the device routes the call to the user's mobile number (if exists in the AD).
5. **Alternative route:** If the call routing to all the above fails (e.g., due to unavailable destination - call busy), the device can route the call to an alternative destination if an alternative routing rule is configured.
6. **"Redundant" route:** If the query failed (i.e., no attribute found in the AD), the device uses the routing rule matching the "LDAP_ERR" prefix destination number value.

The flowchart below summarizes the device's process for querying the AD and routing the call based on the query results:

Figure 27: LDAP Query Flowchart



Note: If you are using the device's local LDAP cache, see 'Configuring the Device's LDAP Cache' on page 189 for the LDAP query process.

14.4.11.2 Configuring AD-Based Routing Rules

The following procedure describes how to configure outbound IP routing based on LDAP queries.

➤ **To configure LDAP-based IP routing for Lync Server:**

1. Configure the LDAP server parameters, as described in 'Configuring LDAP Servers' on page 181.
2. Configure the AD attribute names used in the LDAP query:
 - a. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).

Figure 28: LDAP Parameters for Microsoft Lync Server 2010

MS LDAP Settings		
MS LDAP OCS Number Attribute Name	msRTCSIP-Line	
MS LDAP PBX Number Attribute Name	telephoneNumber	
MS LDAP MOBILE Number Attribute Name	mobile	
MS LDAP DISPLAY Name Attribute Name	displayName	
MS LDAP PRIVATE Number Attribute Name	msRTCSIP-PrivateLine	
MS LDAP Primary Key	telephoneNumber	
MS LDAP Secondary Key		

- b. Configure the LDAP attribute names as desired.
3. Configure AD-based IP-to-IP routing rules:
 - a. Open the IP-to-IP Routing Table page (Configuration tab > VoIP menu > SBC > Routing SBC > IP-to-IP Routing Table). For more information, see Configuring SBC IP-to-IP Routing Rules.
 - b. Configure query-result routing rules for each IP domain (private, PBX / IP PBX, Lync clients, and mobile), using the LDAP keywords (case-sensitive) in the Destination Username Prefix field:
 - ◆ PRIVATE: Private number
 - ◆ OCS: Lync client number
 - ◆ PBX: PBX / IP PBX number
 - ◆ MOBILE: Mobile number
 - ◆ LDAP_ERR: LDAP query failure
 - c. Configure a routing rule for routing the initial call (LDAP query) to the LDAP server, by setting the 'Destination Type' field to LDAP for denoting the IP address of the LDAP server.
 - d. For alternative routing, enable the alternative routing mechanism and configure corresponding SIP reasons for alternative routing. For this feature, alternative routing starts from the table row located under the LDAP query row.

The table below shows an example for configuring AD-based SBC routing rules in the IP-to-IP Routing Table:

AD-Based SBC IP-to-IP Routing Rule Configuration Examples

Index	Destination Username Prefix	Destination Type	Destination Address
1	PRIVATE:	Dest Address	10.33.45.60
2	PBX:	Dest Address	10.33.45.65
3	OCS:	Dest Address	10.33.45.68

Index	Destination Username Prefix	Destination Type	Destination Address
4	MOBILE:	Dest Address	10.33.45.100
5	LDAP_ERR	Dest Address	10.33.45.80
6	*	LDAP	
7	*	Dest Address	10.33.45.72

The configured routing rule example is explained below:

- **Rule 1:** Sends call to private telephone line (at 10.33.45.60) upon successful AD query result for the private attribute.
- **Rule 2:** Sends call to IP PBX (at 10.33.45.65) upon successful AD query result for the PBX attribute.
- **Rule 3:** Sends call to Lync client (i.e., Mediation Server at 10.33.45.68) upon successful AD query result for the Lync attribute.
- **Rule 4:** Sends call to user's mobile phone number (to PSTN through the device's IP address at 10.33.45.100) upon successful AD query result for the Mobile attribute.
- **Rule 5:** Sends call to IP address of device (10.33.45.80) if AD query failure (e.g., no response from LDAP server or attribute not found).
- **Rule 6:** Sends query for original destination number of received call to the LDAP server.
- **Rule 7:** Alternative routing rule that sends the call of original dialed number to IP destination 10.33.45.72. This rule is applied in any of the following cases
 - LDAP functionality is disabled.
 - LDAP query is successful but call fails (due to, for example, busy line) to all the relevant attribute destinations (private, Lync, PBX, and mobile), and a relevant SBC Alternative Routing Reason (see Configuring SIP Response Codes for Alternative Routing Reasons) has been configured.

Once the device receives the original incoming call, the first rule that it uses is Rule 6, which queries the AD server. When the AD replies, the device searches the table, from the first rule down, for the matching destination phone prefix (i.e., "PRIVATE:", "PBX:", "OCS:", "MOBILE:", and "LDAP_ERR:"), and then sends the call to the appropriate destination.

14.5 Least Cost Routing

This section provides a description of the device's least cost routing (LCR) feature and how to configure it.

14.5.1 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 14-8: 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:

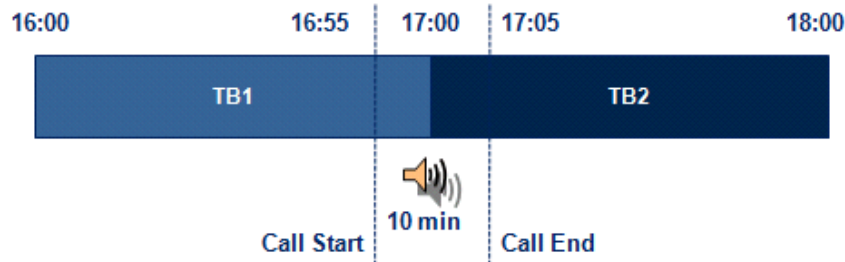
- Index 1 - Cost Group "A" has the lowest connection cost and minute cost
 - Index 2 - Cost Group "B" takes precedence over Index 4 entry based on the first-matched method rule
 - Index 3 - no Cost Group is assigned, but as the Default Cost parameter is set to **Min**, it is selected as the cheapest 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 14-29: 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

14.5.2 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 204.
2. Configure Cost Groups - see "Configuring Cost Groups" on page 205.
3. Configure Time Bands for a Cost Group - see "Configuring Time Bands for Cost Groups" on page 206.
4. Assign Cost Groups to outbound IP routing rules - see "Assigning Cost Groups to Routing Rules" on page 208.

14.5.2.1 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 following procedure 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 14-30: Routing Rule Groups Table - Add Record

The screenshot shows a dialog box titled "Add Record" with a close button (X). It contains four input fields: "Index" with the value "1", "LCR Enable" with a dropdown menu set to "Enabled", "LCR Call Length" with the value "10", and "Default Cost" with a dropdown menu set to "Lowest Cost". At the bottom right, there are two buttons: "Submit" and "Cancel".

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 14-9: 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.

14.5.2.2 Configuring Cost Groups

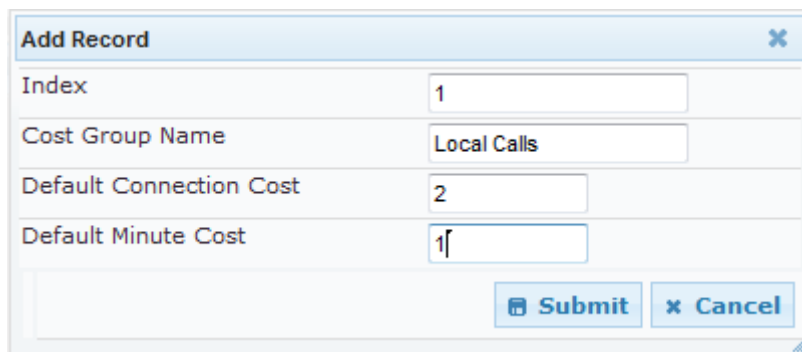
The Cost Group table lets you configure Cost Groups. A Cost Group defines a fixed call connection cost and a call rate (charge per minute). Once configured, you can configure Time Bands per Cost Group. Up to 10 Cost Groups can be configured.

The following procedure 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:



The dialog box is titled "Add Record" and contains four input fields:

- Index:** 1
- Cost Group Name:** Local Calls
- Default Connection Cost:** 2
- Default Minute Cost:** 1

At the bottom right, there are two buttons: "Submit" and "Cancel".

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 14-10: 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.

14.5.2.3 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 following procedure 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
Submit Cancel	

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 14-11: 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).
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).

14.5.2.4 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 [340](#)

14.6 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 [340](#)

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 following procedure 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 14-31: Call Setup Rules Table - Add Record

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 14-12: 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.

Parameter	Description
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 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)

Parameter	Description
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). ▪ [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] Run Rules Set = Performs a different Rule Set ID, specified in the 'Action Value' parameter (below). ▪ [21] Exit = Stops the Rule Set ID and returns a result ("True" or "False").
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)

14.6.1 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)

15 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 406.

15.1 Reporting Voice Quality of Experience to SEM

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.

SEM is a VoIP-quality monitoring and analysis tool. SEM provides comprehensive details on voice traffic quality, allowing system administrators to quickly identify, fix and prevent issues that could affect the voice calling experience in enterprise and service provider VoIP networks. IT managers and administrators can employ SEM in their VoIP networks to guarantee effective utilization, smooth performance, reliable QoS levels, and SLA fulfillment.



Note: For information on the SEM server, refer to the *EMS User's Manual*.

15.1.1 Configuring the SEM Server

The device can be configured to report QoE voice metrics to a single SEM server or to two SEM/EMS servers deployed in a Geographic Redundancy, High-Availability (HA) mode. Geographic Redundancy is when each SEM/EMS server is located in a different network subnet and has its own IP address. Thus, for the device to report QoE to both servers, you need to configure the IP address of each server.

For regular HA mode, when both EMS/SEM servers are located in the same subnet, a single EMS/SEM server (global, virtual) IP address is used for all network components (EMS clients and managed devices). Thus, in such a setup, you need to configure only this IP address.

➤ **To configure the SEM server:**

1. Open the Session Experience Manager Server page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Session Experience Manager Server**).

Figure 15-1: Session Experience Manager Server Page

Session Experience Manager Server	
Server IP	0.0.0.0
Redundant Server IP	0.0.0.0
Port	5000
Interface Name	OAMP
QOE Connection by TLS	Disable

2. In the 'Server IP' field, enter the primary SEM server's IP address.

3. If Geographical-Redundancy HA mode exists, in the 'Redundant Server IP' field, enter the secondary SEM server's IP address.
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. (Optional) Configure a TLS connection with the SEM server:
 - a. From the 'QOE Connection by TLS' drop-down list, select **Enable**.
 - b. From the 'Qoe TLS Context Name' drop-down list, select the desired TLS Context, which defines the TLS settings (e.g., certificates).
6. Click **Submit**, and then save ("burn") your settings to flash memory.

15.1.2 Configuring Clock Synchronization between Device and SEM

To ensure accurate call quality statistics and analysis by the SEM server, you must configure the device and the SEM server with the same clock source for clock synchronization. In other words, you need to configure them with the same NTP server.

The NTP server can be one of the following:

- AudioCodes EMS server (also acting as an NTP server)
- Third-party, external NTP server

Once you have determined the NTP server, all the elements--device, SEM, and EMS--must be configured with the same NTP server address.

To configure, the NTP server's address on the device, see [Configuring Automatic Date and Time using SNTP](#) on page 99.

15.1.3 Enabling RTCP XR Reporting to SEM

In order for the device to be able to send voice metric reports to the SEM, you need to enable the RTP Control Protocol Extended Reports (RTCP XR) VoIP management protocol. RTCP XR defines a set of voice metrics that contain information for assessing VoIP call quality and diagnosing problems. Enabling RTCP XR means that the device can send RTCP XR messages, containing the call-quality metrics, to the SEM server.

For enabling RTCP XR reporting, see [Configuring RTCP XR](#) on page 451. For determining what to report to the SEM, see [Configuring Quality of Experience Profiles](#) on page 217.

15.2 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-metric monitoring profiles (Quality of Experience Profiles) can be configured and applied to specific network links, including IP Groups (see "Configuring IP Groups" on page 240), Media Realms (see "Configuring Media Realms" on page 227), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 231). 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 215.
- 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 224.

- 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 349).



Note: For your convenience, the device provides pre-configured Quality of Experience Profiles. One of these pre-configured profiles is the default Quality of Experience Profile. Therefore, if you do not configure a Quality of Experience Profile, this default is used.

The following procedure 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 15-2: Quality of Experience Profile - Add Record

3. Configure a QoE Profile according to the parameters described in the table below.
4. Click **Submit**.

Table 15-1: 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 15-3: Quality of Experience Page - Add Record Dialog Box

Add Record	
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
Submit Cancel	

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 15-2: 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_monitoredParameter]	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

Parameter	Description
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 = Pre-configured low sensitivity threshold values. Thus, reporting is done only if changes in parameters' values is significant. ▪ [2] Medium = (Default) Pre-configured medium sensitivity threshold values. ▪ [3] High = 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.
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.

15.3 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 240), Media Realms (see "Configuring Media Realms" on page 227), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 231).

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 224.
- 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 349).
- 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 following procedure 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 15-4: Bandwidth Profile Page - Add Record

Add Record	
Index	0
Name	ITSP-A
Egress Audio Bandwidth [Kbps]	64000
Ingress Audio Bandwidth [Kbps]	-1
Egress Video Bandwidth [Kbps]	-1
Ingress Video Bandwidth [Kbps]	-1
Total Egress Bandwidth [Kbps]	-1
Total Ingress Bandwidth [Kbps]	-1
Warning Threshold [%]	70
Hysteresis [%]	10
Generate Alarm	Enable
Submit Cancel	

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 15-3: 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).

15.4 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 216) and bandwidth utilization (configured in "Configuring Bandwidth Profiles" on page 221). 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 following procedure 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 15-5: Media Enhancement Profile Table - Add Record

3. Configure a Media Enhancement Profile according to the parameters described in the table below.
4. Click **Submit**.

Table 15-4: 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 15-6: Media Enhancement Rules - Add Record

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 15-5: Media Enhancement Rules Table Parameter Descriptions

Parameter	Description
Index CLI: rule-index [MediaEnhancementRules_RuleIndex]	Defines the index of the table row entry.

Parameter	Description
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 .

16 Control Network

This section describes configuration of the network at the SIP control level.

16.1 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 240) and SRDs (see "Configuring SRDs" on page 233).

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 216.
- **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 221.

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 231.



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 following procedure 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 16-1: Media Realm Page - Add Record Dialog Box

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 16-1: 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.
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 according to the BaseUDPPort parameter. For more information, see Configuring RTP Base UDP Port on page 153. ▪ The base UDP port number (BaseUDPPort parameter) must be greater than the highest UDP port configured for a SIP Interface (see Configuring SIP Interfaces on page 236). For example, if your highest configured UDP port for

Parameter	Description
	a SIP Interface is 6060, you must configure the BaseUDPPort parameter to any value greater than 6060. - The port must be different from ports configured for SIP traffic (i.e., ports configured for SIP Interfaces). For example, if the RTP port range is 6000 to 6999, the SIP port can be less than 6000 or greater than 6999. - 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.
Port Range End CLI: port-range-end [CpMediaRealm_PortRangeEnd]	(Read-only field) Displays the ending port for the range of media interface UDP ports. The device automatically populates the parameter with a value, calculated by the summation of the 'Port Range Start' parameter and 'Number of Media Session Legs' parameter (multiplied by the port spacing) minus 1: $\text{start port} + (\text{sessions} * \text{port spacing}) - 1$ For example, a port starting at 6,000, 5 sessions and 10 port spacing: $6,000 + (5 * 10) - 1 = 6,000 + (50) - 1 = 6,000 + 49 = 6,049$ The device allocates the UDP ports for RTP, RTCP and T.38 in "jumps" (spacing) of 5 or 10 (default), configured by the UdpPortSpacing parameter. For example, if the port range starts at 6000 and the UDP port spacing is 10, the available ports include 6000, 6010, 6020, 6030, and so on (depending on number of media sessions). For RTCP and T.38 traffic, the port offset from the RTP port used for the voice session (channel) is one and two, respectively. For example, if the voice session uses RTP port 6000, the RTCP port and T.38 port for the session is 6001 and 6002, respectively. However, you can configure the device to use the same port for RTP and T.38 packets, by setting the T38UseRTPPort parameter to 1.
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 216.

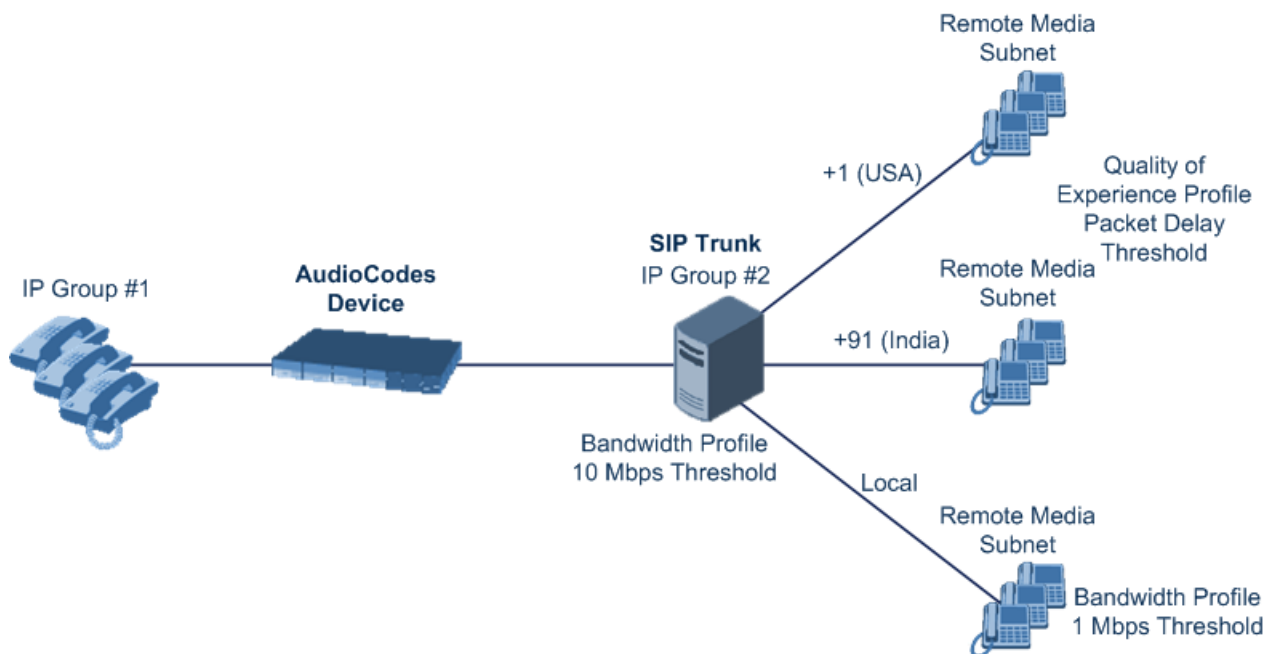
Parameter	Description
BW Profile CLI: bw-profile [CpMediaRealm_BWProfile]	Assigns a Bandwidth Profile to the Media Realm. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 221 .

16.2 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 216 and "Configuring Bandwidth Profiles" on page 221, 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 16-2: Remote Media Subnets Example



The following procedure 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 16-3: Remote Media Subnet - Add Record

The screenshot shows a dialog box titled "Add Record" with a close button (X) in the top right corner. It contains several input fields and dropdown menus:

- Index:** A text box containing the value "0".
- SubRealm Name:** An empty text box.
- Prefix Length:** A text box containing the value "16".
- Address Family:** A dropdown menu with "IPv4" selected.
- Destination IP:** A text box containing the value "0.0.0.0".
- QOE Profile Name:** A dropdown menu with "None" selected.
- BW Profile Name:** A dropdown menu with "None" selected.

At the bottom right of the dialog, there are two buttons: "Submit" (with a checkmark icon) and "Cancel" (with an X icon).

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 16-2: 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.
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.

16.3 Configuring SRDs

The SRD table lets you configure up to 500 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 [236](#))
- Associate it with an IP Group (see "Configuring IP Groups" on page [240](#))
- Associate it with a Proxy Set (see "Configuring Proxy Sets" on page [251](#))
- Associate it with an Admission Control rule (see Configuring Admission Control Table on page [326](#))
- Define it as a Classification rule for incoming SIP requests (see "Configuring Classification Rules" on page [332](#))
- Use it as a destination IP-to-IP routing rule (see Configuring SBC IP-to-IP Routing Rules on page [340](#))

The following procedure 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 16-4: SRD Settings Page

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 16-3: SRD Table Parameter Descriptions

Parameter	Description
Index [SRD_Index]	Defines an index for the new table record.
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 227). 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. ▪ [2] DisableWhenSingleNAT = No Media Anchoring. Media stream flows directly between endpoints if they are located behind the same NAT. 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

Parameter	Description
	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 302.
Block Unregistered Users CLI: block-un-reg-users [SRD_BlockUnRegUsers]	Determines whether the device blocks (rejects) incoming calls (INVITE requests) from unregistered users (pertaining to User-type IP Groups) for 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-registr [SRD_EnableUnAuthenticatedRegistrations]	Enables the device to accept REGISTER requests and register them in its registration database from new users that have not been authenticated by a proxy/registrar server (due to proxy down) and thus, re-routed to a User-type IP Group. In normal operation scenarios in which the proxy server is available, the device forwards the REGISTER request to the proxy and if authenticated by the proxy (i.e., device receives a success response), the device adds the user to its registration database. The routing to the proxy is according to the SBC IP-to-IP Routing table where the destination is the proxy's IP Group. However, when the proxy is unavailable (e.g., due to network connectivity loss), the device can accept REGISTER requests from new users if a matching alternative routing rule exists in the SBC IP-to-IP Routing table where the destination is the user's User-type IP Group (i.e., call survivability scenarios) and if the parameter is enabled. [0] Disable = The device rejects REGISTER requests from new users that were not authenticated by a proxy server. [1] Enable = (Default) The device accepts REGISTER requests from new users even if they were not authenticated by a proxy server, and registers the user in its registration database. Note: Regardless of the parameter, the device always accepts registration refreshes from users that are already registered in its database.

16.4 Configuring SIP Interfaces

The SIP Interface table lets you configure up to 500 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 associated with an SRD. For each SIP Interface, you can assign a SIP message policy rule, assign SIP message manipulation rules, enable TLS mutual authentication, enable TCP keepalive, assign a SSL/TLS certificate (TLS Context), and configure 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 following procedure 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
SIP Interface Name	
Network Interface	Not Configured
Application Type	GW & IP2IP
UDP Port	5060
TCP Port	5060
TLS Port	5061
SRD	0
Message Policy	None
TLS Context Name	None
TLS Mutual Authentication	
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 16-4: SIP Interface Table Parameter Descriptions

Parameter	Description
Index [SIPInterface_Index]	Defines an index for the new table record.
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 112). 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. - The base UDP port number (BaseUDPPort parameter) for RTP traffic must be greater than the highest UDP port configured for a SIP Interface. For example, if your highest configured UDP port for a SIP Interface is 6060, you must configure the BaseUDPPort parameter to any value greater than 6060. For more information on base UDP port, see Configuring RTP Base UDP Port on page 153. - 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: - 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).

Parameter	Description
SRD CLI: srd [SIPInterface_SRD]	Assigns an SRD ID to the SIP Interface (configured in "Configuring SRDs" on page 233). 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" on page 272).
TLS Context Name CLI: tls-context-name [SIPInterface_TLSTContext]	Assigns a TLS Context (SSL/TLS certificate) to the SIP Interface. The TLS Context is assigned by name, as configured in the 'Name' field of the TLS Contexts table. - Incoming calls: This TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call or classification to an IP Group based on Proxy Set fails. - Outgoing calls: This TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call. For more information about how certificates are associated with calls and for configuring TLS Contexts, see Configuring SSL/TLS Certificates on page 85.
TLS Mutual Authentication CLI: tls-mutual-auth [SIPInterface_TLSMutualAuthentication]	Enables TLS mutual authentication for the SIP Interface. [-1] Not Configured = (Default) The SIPRequireClientCertificate 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.

Parameter	Description
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).
Web: Pre Classification ManSet CLI: preclassification-manset [SIPInterface_PreClassificationManipulationSet]	Assigns a Message Manipulation Set ID to the SIP Interface. This lets you apply SIP message manipulation rules on incoming SIP initiating-dialog request messages (not in-dialog), received on this SIP Interface, prior to the Classification process. By default, no Message Manipulation Set ID is defined. For configuring Message Manipulation Sets, see Configuring SIP Message Manipulation on page 266. Notes: ▪ The Message Manipulation Set assigned to a SIP Interface that is associated with an outgoing call, is ignored. Only the Message Manipulation Set assigned to the associated IP Group is applied to the outgoing call. ▪ If both the SIP Interface and IP Group associated with the incoming call are assigned a Message Manipulation Set, the one assigned to the SIP Interface is applied first.

16.5 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 251).

IP Groups can be used for the following:

- Classification of incoming SIP dialog-initiating requests (e.g., INVITE messages) to IP Groups based on Proxy Set. If the source address of the incoming SIP dialog is defined for a Proxy Set, the device assigns ("bonds") the SIP dialog to the IP Group associated with the Proxy Set. The feature is configured using the IP Group table's 'Classify by Proxy Set' parameter. For more information and recommended security guidelines, see the parameter's description, later in this section.
- Representing the source and destination of the call in IP-to-IP Routing rules (see [Configuring SBC IP-to-IP Routing Rules](#) on page 340).
- 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 259).

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 216.
- **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 221.



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.
- If different SRDs are configured in the IP Group and Proxy Set tables, the SRD defined for the Proxy Set takes precedence.

The following procedure 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:

Common	
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
QoS Profile	None
Bandwidth Profile	None
Media Enhancement Profile	None
Always Use Source Address	No

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

Table 16-5: 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.

Parameter	Description
	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: ▪ 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 251.
SIP Group Name CLI: sip-group-name [IPGroup_SIPGroupName]	Defines the SIP Request-URI host name 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. In other words, it replaces the original host name. The valid value is a string of up to 100 characters. The default is an empty field.

Parameter	Description
	Note: - If this parameter is not configured, the value of the global parameter, ProxyName is used instead (see "Configuring Proxy and Registration Parameters" on page 263). - The parameter overrides inbound message manipulation rules that manipulate the host name in Request-URI, To, and/or From SIP headers. If you configure the parameter and you want to manipulate the host name in any of these SIP headers, you must apply your manipulation rule (Manipulation Set ID) to the IP Group as an Outbound Message Manipulation Set (see the IPGroup_OutboundManSet parameter), when the IP Group is the destination of the call. If you apply the Manipulation Set as an Inbound Message Manipulation Set (see the IPGroup_InboundManSet parameter), when the IP Group is the source of the call, the manipulation rule is overridden by the SIP Group Name parameter.
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 259).
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 233. - 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 227). 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 275. 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

Parameter	Description
	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 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.
UI 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: <pre>User-to-User: 00FA080019001038F725B3;encoding=hex</pre> 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 216.
Bandwidth Profile CLI: bandwidth-profile [IPGroup_BWProfile]	Assigns a Bandwidth Profile rule. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 221.
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 224.
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 on page 132.
CLI: Msg-Man-User-Defined-	Defines a value for the SIP user part that can be used in

Parameter	Description
String1 [IPGroup_MsgManUserDef1]	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 266.
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. For configuring Message Manipulation rules, see "Configuring SIP Message Manipulation" on page 266.

Parameter	Description
SBC Parameters	
Classify By Proxy Set CLI: classify-by-proxy-set [IPGroup_ClassifyByProxySet]	Enables classification of incoming SIP dialogs (INVITEs) to Server-type IP Groups based on Proxy Set (assigned using the IPGroup_ProxySetName parameter). [0] Disable [1] Enable = (Default) The device searches the Proxy Set table for a Proxy Set that is configured with the same source IP address as that of the incoming INVITE (if host name, then according to the dynamically resolved IP address list). If such a Proxy Set is found, the device classifies the INVITE as belonging to the IP Group associated with the Proxy Set. Note: - The parameter is applicable only to Server-type IP Groups. - For security, it is recommended to classify SIP dialogs based on Proxy Set only if the IP address of the IP Group is unknown. In other words, if the Proxy Set associated with the IP Group is configured with an FQDN. In such cases, the device classifies incoming SIP dialogs to the IP Group based on the DNS-resolved IP address. If the IP address is known, it is recommended to use a Classification rule instead (and disable the Classify by Proxy Set feature), where the rule is configured with not only the IP address, but also with SIP message characteristics to increase the strictness of the classification process (see Configuring Classification Rules on page 332). The reason for preferring classification based on Proxy Set when the IP address is unknown is that IP address forgery (commonly known as IP spoofing) is more difficult than malicious SIP message tampering and therefore, using a Classification rule without an IP address offers a weaker form of security. When classification is based on Proxy Set, the Classification table for the specific IP Group is ignored. - If you have assigned the same Proxy Set to multiple IP Groups, disable the parameter and instead, use Classification rules to classify incoming SIP dialogs to these IP Groups. If the parameter is enabled, the device is unable to correctly classify incoming INVITEs to their appropriate IP Groups. - Classification by Proxy Set occurs only if classification based on the device's registration database fails (i.e., the INVITE is not from a registered user).
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 266. Note: The IPGroup_SIPGroupName parameter overrides inbound message manipulation rules (assigned to the IPGroup_InboundManSet parameter) that manipulate the host name in Request-URI, To, and/or From SIP headers. If you want to manipulate the host name using message manipulation rules in

Parameter	Description
	any of these SIP headers, you must apply your manipulation rule (Manipulation Set ID) to the IP Group as an Outbound Message Manipulation Set (see the IPGroup_OutboundManSet parameter), when the IP Group is the destination of the call.
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 266. Note: If you assign a Message Manipulation Set ID that includes rules for manipulating the host name in the Request-URI, To, and/or From SIP headers, the parameter overrides the IPGroup_SIPGroupName parameter.
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 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 402). This is applicable only to User-type IP Groups. ✓ Authenticates SIP servers. 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	Defines call forking of INVITE messages to up to five separate SIP outgoing legs for User-type IP Groups. This occurs if multiple

Parameter	Description
[IPGroup_EnableSBCCClientForking]	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. - 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 to use as a call matching characteristic based on destination URIs. The parameter is used for classification and routing purposes. The device first uses the parameter's settings as a matching

Parameter	Description
	characteristic (input) to classify the incoming INVITE to an IP Group (source IP Group) in the Classification table. Once classified, the device uses the parameter for routing the call. For example, if set to To, the URI in the To header of the incoming INVITE is used as a matching characteristic for classifying the call to an IP Group in the Classification table. Once classified, the device uses the URI in the To header as the destination. ▪ [-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: ▪ The 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, 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.
SIP Connect sip-connect [IPGroup_SIPConnect]	Defines the IP Group as a registered server that represents multiple users. The device saves registrations received from the IP Group, with the IP address as a key in its registration database. The device classifies incoming SIP dialog requests (e.g., INVITEs) from the IP Group according to the received IP address. For requests routed to the IP Group users, the device replaces the Request-URI header with the incoming To header (which contains the remote phone number). ▪ [0] No (default) ▪ [1] Yes Note: The parameter is applicable only to User-type IP Groups.
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

Parameter	Description
	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). - To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.

16.6 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 configured as an IP address and/or DNS host name (FQDN), enabling you to implement load balancing and redundancy between multiple servers. If you configure the address as an FQDN, you can configure the method (A-record DNS, SRV, or NAPTR) for resolving the domain name to an IP address. The device supports up to 30 DNS-resolved IP addresses. (If the DNS resolution provides more than this number, the device uses the first 30 IP addresses in the received list, and ignores the rest.)

You can assign each Proxy Set with a specific SSL/TLS certificate (TLS Context), enabling the use of different certificates per SIP entity (IP Group).

You can enable the device's keep-alive feature per Proxy Set, which determines whether proxies (addresses) configured for the Proxy Set are online or offline. If offline, the device will not route the call to the specific proxy. You can configure the device to send either SIP OPTIONS or REGISTER messages for the keep-alive. The keep-alive feature is required when using the proxy load-balancing or redundancy feature. For load-balancing, the device performs keep-alive on all proxies. For Parking-type redundancy, the device performs keep-alive only on the currently active proxy. For Homing-type redundancy, the device performs keep-alive on the current proxy as well as the "main" proxy. When using SIP OPTIONS, you can configure the device to consider the proxy as offline if specific SIP response codes are received from the keep-alive messages. To ensure that a previously offline proxy is now online, you can configure the number of required consecutive successful keep-alive messages (SIP OPTIONS only) before the device considers the proxy as being online. This mechanism avoids the scenario in which the device falsely detects a proxy as being online when it is actually offline, resulting in call routing failure. To view the connectivity status of Proxy Sets, see Viewing Proxy Set Status on page 448.

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 240.



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 332).

The following procedure 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 16-5: Proxy Sets Table Page

Proxy Set ID	
Proxy Set ID	1

	Proxy Address	Transport Type
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

Proxy Name	
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
TLS Context	-1

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 16-6: 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.
Proxy Address CLI: voip-network proxy-ip > proxy-address [Proxylp_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)

Parameter	Description
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 setting 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.
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 the host names (and their weights). The device then performs DNS A-record queries per host name (according to the received weights). 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. Note: An SRV query can return up to four host names. For each host name, the subsequent DNS A-record query can be resolved into up to 15 IP addresses. However, the device supports up to 30 DNS-resolved IP addresses. If the device receives more than this number of DNS-resolved IP addresses, the device uses the first 30 IP addresses in the received list, and ignores the rest.

Parameter	Description
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 messages 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. You can also configure whether to use the device's IP address or string name ("gateway name") in the OPTIONS message (see the UseGatewayNameForOptions parameter). [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). Notes: - Proxy keep-alive using REGISTER messages (Using REGISTER option) is applicable only to the Parking redundancy mode ('Redundancy Mode' parameter configured to Parking). - For Survivability mode for User-type IP Groups, you must enable this Proxy Keep-Alive feature. - If you enable this Proxy Keep-Alive feature and the proxy uses the TCP/TLS transport type, you can enable CRLF Keep-Alive feature, using the UsePingPongKeepAlive parameter. - If you enable this Proxy Keep-Alive feature, the device can operate with multiple proxy servers (addresses) for redundancy and load balancing (see the 'Proxy Load Balancing Method' 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 [ProxySet_KeepAliveFailureResp]	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 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.
Success Detection Retries success-detect-retries	Defines the minimum number of consecutive, successful keep-alive messages that the device sends to an offline proxy, before

Parameter	Description
[ProxySet_SuccessDetectionRetries]	the device considers the proxy as being online. The valid range is 1 to 10. The default is 1. Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options.
Success Detection Interval success-detect-int [ProxySet_SuccessDetectionInterval]	Defines the interval (in seconds) between each keep-alive retries (as configured by the 'Success Detection Retries' parameter) that the device performs for offline proxies. The valid range is 1 to 30. The default is 10. Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options.
Failure Detection Retransmissions fail-detect-rtx [ProxySet_FailureDetectionRetransmissions]	Defines the maximum number of UDP retransmissions that the device sends to an offline proxy, before the device considers the proxy as being offline. The valid range is -1 to 255. The default is -1 (i.e., the settings of the global parameter SIPMaxRtxis applied). Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options.
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.

Parameter	Description
	[0] No (default) [1] Yes = The device sends SIP INVITE/REGISTER messages to the first address listed in the Proxy Address table that is configured for the Proxy Set. If a SIP response is received and this response code is configured in the Alternative Routing Reasons table (see Configuring SIP Response Codes for Alternative Routing Reasons on page 349) for SBC, the device assumes that the proxy is down and sends the message to the next available proxy (address) in the list.
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 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 233.
Web: Classification Input CLI: voip-network proxy-set > classification-input [ProxySet_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.
Web/EMS: TLS Context Index CLI: tls-context-index [ProxySet_TLSContext]	Assigns a TLS Context (SSL/TLS certificate) to the Proxy Set. The TLS Context is assigned by index number, as configured in the TLS Contexts table. - Incoming calls: If the 'Transport Type' parameter (above) is set to TLS and the incoming call is successfully classified to an IP Group based on this Proxy Set, this TLS Context is used. If the 'Transport Type' parameter is set to UDP or classification to this Proxy Set fails, this TLS Context is not used. Instead, the device uses the TLS Context configured for the SIP Interface (see Configuring SIP Interfaces on page 236) used for the call; otherwise, the default TLS Context (ID 0) is used. - Outgoing calls: If the 'Transport Type' parameter (above) is set to TLS and the outgoing call is sent to an IP Group that is

Parameter	Description
	associated with this Proxy Set, this TLS Context is used. Instead, the device uses the TLS Context configured for the SIP Interface (see Configuring SIP Interfaces) used for the call; otherwise, the default TLS Context (ID 0) is used. If the 'Transport Type' parameter is set to UDP, the device uses UDP to communicate with the proxy and no TLS Context is used. For more information about how certificates are associated with calls and for configuring TLS Contexts, see Configuring SSL/TLS Certificates on page 79.

This page is intentionally left blank.

17 SIP Definitions

This section describes configuration of SIP parameters.

17.1 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 505.

17.2 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 following procedure 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 17-1: Account Table - Add Record

The 'Add Record' dialog box contains the following fields and values:

Field	Value
Index	0
Served IP Group	-1
Serving IP Group	1
User Name	
Password	
Host Name	
Register	No
Contact User	
Application Type	GW & IP2IP

Buttons: Submit, Cancel

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 447.

Table 17-1: 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.

Parameter	Description
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 CLI: register [Account_Register]	Enables registration. ▪ [0] No (Default) ▪ [1] Regular = Regular registration process. For more information, see "Regular Registration Mode" on page 261. ▪ [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 262. 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.

17.2.1 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.6.80A.014
Content-Length: 0
```

17.2.2 Single Registration for Multiple Phone Numbers using GIN

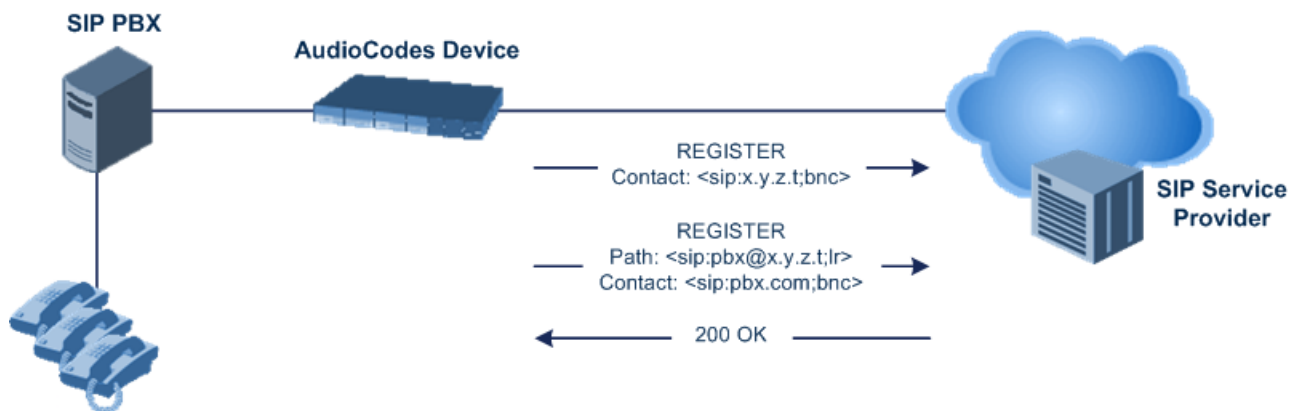
When you configure the registration mode in the Account table to **GIN**, the Global Identifiable Number (GIN) registration method is used, according to RFC 6140. The device performs GIN-based registration of users to a SIP registrar on behalf of a SIP PBX. In effect, the PBX registers with the service 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.

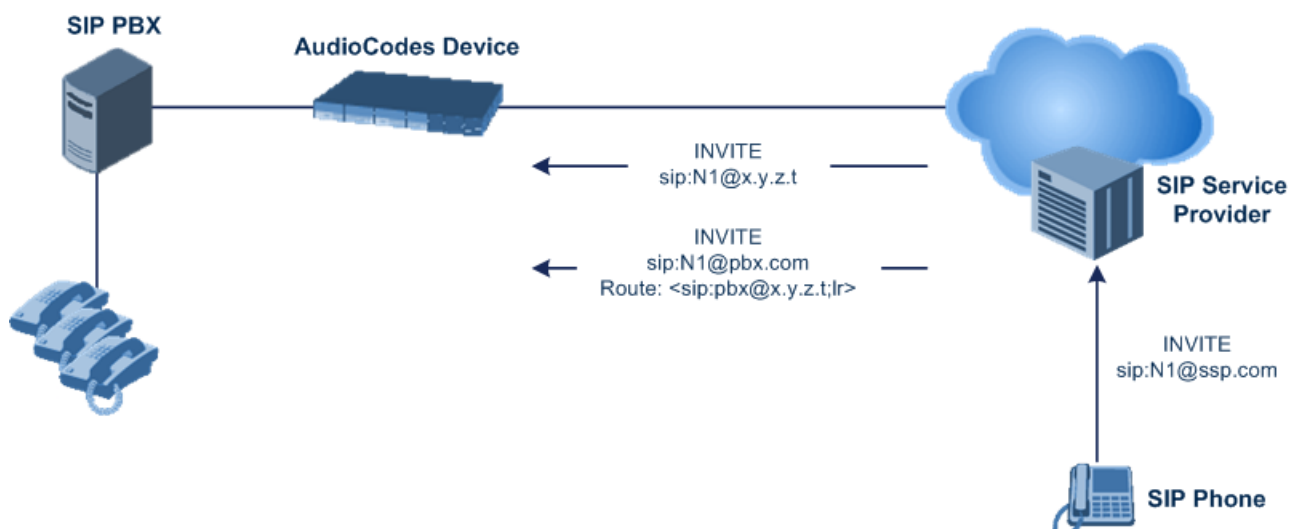
The SIP REGISTER request sent by the device for GIN registration with a SIP server provider contains the Require and Proxy-Require headers. These headers contain the token 'gin'. The Supported header contains the token 'path' and the URI in the Contact header contains the parameter 'bnc' without a user part:

```
Contact: <sip:198.51.100.3;bnc>;
```

The figure below illustrates the GIN registration process:



The figure below illustrates an incoming call using GIN:



17.3 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 505.



Note: To view the registration status of endpoints with a SIP Registrar/Proxy server, see "Viewing Registration Status" on page 447.

➤ **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 259)

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 251 for a description of this page).

17.3.1 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 9000/v.6.80.010.006
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 "a9a031cfddeb10d91c8e7b4926086f7e".
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 9000/v.6.80.010.006
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
```

17.4 Configuring SIP Message Manipulation

The Message Manipulations table lets you configure up to 500 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 specific calls.

- Message manipulation rules can be applied pre- or post-classification:
 - **Pre-classification Process:** Message manipulation can be done on incoming SIP dialog-initiating messages (e.g., INVITE) prior to the classification process. You configure this by assigning the Manipulation Set ID to the SIP Interface on which the call is received (see [Configuring SIP Interfaces](#) on page 236).
 - **Post-classification Process:** Message manipulation can be done on inbound and/or outbound SIP messages after the call has been successfully classified. You configure this by assigning the Manipulation Set ID to the relevant IP Group in the IP Group table (see [Configuring IP Groups](#) on page 240).

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

- Multiple manipulation rules using the same condition. The following figure shows a configuration example where rules 1 and 2 ('Row Rule' configured to **Use Previous Condition**) use the condition configured for rule 0 ('Row Rule' configured to **Use Current Condition**). For more information, see the description of the 'Row Rule' parameter in this section.

Figure 17-2: Configuration Example of Message Manipulation Rules using Same Condition

Index	Manipulation Name	Manipulation Set ID	Message Type	Condition	Action Subject	Action Type	Action Value
0	To header for urgent	0	invite.request	header.request-uri.url.user == '100'	header.to	Modify	header.to + ';urgent=1'
1	Add emergency	0			header.priority	Add	'emergency'
2	User-agent	0			header.user-agent	Modify	'trunk-a'

The figure below illustrates a SIP message manipulation example:

Figure 17-3: SIP Header Manipulation Example





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 SIP Group Name (IPGroup_SIPGroupName) parameter overrides inbound message manipulation rules that manipulate the host name in Request-URI, To, and/or From SIP headers. If you configure a SIP Group Name for the IP Group (see Configuring IP Groups on page 240) and you want to manipulate the host name in any of these SIP headers, you must apply your manipulation rule (Manipulation Set ID) to the IP Group as an Outbound Message Manipulation Set (IPGroup_OutboundManSet), when the IP Group is the destination of the call. If you apply the Manipulation Set as an Inbound Message Manipulation Set (IPGroup_InboundManSet), when the IP Group is the source of the call, the manipulation rule will be overridden by the SIP Group Name.

The following procedure 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 17-4: Message Manipulations Table - Add Record Dialog Box

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 17-5: Message Manipulations Page

Add + Insert + Edit ✎ Delete 🗑 Up ↑ Down ↓ Show/Hide 📄							
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 Suffix	'.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='Unknown'	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 17-2: 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 CLI: message-type [MessageManipulations_MessageType]	Defines the SIP message type that you want to manipulate. 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).

Parameter	Description
ion]	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_Action Subject]	Defines the SIP header upon which the manipulation is performed. The valid value is a string (case-insensitive).
Action Type CLI: action-type [MessageManipulations_Action Type]	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_Action Value]	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.
Row Role CLI: row-role [MessageManipulations_RowRole]	Determines which message manipulation condition (configured by the 'Condition' parameter) to use for the rule. [0] Use Current Condition = (Default) The condition configured in the table row of the rule is used. [1] Use Previous Condition = The condition configured in the first table row above the rule that is configured to Use Current Condition is used. For example, if Index 3 is configured to Use Current Condition and Index 4 and 5 are configured to Use Previous Condition, Index 4 and 5 use the condition

Parameter	Description
	configured for Index 3. A configuration example is shown in the beginning of this section. The option allows you to use the same condition for multiple manipulation rules. Notes: ▪ When configured to Use Previous Condition, the 'Message Type' and 'Condition' parameters are not applicable and if configured are ignored. When multiple manipulation rules apply to the same header, the next rule applies to the resultant string of the previous rule.

17.5 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.

To apply SIP Message Policy rules, you need to assign them to SIP Interfaces associated with the relevant IP Groups (see "Configuring SIP Interfaces" on page 236).

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 following procedure 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 17-6: 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
Submit Cancel	

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 17-3: 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.
SIP Method Blacklist-Whitelist Policy	
Method List CLI: method-list [MessagePolicy_MethodList]	Defines SIP methods (e.g., INVITE\BYE) to blacklist or whitelist. Multiple methods are separated by a backslash (\). The method values are case-insensitive.
Method List Type CLI: method-list-type [MessagePolicy_MethodListType]	Defines the policy (blacklist or whitelist) for the SIP methods specified in the 'Method List' parameter (above). ▪ [0] Policy Blacklist = The specified methods are rejected. ▪ [1] Policy Whitelist = (Default) Only the specified methods are allowed; the others are rejected.
SIP Body Blacklist-Whitelist Policy	
Body List CLI: body-list [MessagePolicy_BodyList]	Defines the SIP body type (i.e., value of the Content-Type header) to blacklist or whitelist. For example, application/sdp. The values of this parameter are case-sensitive.

Parameter	Description
Body List Type CLI: body-list-type [MessagePolicy_BodyListType]	Defines the policy (blacklist or whitelist) for the SIP body specified in the 'Body List' parameter (above). ▪ [0] Policy Blacklist =The specified SIP body is rejected. ▪ [1] Policy Whitelist = (Default) Only the specified SIP body is allowed; the others are rejected.

18 Coders and Profiles

This section describes configuration of the coders and SIP profiles parameters.

18.1 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 [240](#)

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 following procedure 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 18-1: IP Profile Table - Add Record

Common GW SBC	
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
Submit Cancel	

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 18-1: IP Profile Settings Table Parameter Descriptions

Parameter	Description
Common	
Web: Index CLI: [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	Defines the DiffServ value for Premium Control CoS content (Call Control applications).

Parameter	Description
[IpProfile_SigIPDiffServ]	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. ▪ [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]	Defines the device's handling of calls when RTP packets (media) are not received within a user-defined timeout (configured by the BrokenConnectionEventTimeout parameter). ▪ [0] Ignore = The call is maintained despite no media and is released when signaling ends the call (i.e., SIP BYE). ▪ [1] Disconnect = (Default) The device ends the call. ▪ [2] Reroute = (SBC application only) The device ends the call and searches the IP-to-IP Routing table for a matching rule and if found, generates a new INVITE to the corresponding destination (i.e., alternative routing). You can configure a routing rule whose matching characteristics is explicitly for calls with broken RTP connections. This is done using the Call Trigger parameter, as described in Configuring SBC IP-to-IP Routing Rules on page 340. Note: ▪ The device can only detect a broken RTP connection if silence compression is disabled for the RTP session. ▪ If during a call the source IP address (from where the RTP packets are received by the device) is changed without notifying the device, the device rejects these RTP packets. To overcome this, configure the DisconnectOnBrokenConnection parameter to 0. By this configuration, 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.

Parameter	Description
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: <pre> 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:bnuYZnMxSfUiGitviWJZmzr7OF3AiR0015Vnh0kH 2^31 </pre> 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 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: <pre> a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:R1VyAlxV/qwBjkEkl4kSJyl3wCtYeZLq1/QFuxw 2^31 1:1 </pre> 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.

Parameter	Description
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_SBCAllowedCodersGroupID]	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 Groups on page 329. For a description of the Allowed Coders feature, see "Restricting Coders" on page 303.

Parameter	Description
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 331.
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 after 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 303.
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.

Parameter	Description
Web: P-Asserted-Identity CLI: sbc-assert-identity [lpProfile_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 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 [lpProfile_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 309. ▪ [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. Note: If the Diversion header is used, you can specify the URI type (e.g., "tel:") to use in the header, using the SBCDiversionUriType parameter.
Web: History-Info Mode CLI: sbc-history-info-mode [lpProfile_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 309. ▪ [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 [lpProfile_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.

Parameter	Description
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 setup and after call establishment.
Web: Remote re-INVITE 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.

Parameter	Description
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: - 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. - The incoming INVITE is identified as a REFER-resultant INVITE according to this special prefix. - 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. - 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 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.,

Parameter	Description
	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 Support 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 Behavior CLI: sbc-rmt-early-media-rtp [IpProfile_SBCRemoteEarlyMediaRTP]	Defines whether the destination UA sends RTP immediately after it sends 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 311.

Parameter	Description
Web: Remote RFC 3960 Gateway Model Support CLI: sbc-rmt-rfc3960-supp [IpProfile_SBCRemoteSupports RFC3960]	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 311. [1] Supported = UA is capable of receiving 18x messages with delayed RTP.
Web: Remote Can Play Ringback CLI: sbc-rmt-can-play-ringback [IpProfile_SBCRemoteCanPlay Ringback]	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 311.
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.

Parameter	Description
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 'SBC Remote Hold Format' parameter is set to 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 incoming INVITEs containing the Replaces header for the SIP entity (which does not support the header) associated with the IP Profile. The Replaces header is used to replace an existing SIP dialog with a new dialog such as in call transfer or call pickup. [0] Standard = (Default) The SIP entity supports INVITE messages containing Replaces headers. The device forwards the INVITE message containing the Replaces header to the SIP entity. The device may change the value of the Replaces header to reflect the call identifiers of the leg. [1] Handle Locally = The SIP entity does not support INVITE messages containing Replaces headers. The device terminates the received INVITE containing 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. [2] Keep as is = The SIP entity supports INVITE messages containing Replaces headers. The device forwards the Replaces header as is in incoming REFER and outgoing INVITE messages from/to the SIP entity (i.e., Replaces header's value is unchanged). 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

Parameter	Description
	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.
Adapt RFC2833 BW to Voice coder BWSbc-adapt-rfc2833-bw-voice-bw [IpProfile_SBCAdaptRFC2833B WToVoiceCoderBW]	Defines the 'telephone-event' type (8000 or 16000) in the SDP that the device sends in the outgoing SIP 200 OK message for DTMF payload negotiation (sampling rate). [0] Disable = (Default) The device always sends the 'telephone-event' as 8000 in the outgoing SIP 200 OK, even if the SDP of the incoming INVITE contains multiple telephone-event types (e.g., 8000 and 16000). [1] Enable = The type of 'telephone-event' that the device sends in the outgoing SIP 200 OK message is according to the coder type (narrowband or wideband). If narrowband, it sends the 'telephone-event' as 8000; if wideband, it sends it as 16000. An example when the parameter is configured to Enable is shown below, whereby the 'telephone-event' is "16000" in the outgoing message due to the wideband coder: SDP in incoming INVITE: <pre> a=rtpmap:97 AMR-WB/16000/1 a=fmtp:97 mode-change-capability=2 a=rtpmap:98 AMR-WB/16000/1 a=fmtp:98 octet-align=1; mode-change-capability=2 a=rtpmap:100 AMR/8000/1 a=fmtp:100 mode-change-capability=2 a=rtpmap:99 telephone-event/16000/1 a=fmtp:99 0-15 a=rtpmap:102 telephone-event/8000/1 a=fmtp:102 0-15 </pre> SDP in outgoing 200 OK: <pre> m=audio 6370 RTP/AVP 97 99 a=rtpmap:99 telephone-event/16000/1 a=fmtp:99 0-15 a=sendrecv a=ptime:20 a=maxptime:120 a=rtpmap:97 AMR-WB/16000 a=fmtp:97 mode-change-capability=2;mode-set=0,1,2,3,4,5,6,7, </pre>
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 the ptime according to the 'Preferred Ptime' parameter (see below) if it is configured to a non-zero value. Note: Regardless of the settings of this parameter, if a non-zero value is configured for the 'Preferred Ptime' parameter (see below), it is used as the ptime in the SDP offer.
Web: Preferred Ptime CLI: sbc-preferred-ptime [IpProfile_SBCPreferredPTime]	Defines the packetization time (ptime) in msec for the SIP entity associated with the IP Profile, in the outgoing SDP offer. If the 'SDP Ptime Answer' parameter (see above) is configured to Preferred Value [2] and the 'Preferred Ptime' parameter is

Parameter	Description
	configured to a non-zero value, the configured ptime is used (enabling ptime transrating if the other side uses a different ptime). If the 'SDP Ptime Answer' parameter is configured to Remote Answer [0] or Original Offer [1] and the 'Preferred Ptime' parameter is configured to a non-zero value, the configured value is used as the ptime in the SDP offer. The valid range is 0 to 200. The default is 0 (i.e., a preferred ptime is not used).
Web: Use Silence Suppression CLI: sbc-use-silence-supp [lpProfile_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. Note: The parameter requires DSP resources.
Web: Play RBT To Transferee CLI: sbc-play-rbt-to-xferee [lpProfile_SBCPlayRBTTToTransferee]	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 399.

Parameter	Description
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.

This page is intentionally left blank.

Part V

Session Border Controller Application

19 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 406.
- 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 611.

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.

19.1 SIP Network Definitions

The device's SBC application can implement multiple SIP signaling and RTP (media) interfaces.

19.2 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.

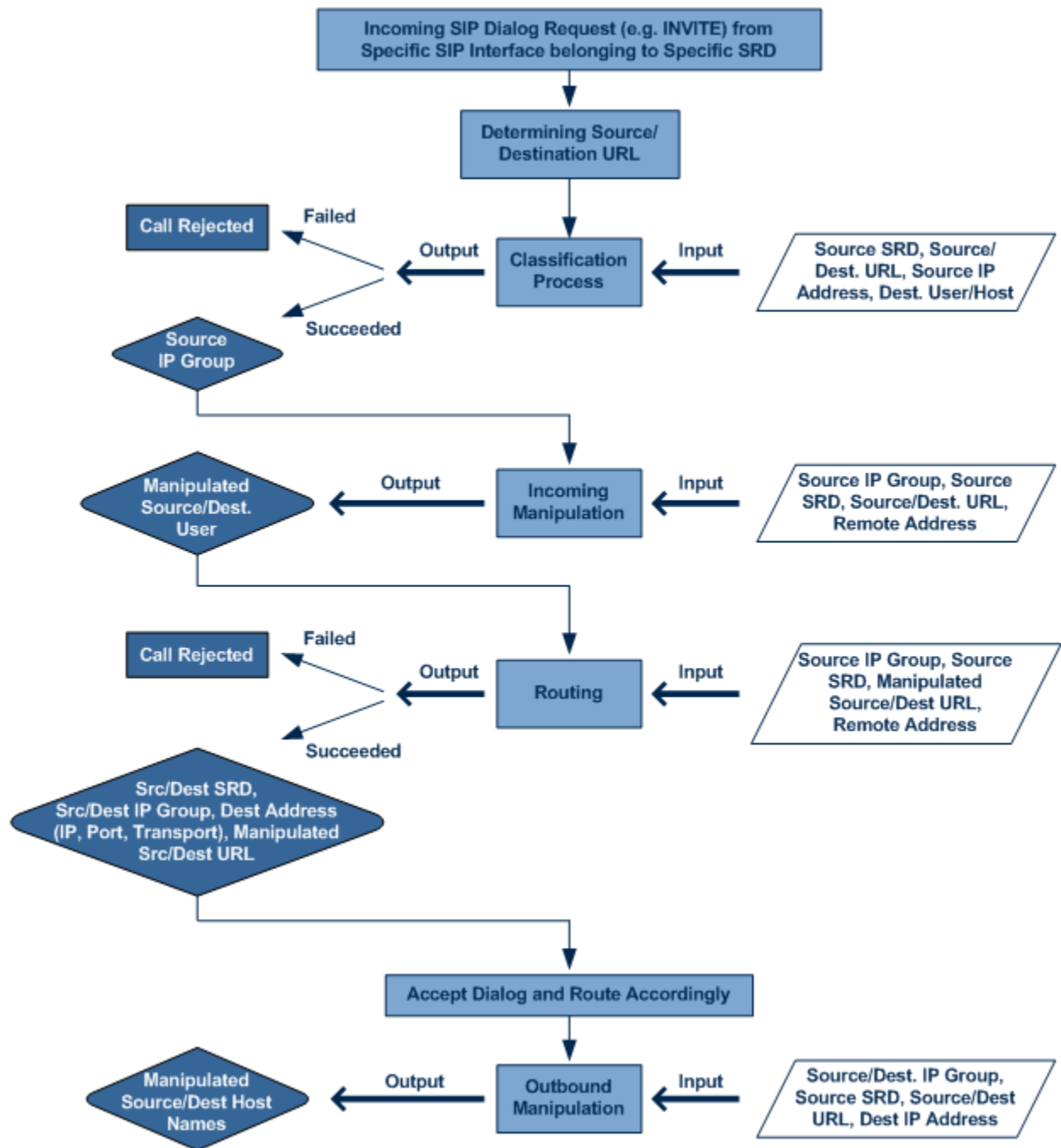


Note: You can determine the SIP header from where the device obtains the source URL in the incoming SIP request. This is done in the IP Group table using the 'Source URI Input' parameter.

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 332.
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 340.
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 351.
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 266.

The flowchart below illustrates the SBC process:

Figure 19-1: Routing Process



19.3 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.

19.3.1 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, SBCRandomizeExpires, 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).

19.3.2 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.

19.3.3 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.

19.3.4 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 feature is configured using the 'User Registration Grace Time' parameter (SBCUserRegistrationGraceTime).

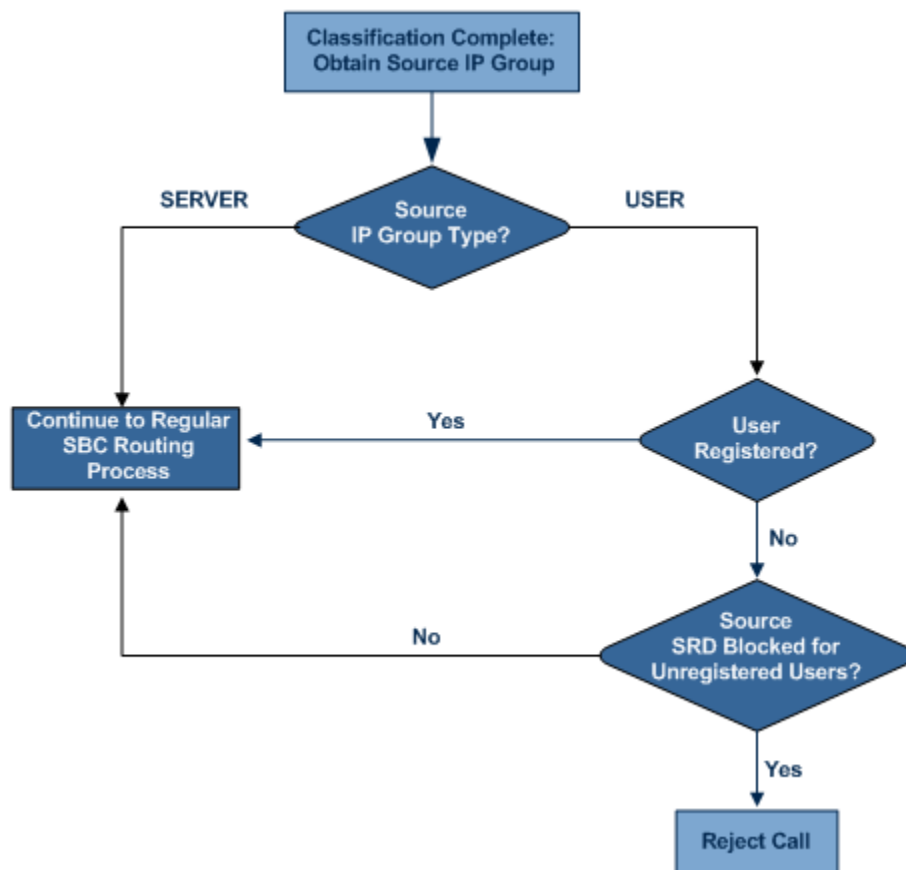
The device keeps registered users in its Users Registration database even if connectivity with the SIP proxy server is lost (i.e., proxy does not respond to users' registration refresh requests). The device removes users from the database only when their registration expiry time is reached (with the additional grace period, if configured).

19.3.5 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 19-2: Blocking Incoming Calls from Unregistered Users



19.4 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 301.

- **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 302).

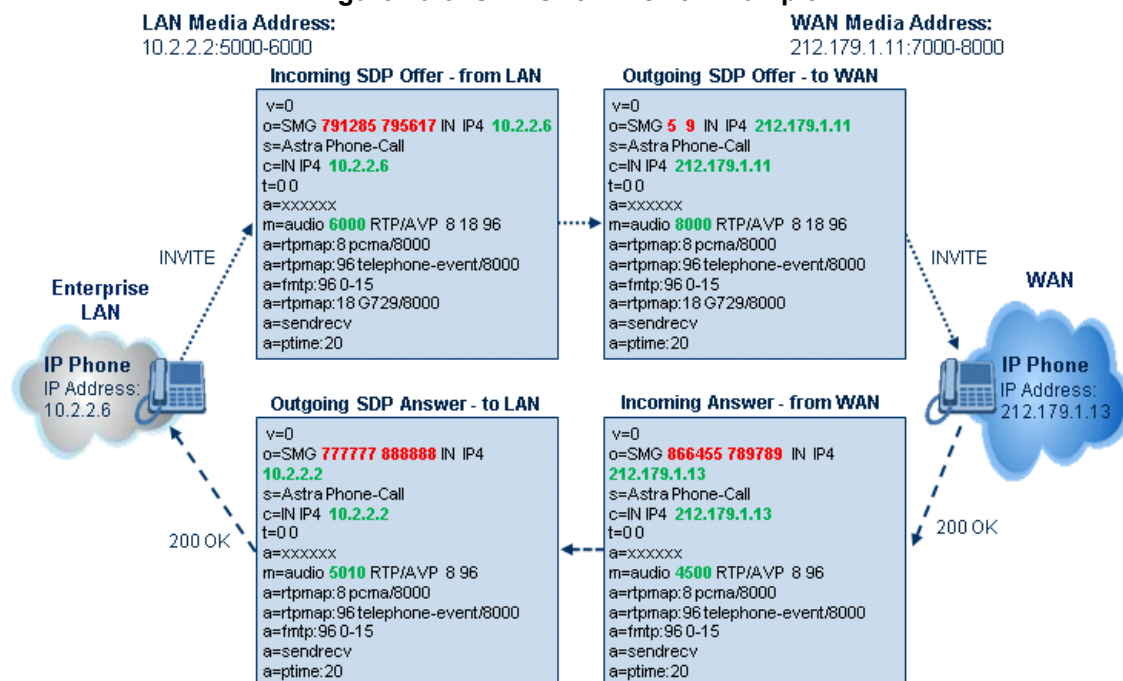
19.4.1 Media Anchoring without Transcoding (Transparent)

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 19-3: SDP Offer/Answer Example



19.4.2 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.

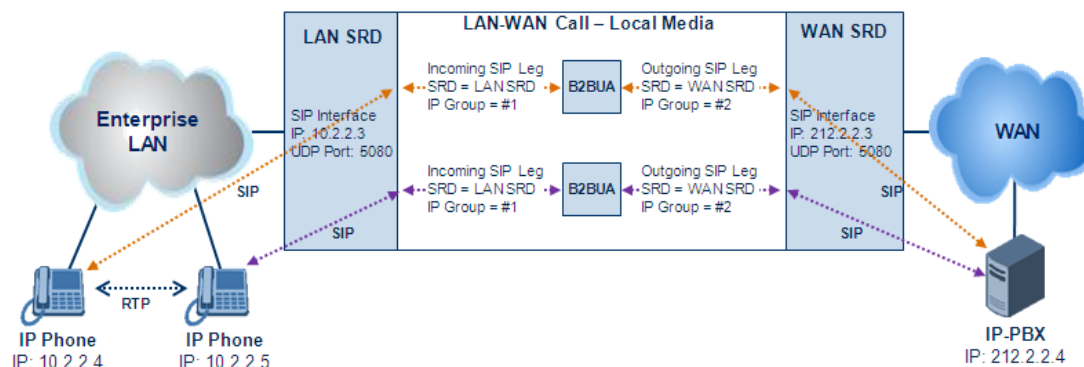
By default, media packets traverse the device to solve NAT problems, enforce media security policy, perform media transcoding between the two legs, and media monitoring. In certain deployments, specific calls do not require media anchoring, for example, when there is no need for NAT, security, or transcoding. 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 19-4: 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.

19.4.3 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.

19.4.4 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.

19.4.5 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.

19.4.6 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.

19.5 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.

19.6 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.

19.6.1 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 402). 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 240).

19.6.2 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.

19.7 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 even 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.

19.7.1 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'.

19.7.1.1 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:

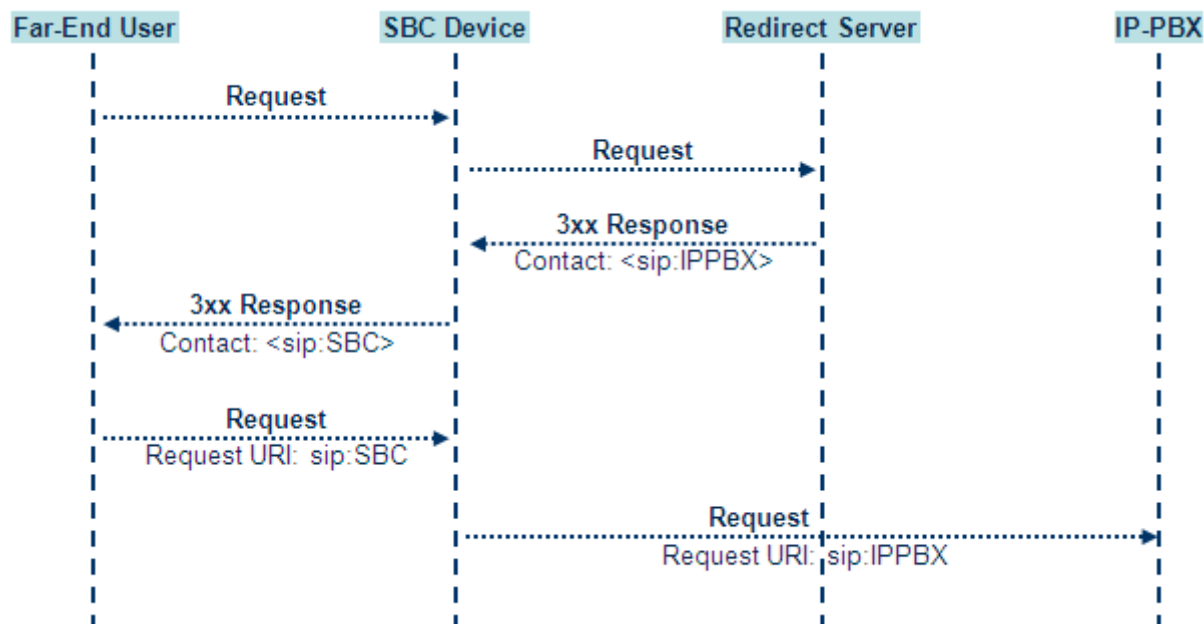
- Changes the host part to the device's IP address – this change causes the remote user agent to send the INVITE to the device.
- Adds a special prefix ("T~&R_") to the Contact user part – to identify the new INVITE as a 3xx resultant INVITE.

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 19-5: 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 value (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).

19.7.1.2 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).

19.7.2 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. If the Diversion header is used, you can specify the URI type (e.g., "tel:") to use in the header, using the `SBCDiversionUriType` parameter.

This feature is configured in the IP Profile table (IPProfile parameter) using the following 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 19-1: 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.
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.

19.7.3 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 where 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.

19.7.4 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.

19.7.5 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'.

19.7.6 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 19-6: SBC Early Media RTP 18x without SDP

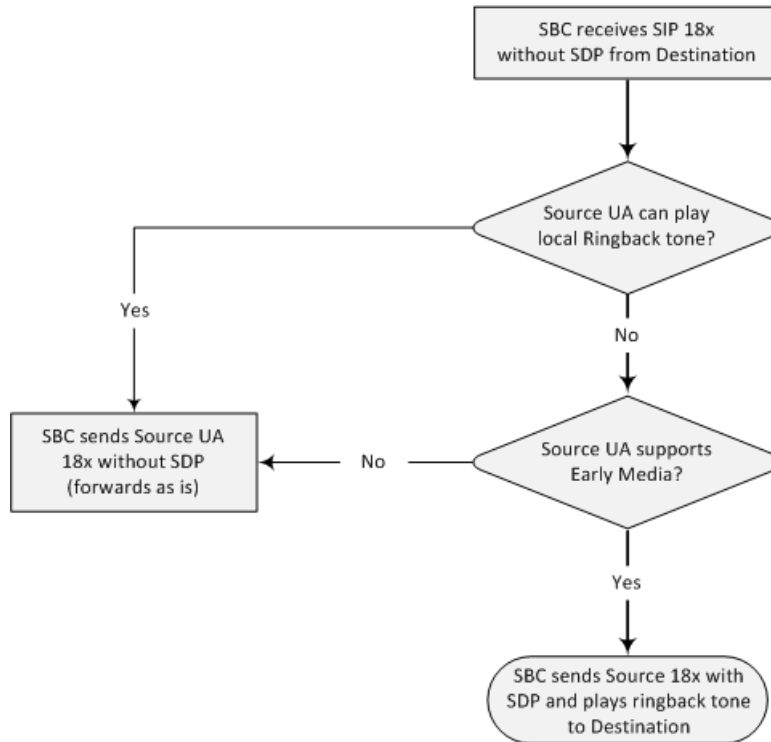
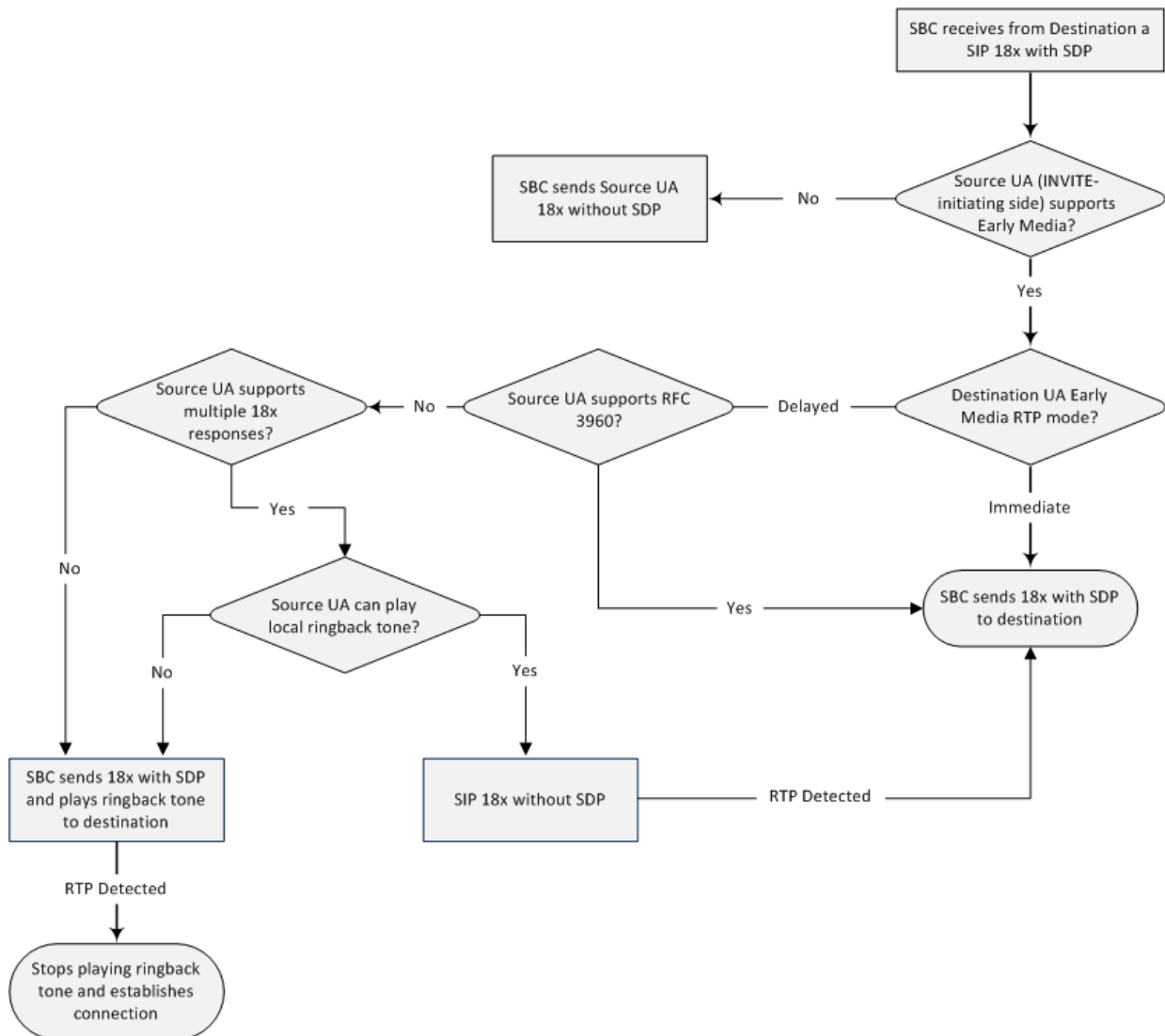


Figure 19-7: Early Media RTP - SIP 18x with SDP

19.7.7 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'.

19.7.8 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'.

19.7.9 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.

19.7.10 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'.

19.7.11 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 [275](#).

19.8 Call Survivability

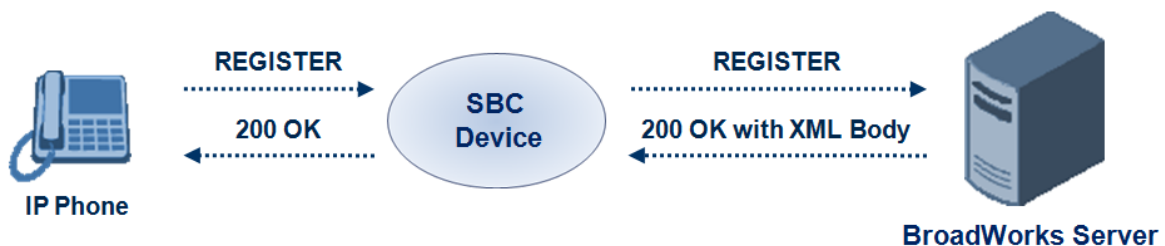
This section describes various call survivability features supported by the SBC device.

19.8.1 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 19-8: 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:

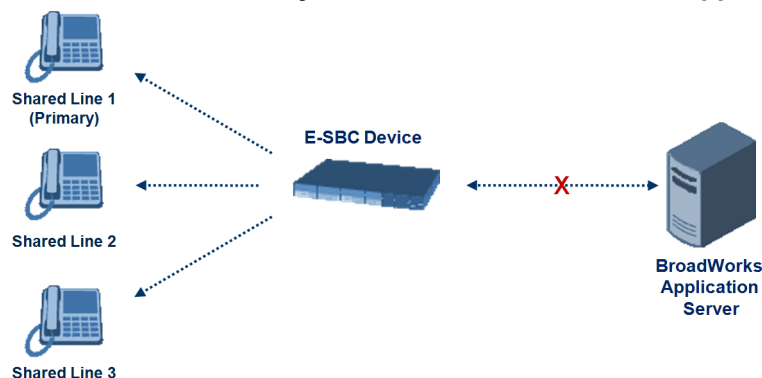
```
<?xml version="1.0" encoding="utf-8"?>
<BroadsoftDocument version="1.0" content="subscriberData">
  <phoneNumbers>
    <phoneNumber>2403645317</phoneNumber>
    <phoneNumber>4482541321</phoneNumber>
  </phoneNumbers>
  <aliases>
    <alias>sip:bob@broadsoft.com</alias>
    <alias>sip:rhughes@broadsoft.com</alias>
  </aliases>
  <extensions>
    <extension>5317</extension>
    <extension>1321</extension>
  </extensions>
</BroadsoftDocument>
```

19.8.2 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 320. Note that incoming calls specific to extensions 601 or 602 ring only at these specific extensions.

Figure 19-9: 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 following procedure 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 240), 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 340), 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 353), 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).

19.8.3 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 finds. 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 19-10: Normal Operation in Call Center Application

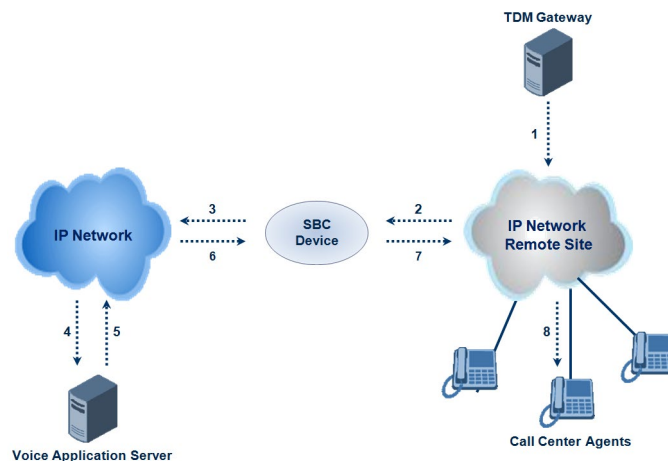
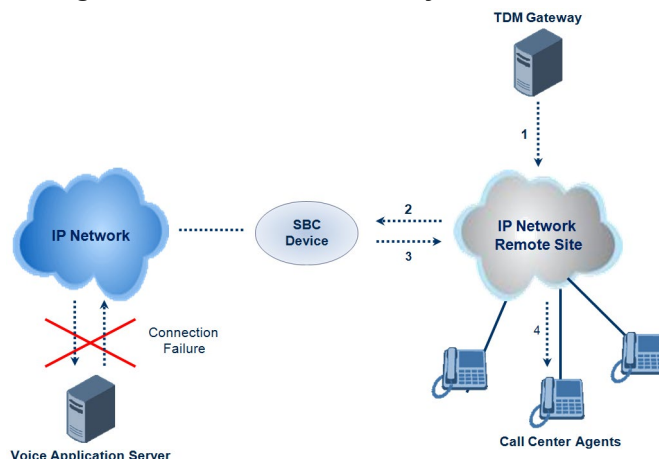


Figure 19-11: Call Survivability for Call Center



➤ **To configure call survivability for a call center application:**

1. In the IP Group table (see "Configuring IP Groups" on page 240), 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 332), 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 340), 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 19-12: 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 ▼
Submit ✕ Cancel	

19.8.4 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 `SBCEnableSurvivabilityNotice` 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>
```

19.9 Call Forking

This section describes various Call Forking features supported by the device.

19.9.1 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 240).

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.

19.9.2 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.

19.9.3 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 340.

19.10 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.

This page is intentionally left blank.

20 SBC Configuration

This chapter describes the configuration of the SBC application.

20.1 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 406.

➤ **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.

20.2 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 586.

➤ **To configure general parameters:**

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

Figure 20-1: 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 390.

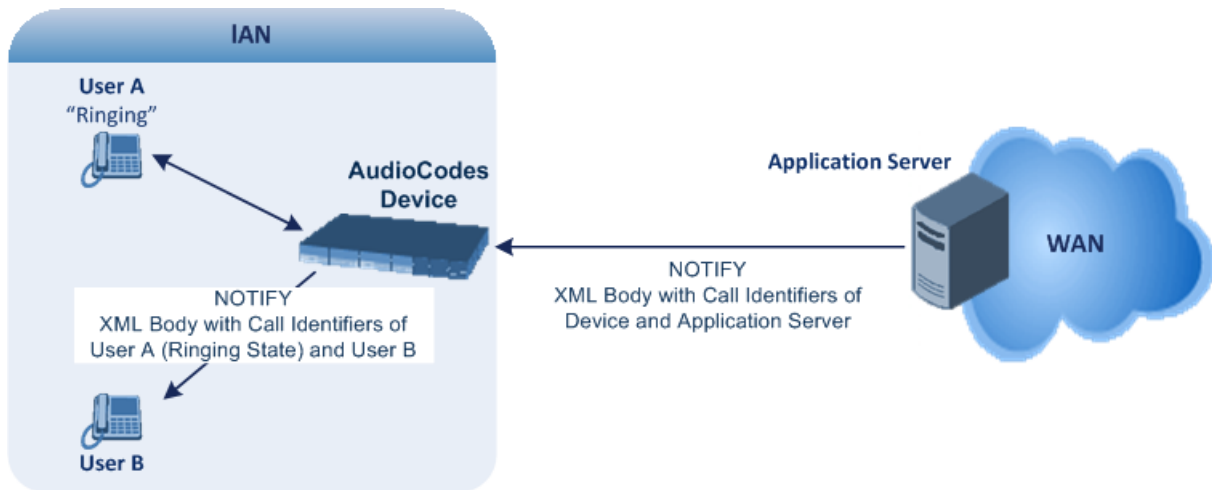
20.2.1 Interworking Dialog Information in SIP NOTIFY Messages

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 20-2: Interworking NOTIFY XML Body for 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>
```

20.3 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 (“cached 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 480 "Temporarily Unavailable" 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 following procedure 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 20-3: Admission Control Page - Add Record Dialog Box

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 20-1: 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_SRDI]	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 .
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) ▪ [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.

Parameter	Description
Rate CLI: rate [SBCAdmissionControl_Rate]	Defines the rate (in seconds) at which tokens are added to the token bucket per second (i.e., token rate). The default is 0 (i.e., unlimited rate). Notes: You must first configure the Maximum Burst parameter (see below) before configuring the Rate parameter. 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 480 "Temporarily Unavailable" 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.

20.4 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 275). 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 303.

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 304.

The following procedure 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 20-4: Allowed Audio Coders Group Page

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 20-2: Allowed Audio Coders Group Table Parameter Descriptions

Parameter	Description
Allowed Coders Group ID [AllowedCodersGroupX]	Defines an index number for the new table record.
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.

20.5 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 default video coders 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 275). 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 303.

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 304.

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 20-3: 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 default 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.

20.6 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 332
- Message Condition rules - see "Configuring Message Condition Rules" on page 339
- IP-to-IP routing rules - see "Configuring SBC IP-to-IP Routing Rules" on page 340
- Alternative routing reasons - see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 349

20.6.1 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).

Configuration of Classification rules includes two areas:

- **Rule:** Defines the matching characteristics of the incoming IP call (e.g, source SIP Interface and IP address).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (i.e., classifies the call to the specified IP Group).
- The device searches the table from top to bottom for the first rule that matches the characteristics of the incoming call. If it finds a matching rule, it classifies the call to the IP Group configured for that rule.

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 - Based on Proxy Set:** If the database search fails, the device performs classification based on Proxy Set. This classification is applicable only to Server-type IP Groups and is done only if classification based on Proxy Set is enabled (see the 'Classify By Proxy Set' parameter in the IP Group table in "Configuring IP Groups" on page 240). The device checks whether the incoming INVITE's IP address (if host name, then according to the dynamically resolved IP address list) is configured for a Proxy Set (in the Proxy Set table). If such a Proxy Set exists, the device classifies the INVITE to the IP Group that is associated with the Proxy Set. The Proxy Set is assigned to the IP Group in the IP Group table.

If classification based on Proxy Set fails (or classification based on Proxy Set is disabled), the device proceeds to classification based on the Classification table.

**Note:**

- For security, it is recommended to classify SIP dialogs based on Proxy Set only if the IP address of the Server-type IP Group is unknown. In other words, if the Proxy Set associated with the IP Group is configured with an FQDN. In such cases, the device classifies incoming SIP dialogs to the IP Group based on the DNS-resolved IP address. If the IP address is known, it is recommended to use a Classification rule instead (and disable the Classify by Proxy Set feature), where the rule is configured with not only the IP address, but also with SIP message characteristics to increase the strictness of the classification process. The reason for preferring classification based on Proxy Set when the IP address is unknown is that IP address forgery (commonly known as IP spoofing) is more difficult than malicious SIP message tampering and therefore, using a Classification rule without an IP address offers a weaker form of security. When classification is based on Proxy Set, the Classification table for the specific IP Group is ignored.
- If multiple IP Groups are associated with the same Proxy Set, use Classification rules to classify the incoming dialogs to the IP Groups (do not use the Classify by Proxy Set feature).

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:

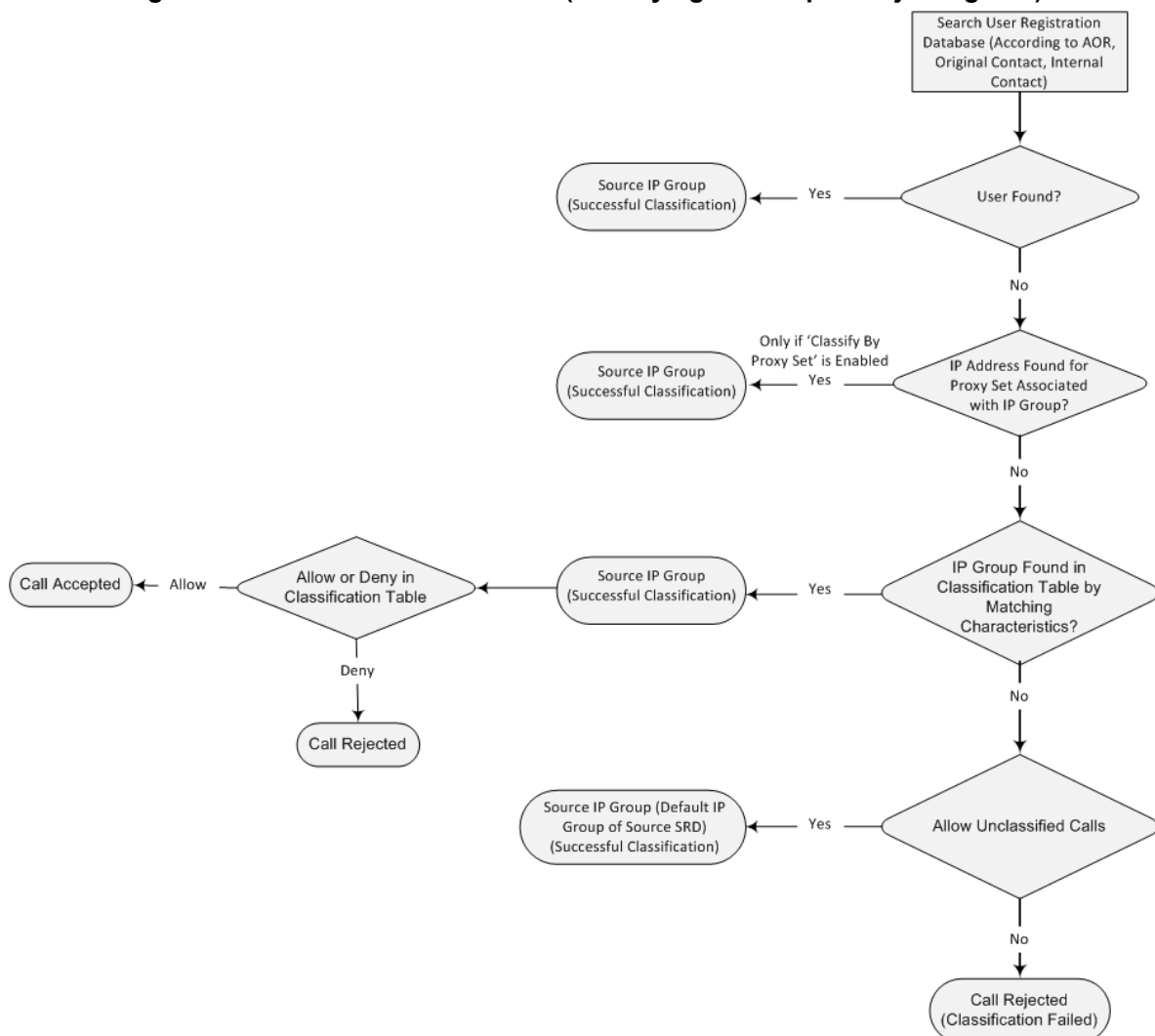
1. The device checks on which SIP listening port (e.g., 5061) the incoming SIP dialog request arrived and the SIP Interface which is configured with this port (in the SIP Interface table).
2. The device checks the SRD that is associated with this SIP Interface (in the SIP Interface table) and then classifies 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 [236](#)).

The flowchart below illustrates the classification process:

Figure 20-5: Classification Process (Identifying IP Group or Rejecting Call)



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 following procedure 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 20-6: Classification Table Page

The screenshot shows a configuration dialog box for a Classification rule. It has a header with 'Rule' and 'Action' tabs. Below the header are several input fields: 'Index' with a value of 0, 'Classification Name' (empty), 'Message Condition' with a dropdown set to 'None', 'Source SRD ID' with a dropdown set to 'None', 'Source IP Address' (empty), 'Source Port' with a value of 0, 'Source Transport Type' with a dropdown set to 'ANY', 'Source Username Prefix' with a value of '*', 'Source Host' with a value of '*', 'Destination Username Prefix' with a value of '*', and 'Destination Host' with a value of '*'. At the bottom right are 'Submit' and 'Cancel' buttons.

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 20-4: 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 339.
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 233. 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 236).

Parameter	Description
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: ▪ "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. Note: ▪ The parameter is applicable only to Server-type IP Groups. ▪ If the IP address is unknown (i.e., configured for the associated Proxy Set as an FQDN), it is recommended to classify incoming dialogs based on Proxy Set (instead of using a Classification rule). For more information on classification by Proxy Set or by Classification rule, see the note bulletin in the beginning of this section.
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 294. 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 501. 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 294. The default is the asterisk (*) symbol, which represents any

Parameter	Description
	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 501.
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	
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 240. By default, no IP Group is defined. Note: The IP Group must be associated with the assigned SRD.

20.6.1.1 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=YSQQKXXREVPYPTNFMWG
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>,<sip:3000@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

2. 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>"), respectively. These SIP headers were determined in IP Group ID 2.

20.6.2 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 332)
- IP-to-IP routing rules in the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 340)
- IP-to-IP outbound manipulation rules in the IP to IP Outbound Manipulation table (see "Configuring IP-to-IP Outbound Manipulations" on page 356)

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 266). 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 following procedure 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 20-7: Condition Table Page - Add Record Dialog Box

The dialog box titled "Add Record" contains three input fields: "Index" with the value "0", "Condition" with the value "param.ipg.src.type==user", and "Description" with the value "IP Group USER". At the bottom right are "Submit" and "Cancel" buttons.

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 20-8: 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 20-5: 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.

20.6.3 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. Configuration of IP-to-IP routing rules includes two areas:

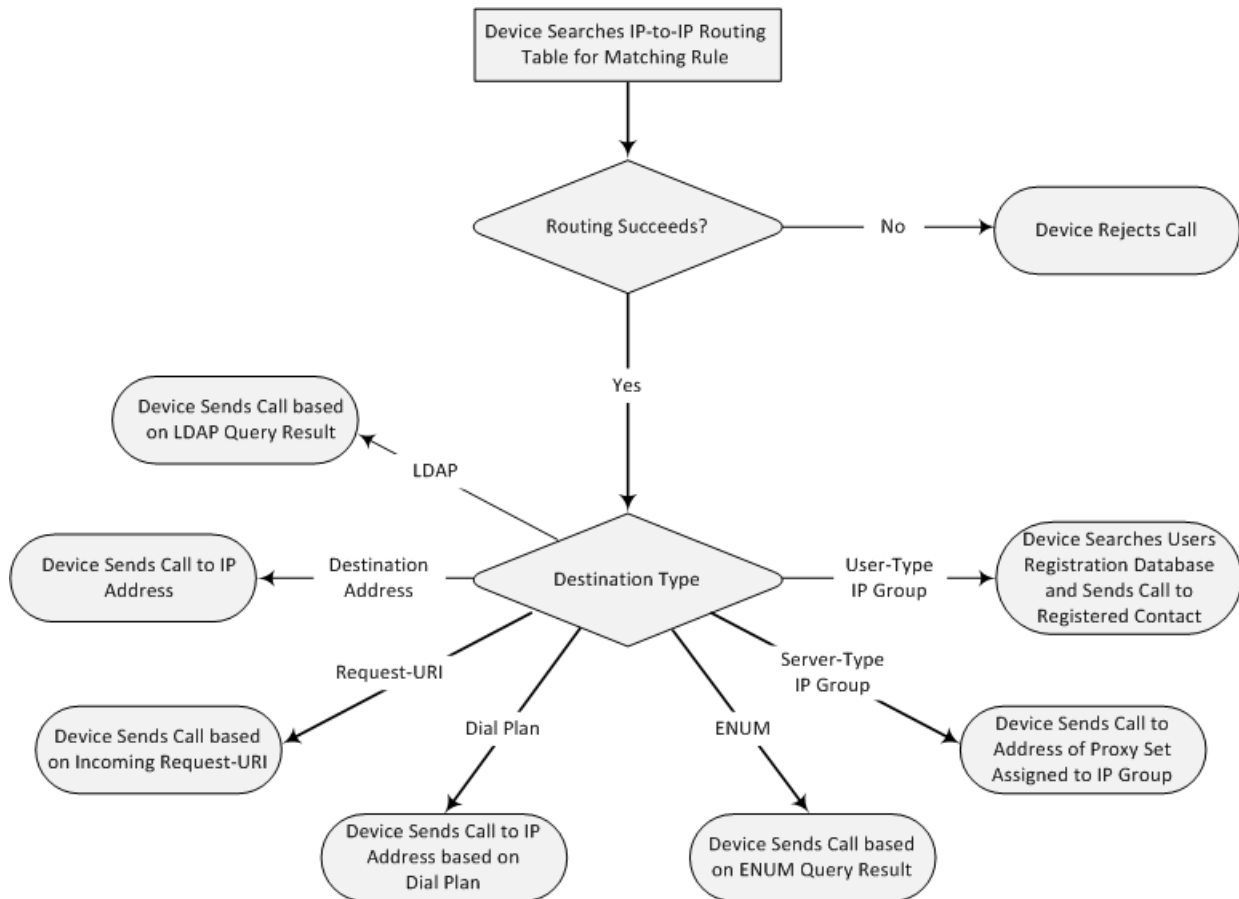
- **Rule:** Defines the characteristics of the incoming SIP dialog message (e.g., IP Group from which the message is received).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (i.e., routes the call to the specified destination).

The device searches the table from top to bottom for the first rule that matches the characteristics of the incoming call. If it finds a matching rule, it sends the call to the destination configured for that rule. If it doesn't find a matching rule, it rejects the call.

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 317).
- 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 174).

Figure 20-9: IP-to-IP Routing Destination Types



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 349).
 - ◆ 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).



Note: If the Proxy Set (see Configuring Proxy Sets on page 251) associated with the destination of the call is configured with multiple IP addresses, the device first attempts to route the call to one of these IP addresses, starting with the first listed address. Only when the call cannot be routed to any of the Proxy Set's IP addresses does the device search the IP-to-IP Routing table for an alternative routing rule for the call.

- **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 175. 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 204).
- **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 400.



Note: Call forking is not applicable to LDAP-based IP-to-IP routing rules.

The following procedure 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**).

- Click **Add**; the following dialog box appears:

Figure 20-10: IP-to-IP Routing Table - Add Record Dialog Box

Rule	Action
Index	1
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
Call Setup Rules Set ID	-1
Submit Cancel	

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

Table 20-6: 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 332). The default is -1. To denote any IP Group, leave this field empty.
Source Username Prefix [IP2IPRouting_SrcUsername Prefix] 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 501. 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	Defines the prefix of the incoming SIP dialog's destination URI

Parameter	Description
[IP2IPRouting_DestUsernamePrefix] CLI: dst-user-name-prefix	(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 501. 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 339.
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 307 and "Interworking SIP REFER Messages" on page 310, 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 forwards to the destination. This rule is not used for re-routing of requests triggered by the receipt of REFER or 3xx. ▪ [5] Broken Connection = If the device detects a broken RTP connection during the call and the Broken RTP Connection feature is enabled (IpProfile_DisconnectOnBrokenConnection parameter is configured to [2]), you can use this option as an explicit matching characteristics to route the call to an alternative destination. Therefore, for alternative routing upon broken RTP detection, position the routing rule configured with this option above the regular routing rule associated with the call. Such a

Parameter	Description
	configuration setup ensures that the device uses this alternative routing rule only when RTP broken connection is detected.
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 209.
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 317. ▪ [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: <pre>[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</pre> 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

Parameter	Description
	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 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 240). 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 127). - 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_DestTranspo	Defines the transport layer type for sending the call: [-1] Not Configured (default)

Parameter	Description
rtType] CLI: dst-transport-type	▪ [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 (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 349). 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). Note: Each Forking Group can contain up to 20 members. In other words, up to 20 routing rules can be configured for the same Forking Group.

Parameter	Description
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 205. By default, no Cost Group is defined.

20.6.4 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 does alternative routing as follows: If the destination Proxy Set is configured with multiple IP addresses (see Configuring Proxy Sets on page 251), the device first attempts to route the call to one of these IP addresses, starting with the first listed address. If unsuccessful, the device then searches for an alternative routing rule in the IP-to-IP Routing table (see 'Configuring SBC IP-to-IP Routing Rules' on page 340).

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, if configured in the table, which 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 326). 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 480 "Temporarily Unavailable" response.

The following procedure 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 20-11: Alternative Routing Reasons Table - Add Record

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 20-7: 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.

20.7 SBC Manipulations

This section describes the configuration of the manipulation rules for the SBC application.

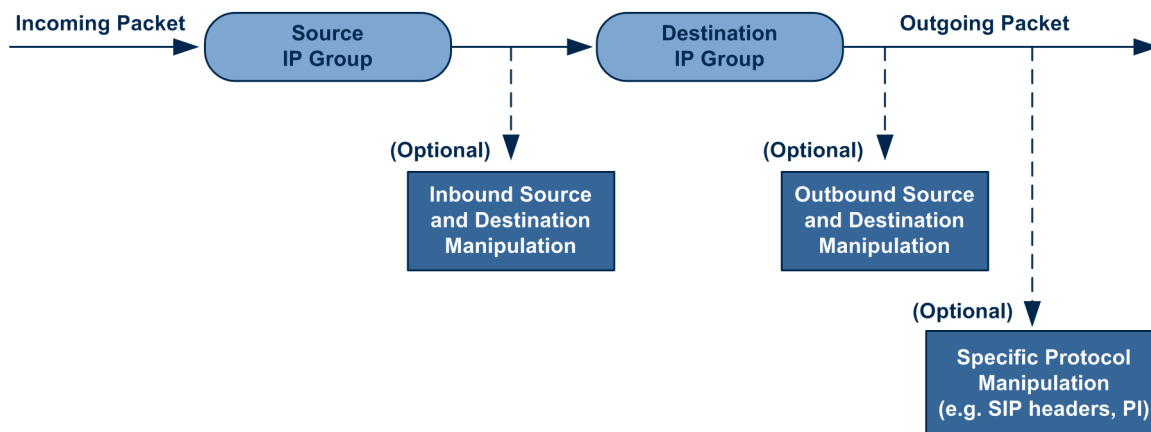


Note: For additional manipulation features, see the following:

- "Configuring SIP Message Policy Rules".
- "Configuring SIP Message Manipulation" on page 266.

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 20-12: SIP URI Manipulation in IP-to-IP Routing



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=01LAN;parameter1=abe
To: <sip:1000@10.2.2.3;user=phone>
Call-ID: USELLLLAN@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;parameter1=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=01LAN;parameter1=abe
```

to

```
From:
<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;parameter1=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=01LAN;parameter1=abe
```

to

```
From:
<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;parameter1=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>
```

20.7.1 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 240).

The following procedure 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 20-13: IP to IP Inbound Manipulation Page - Add Dialog Box

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 20-8: 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

Parameter	Description
tionPurpose]	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 316.
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 501.
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_DestinationUsernamePrefix]	Defines the prefix of the destination SIP URI user name, typically located in the Request-URI and To headers. 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 501.
Destination Host CLI: dst-host [IPInboundManipulation_DestinationHost]	Defines the destination SIP URI host name - full name, typically located in the Request-URI and To headers. 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

Parameter	Description
tedURI]	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".

20.7.2 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 following procedure 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 20-14: IP to IP Outbound Manipulation Page - Add Dialog Box

Rule	Action
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

Submit Cancel

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 20-9: IP to IP Outbound Manipulation Table Parameter Description

Parameter	Description
Index [IPOutboundManipulation_Index]	Defines an index number for the new table record.
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.

Parameter	Description
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 value 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 value 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 501.
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 located in the Request-URI and To headers. 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 501.
Destination Host CLI: dst-host [IPOutboundManipulation_DestHost]	Defines the destination SIP URI host name - full name, typically located in the Request-URI and To headers. 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_MessageCondition]	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 page 339.
Request Type	Defines the SIP request type to which the manipulation rule is

Parameter	Description
CLI: request-type [IPOutboundManipulation_RequestType]	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 307 and "Interworking SIP REFER Messages" on page 310, 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".

Parameter	Description
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.
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]	Defines 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 in the outgoing SIP dialog 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 restriction presentation is 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". ▪ [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). Note: ▪ Restriction is done only after user number manipulation (if any). ▪ The device identifies an incoming user as restricted if one of the following exists:

Parameter	Description
	✓ From header user is "anonymous". ✓ P-Asserted-Identity and Privacy headers contain the value "id".

This page is intentionally left blank.

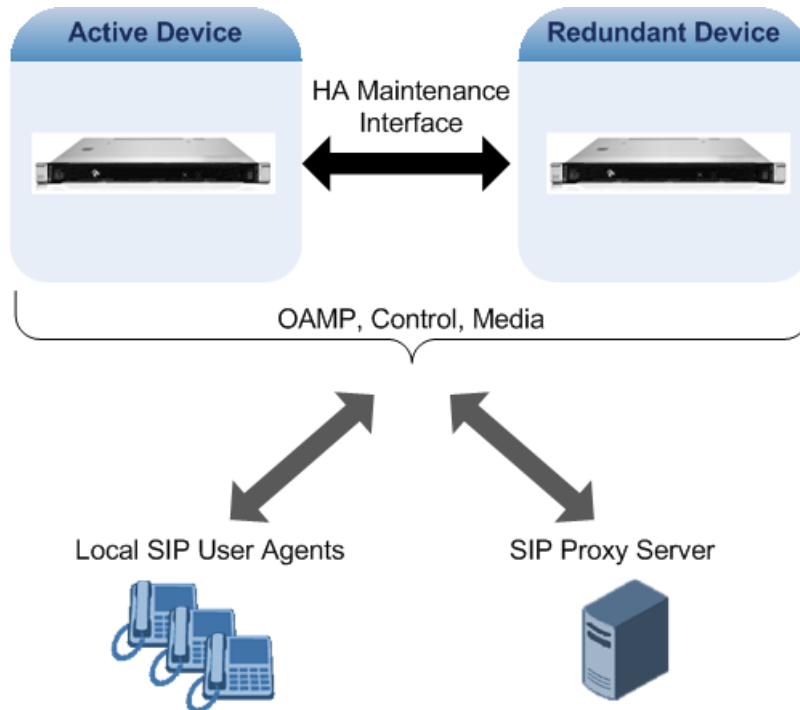
Part VI

High Availability System

21 HA Overview

The device's High Availability (HA) feature provides 1+1 system redundancy using two 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.



21.1 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.

21.2 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.



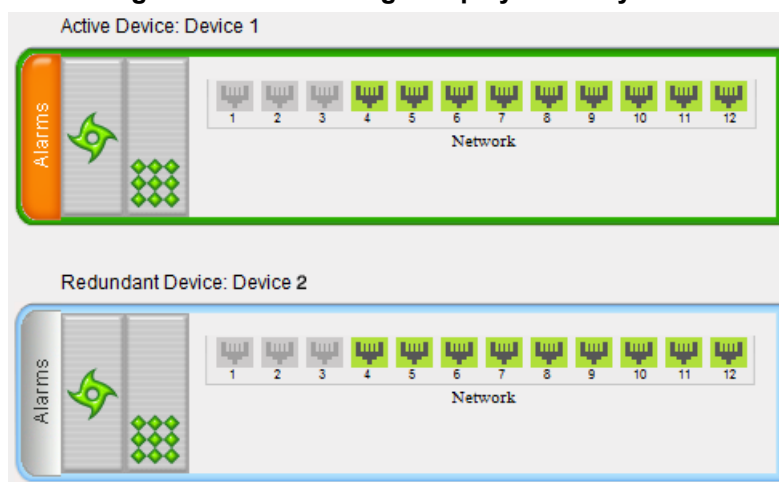
Notes:

- 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.

21.3 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 21-1: 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)

22 HA Configuration

This section describes the configuration of the HA system.

22.1 Initial HA Configuration

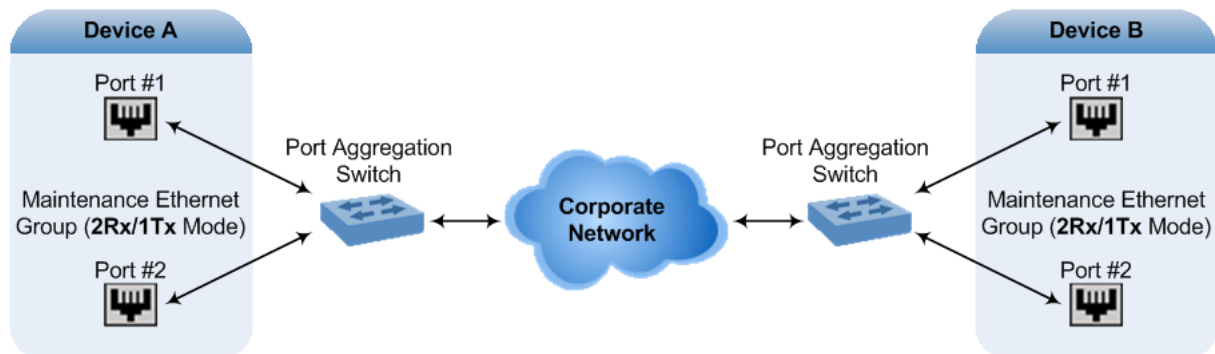
By default, HA is disabled on the device. When a device is loaded with valid HA configuration and it is the first device to be loaded, it becomes the active device. The second device that is loaded with HA configuration becomes the redundant (standby) device.

22.1.1 Network Topology Types and Rx/Tx 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:

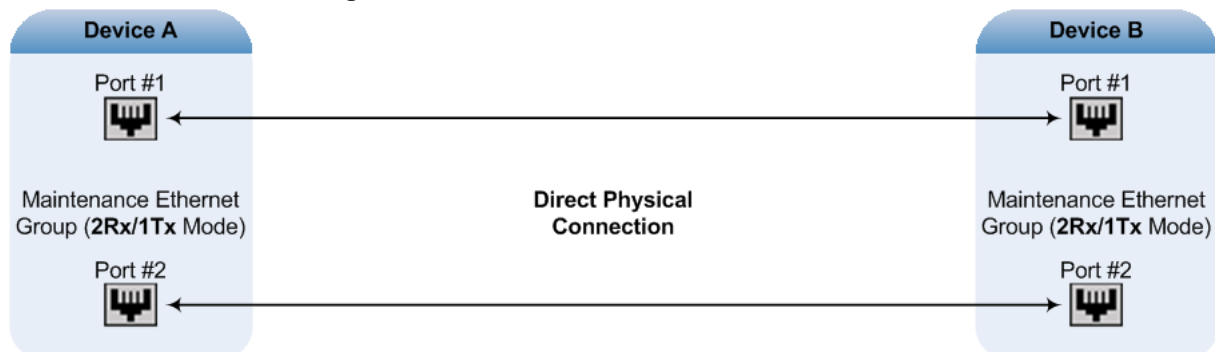
- For Geographical HA (both units are located far from each other), **2Rx/1Tx** port mode connected to a port aggregation switch is the recommended option:

Figure 22-1: Rx/Tx Mode for Geographical HA



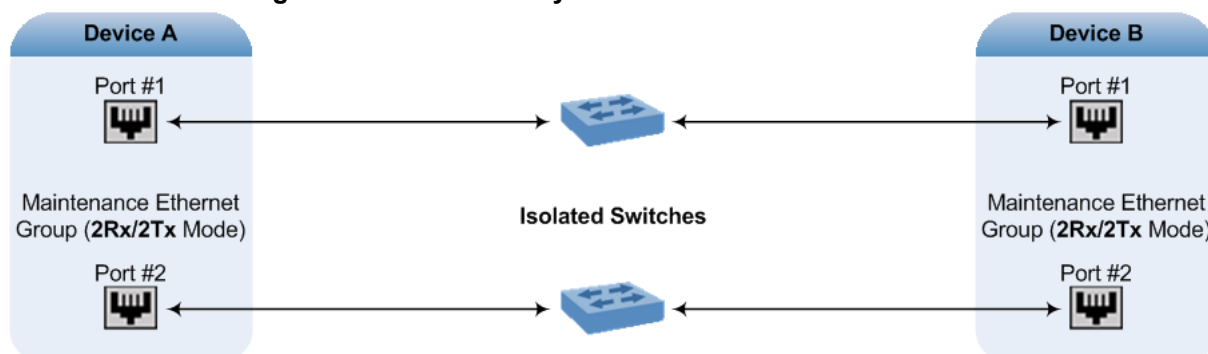
- If the Maintenance ports of both devices are connected directly to each other without intermediation of switches, configure the mode to **2RX/1TX**:

Figure 22-2: Rx/Tx Mode for Direct Connection



- 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 22-3: 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 Rx/Tx modes of the Ethernet ports, see "Network Topology Types and Rx/Tx Ethernet Port Group Settings" on page 369.

22.1.2 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 371
2. Configuring the second device for HA - see "Step 2: Configure the Second Device" on page 373
3. Activating HA on the devices - see "Step 3: Initialize HA on the Devices" on page 373



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 406.
- 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 Rx/Tx Ethernet Port Group Settings" on page 369.
- 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.

22.1.2.1 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 22-4: 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 105). 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 22-5: 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 Ethernet Port Groups" on page 107). The port mode depends on the type of Maintenance connection between the devices, as described in "Network Topology Types and Rx/Tx Ethernet Port Group Settings" on page 369.
4. Configure the HA parameters in the HA Settings page (**Configuration** tab > **System** menu > **HA Settings**):

Figure 22-6: HA Settings Page

HA Settings	
HA Remote Address	10.3.4.61
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 373 for configuring the second device.

22.1.2.2 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 371).



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 373 for completing the HA configuration.

22.1.2.3 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 374.

22.2 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).

22.3 Configuring Firewall Allowed Rules

If you add firewall rules in the Firewall Settings page (see "Configuring Firewall Settings" on page 139) 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 22-7: 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

22.4 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 540.

➤ To configure monitoring of IP entity using pings:

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

Figure 22-8: 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 Interface 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).

This page is intentionally left blank.

23 HA Maintenance

This section describes HA maintenance procedures.

23.1 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).

23.2 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 370.

23.3 Forcing a Switchover

If required, you can force a switchover between active and redundant SBCs. For more information, see "High Availability Maintenance" on page 391.

23.4 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 410.

23.5 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.

23.5.1 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>
```

23.5.2 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
```

23.5.3 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 384) and/or Manual Recovery (see "Manual Recovery" on page 381). 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
```

23.5.4 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
```

23.5.5 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:

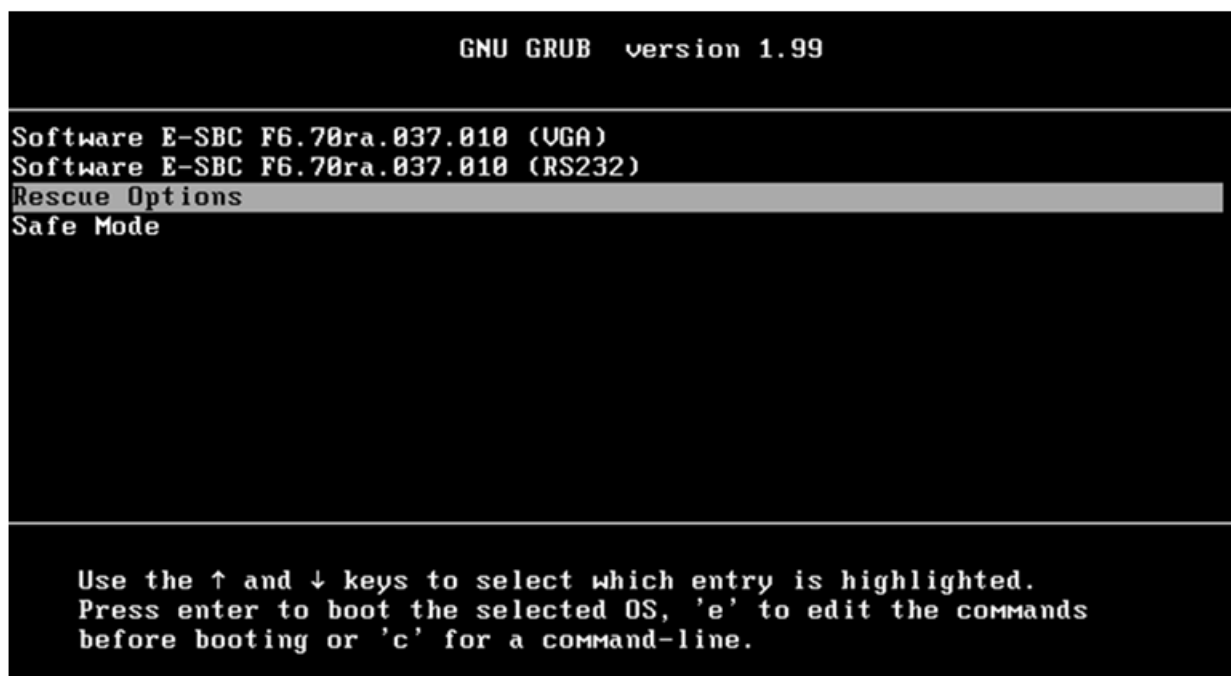
- Return to default snapshot
- Fix current installation
- Browse available system snapshots
- Return to factory snapshot (after install from CD)

23.5.5.1 Returning to the Default Snapshot

➤ To return to the default snapshot:

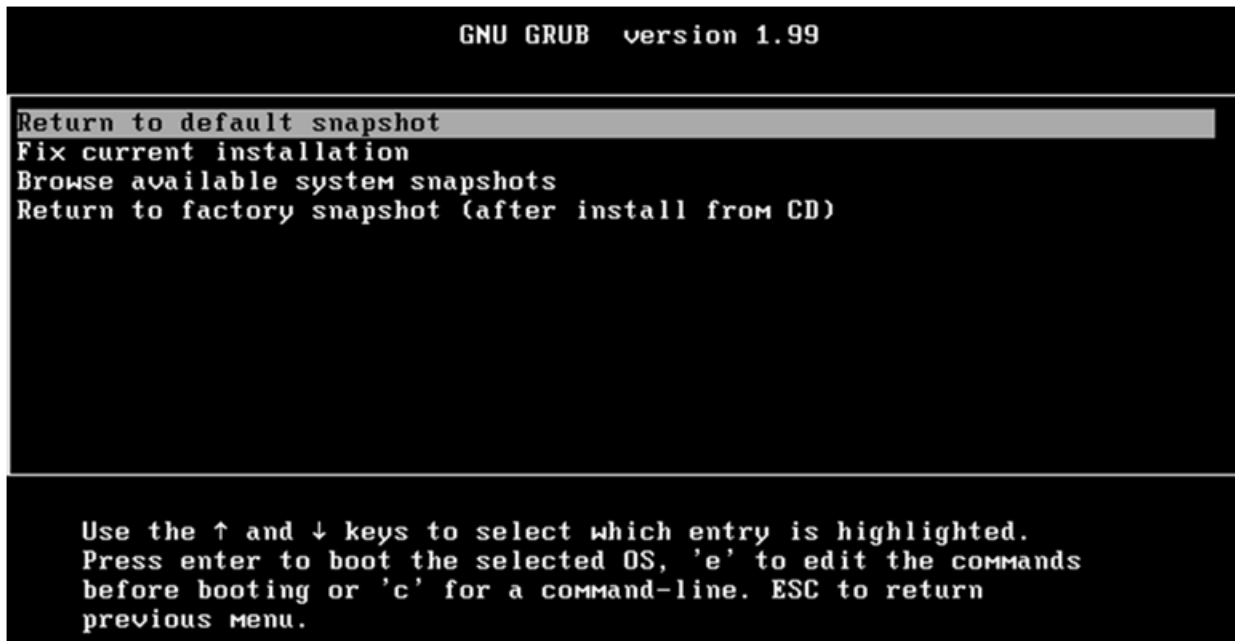
1. Reboot the server.
2. In the GRUB menu that's displayed for 5 seconds during the server start-up, press the Down ↓ key, select **Rescue option**, and then press Enter.

Figure 23-1: Main GRUB Menu



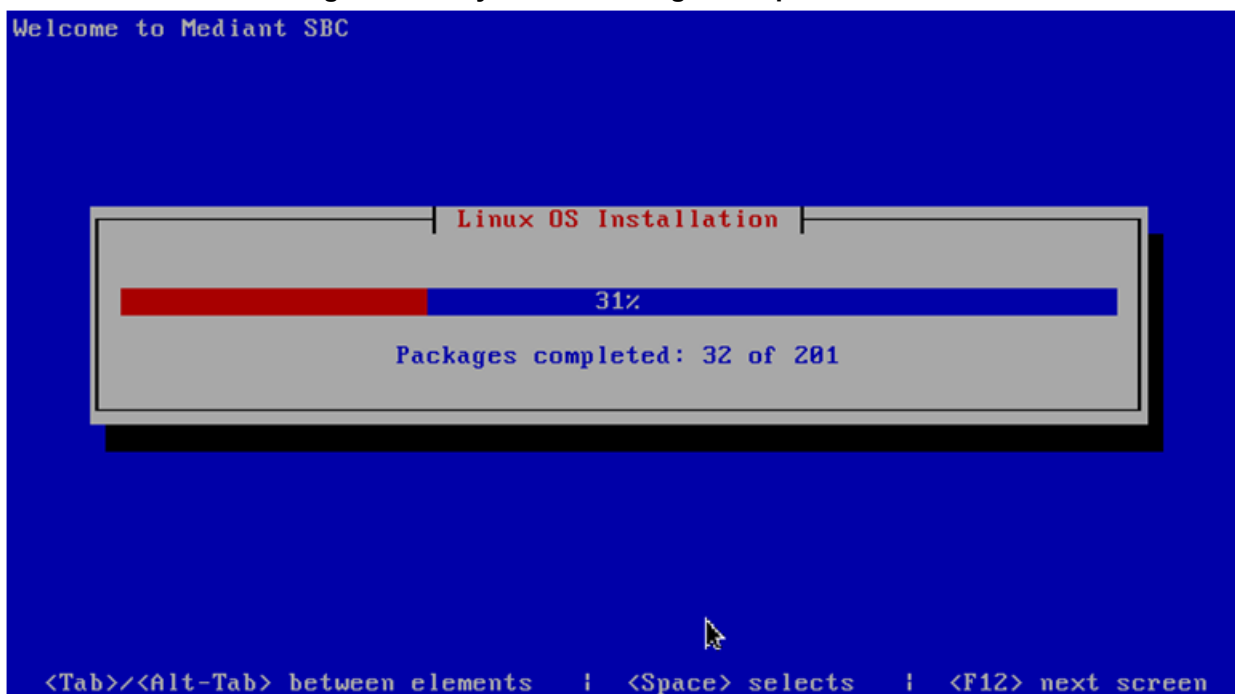
3. In the Rescue Options menu, select **Return to default snapshot**, and then press Enter.

Figure 23-2: Rescue Options 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 23-3: System Returning to Snapshot State



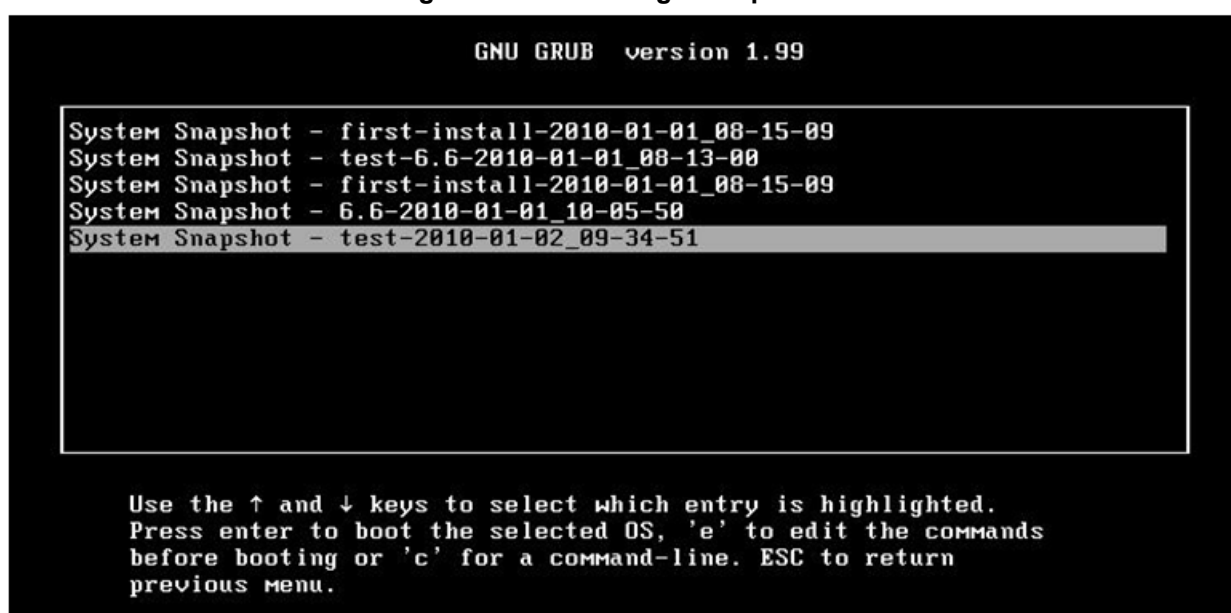
23.5.5.2 Fixing the Current Installation

- **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.

23.5.5.3 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 23-4: Selecting a Snapshot



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.

23.5.5.4 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.

23.5.6 Automatic Recovery

The device 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

24 Basic Maintenance

The Maintenance Actions page allows you to perform the following:

- Reset the device - see "Resetting the Device" on page 387
- Lock and unlock the device - see "Locking and Unlocking the Device" on page 389
- Save configuration to the device's flash memory - see "Saving Configuration" on page 390

➤ 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 24-1: Maintenance Actions Page

▼ Reset Configuration	
Reset Board	Reset
Burn To FLASH	Yes v
Graceful Option	No v
▼ LOCK / UNLOCK	
Lock	LOCK
Graceful Option	No v
Current Admin State	UNLOCKED
▼ Save Configuration	
Burn To FLASH	BURN

24.1 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).



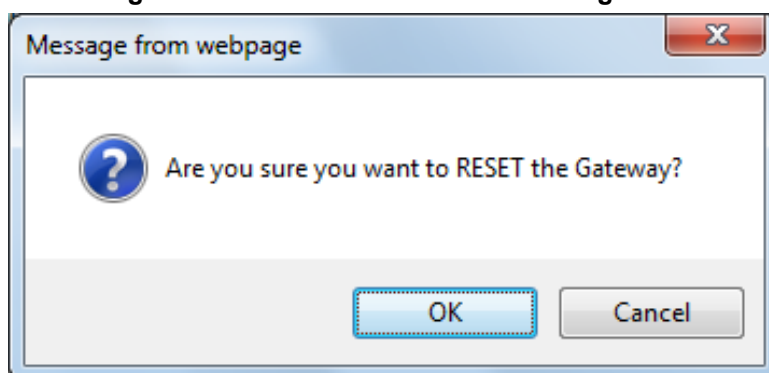
Notes:

- 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 32) 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 33).

➤ **To reset the device:**

1. Open the Maintenance Actions page (see "Basic Maintenance" on page 387).
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 24-2: Reset Confirmation Message Box



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.

24.2 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.

24.3 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 387).
2. Scroll down to the 'LOCK / UNLOCK' group:

Figure 24-3: Locking the Device

LOCK / UNLOCK	
Lock	LOCK
Graceful Option	Yes v
Lock Timeout [sec]	20 v
Gateway Operational State	UNLOCKED

3. From the 'Graceful Option' drop-down list, select one of the following options:

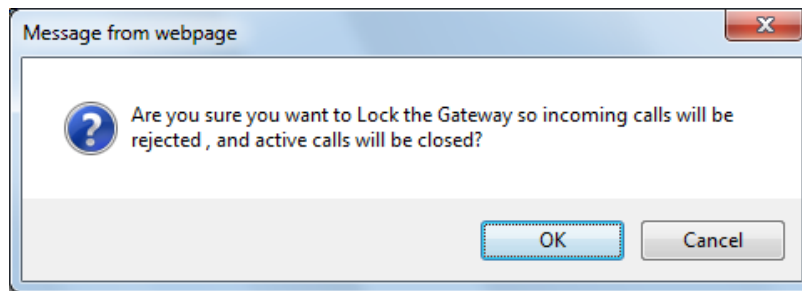
- **Yes:** The device is locked only after the user-defined time in the 'Lock Timeout' field (see Step 3) expires or no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
- **No:** The device is locked regardless of traffic. Any existing traffic is terminated immediately.

Note: These options are only available if the current status of the device is in UNLOCKED state.

4. If you set 'Graceful Option' to **Yes** (in the previous step), then in the 'Lock Timeout' field, enter the time (in seconds) after which the device locks. If no traffic exists and the time has not yet expired, the device locks immediately.

5. Click the **LOCK** button; a confirmation message box appears requesting you to confirm device lock.

Figure 24-4: Device Lock Confirmation Message Box



6. Click **OK** to confirm device lock; if you set 'Graceful Option' to **Yes**, a lock icon is delayed and a window appears displaying the number of remaining calls and time. If you set 'Graceful Option' to **No**, the lock process begins immediately. The 'Gateway Operational State' field displays "LOCKED".

➤ **To unlock the device:**

- Under the 'LOCK / UNLOCK' group, click the **UNLOCK** button. Unlock starts immediately and the device accepts new incoming calls. The 'Gateway Operational State' field displays "UNLOCKED".



Note: The Home page's General Information pane displays whether the device is locked or unlocked (see "Viewing the Home Page" on page 44).

24.4 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 387).
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 389).
- 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 387).
- The Home page's General Information pane displays whether the device is currently "burning" the configuration (see "Viewing the Home Page" on page 44).

25 High Availability Maintenance

This section describes various maintenance procedures for the High Availability mode.

25.1 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 25-1: 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**.

25.2 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 25-2: 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**.

26 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>
```

This page is intentionally left blank.

27 Software Upgrade

This chapter describes various software update procedures.

27.1 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 27-1: 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 77.
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 396.
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 399.
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 399.
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 402.

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 389.
- For deleting auxiliary files, see "Viewing Device Information" on page 437.

The following procedure 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 390 and reset the device (if you have loaded a Call Progress Tones file), see "Resetting the Device" on page 387.

27.1.1 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
```

27.1.2 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.

**Notes:**

- The PRT file only generates (plays) tones; detection of tones is according to the CPT file.
- Local generation of tones is not supported.
- The PRT file supports only calls that use the G.711 coder.
- The PRT file supports only the ringback tone and hold tone.

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 third-party recording utilities 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 395).

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.

27.1.3 Dial Plan File

The Dial Plan file can be used for various digit mapping features, as described in this section.

27.1.3.1 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 395.
5. Select the Dial Plan index that you want to use. This depends on the feature and is described in the respective section.

27.1.3.2 Dial Plan Prefix Tags for Routing

27.1.3.2.1 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 340) 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 following procedure 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.

27.1.3.3 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 399).
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.

27.1.4 User Information File

This section describes the User Info table and how to configure the table.

27.1.4.1 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.

27.1.4.2 User Information File for SBC User Database

You 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.

You can configure up to 3,000 users (table rows) in the SBC User Info table. 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 403
- CLI - see Configuring SBC User Info Table in CLI on page 404
- Loadable User Info file - see "Configuring SBC User Info Table in Loadable Text File" on page 405

27.1.4.2.1 Configuring SBC User Info Table in Web Interface

The following procedure 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 27-1: SBC User Info Table Page

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 390.

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 27-2: SBC User Info Table Parameter Descriptions

Parameter	Description
Index [SBCUserInfoTable_Index]	Defines an index for the new table record.
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.

Parameter	Description
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".

27.1.4.2.2 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
```

27.1.4.2.3 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 395
- *ini* file, using the `UserInfoFileName` parameter - see "Auxiliary and Configuration File Name Parameters" on page 517
- 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.

27.2 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:

- 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.

27.2.1 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 27-2: Software License Key File with Multiple S/N Lines

```
sample.ini - Notepad
File Edit Format Help
[LicenseKeys]
;Board Type 29
S/N241182 =
okRTr5topwYmblZd4NN2a3Qhm4NjfiDaagUyehso94APbBF85hF4by0cmQZf2B8bMcze7JQ9kMSa5h641R1aOkeEb9AddF894Zx
S/N242519 = tmxTr5to0mblZdoPd2a3Qh9zJlfiDafilyehsogOQPbBF8pi4by0c9jdf2B8eOoze7JQgywSa5h6o391aOkeTlIAAddF8c6Fx
S/N226403 = tmxTr5to0lsmblZdoOB2a3Qh9yJlfiDafilyehsogN4PbBF8piZ4by0c9jdf2B8eOoze7JQgpgSa5h6o2x1aOkeTlIAAddF8c6Fx
S/N226417 = r6xTr5to25smbZdflB2a3Qh5OJlfiDa92yehsoix4PbBF8eOZ4by0c52df2B88yoze7JQlNgSa5h6tyx1aOkeXZlIAAddF8amFx
;Board Type 24
S/N241182 =
okRTr5topwYmblZd4NN2a3wkm4NjfiDaagUyehso94APbBF85hF4by0cmQZf2B8bMcze7JQ9kMSa5h641R1aOkeEb9AddF8938s
S/N242519 = tmxTr5to0mblZdoPd2a3wK9zJlfiDafilyehsogOQPbBF8pi4by0c9jdf2B8eOoze7JQgywSa5h6o391aOkeTlIAAddF8c1ss
S/N226403 = tmxTr5to0lsmblZdoOB2a3wK9yJlfiDafilyehsogN4PbBF8piZ4by0c9jdf2B8eOoze7JQgpgSa5h6o2x1aOkeTlIAAddF8c1ss
S/N226417 = r6xTr5to25smbZdflB2a3wK5OJlfiDa92yehsoix4PbBF8eOZ4by0c52df2B88yoze7JQlNgSa5h6tyx1aOkeXZlIAAddF8ahss
```

➤ **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.
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 408.

27.2.2 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 408
- CLI - see Installing Software License Key using CLI on page 409



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.

27.2.2.1 Installing Software License Key using Web Interface

The following procedure 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**).

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 reload 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.

27.2.2.2 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 <"string enclosed in double quotation marks">
```

- To view the Software License Key:

```
show system feature-key
```

27.3 Software Upgrade Wizard

The Web interface's Software Upgrade Wizard lets you easily upgrade the device's software version (.cmp file). The wizard also provides you the option to load other files such as an ini file and auxiliary files (e.g., Call Progress Tone / CPT file). However, loading a .cmp file is mandatory through the wizard and before you can load any other type of file, the .cmp file must be loaded.

The wizard can also upgrade devices set up in High Availability (HA) mode. You can choose between two optional HA upgrade methods:

- **System Reset Upgrade (non-Hitless):** Both the active and redundant devices are upgraded simultaneously. Therefore, this method is traffic-affecting and terminates current calls during the upgrade process. The process is as follows:
 1. The active (current) device loads the .cmp file.
 2. The active device sends the .cmp file to the redundant device.
 3. Both active and redundant devices install and burn the file to flash memory with a reset. In other words, no HA switchover occurs.
- **Hitless Upgrade:** The devices are upgraded without disrupting traffic (i.e., current calls are maintained). The process is as follows:
 1. The active (current) device loads the .cmp file.
 2. The active device sends the .cmp file to the redundant device.
 3. The redundant device installs and burns the file to its flash memory with a reset. The redundant device now runs the new software version.
 4. An HA switchover occurs from the active to redundant device. Therefore, current calls are maintained and now processed by the previously redundant device, which is now the active device.
 5. The previously active device (now in redundant mode) installs and burns the file to its flash memory with a reset. Therefore, both devices now run the new software version.
 6. An HA switchover occurs from the active device (i.e., the initial redundant device) to the redundant device (i.e., the initial active device) to return the devices to their original HA state. Only the initial redundant device undergoes a reset to return to redundant state.

**Notes:**

- You can obtain the latest software files from AudioCodes Web site at <https://www.audiocodes.com/library/firmware/>.
- 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 you start the wizard, the rest of the Web interface is unavailable. After the files are successfully installed with a device reset, access to the full Web interface is restored.
- If you upgraded your firmware (.cmp file) and the "SW version mismatch" message appears in the Syslog or Web interface, your Software License Key does not support the new .cmp file version. If this occurs, contact AudioCodes support for assistance.
- Instead of manually upgrading the device, you can use the device's Automatic Update feature for automatic provisioning (see Automatic Provisioning on page 417).

The following procedure describes how to load files using the Web interface's Software Upgrade Wizard. Alternatively, you can load files using the CLI:

■ cmp file:

```
copy firmware from <URL>
```

■ ini or auxiliary file:

```
copy <ini file or auxiliary file> from <URL>
```

■ HA devices:

• Hitless Software Upgrade:

```
# copy firmware from <URL and file name>
```

• Non-Hitless Software Upgrade:

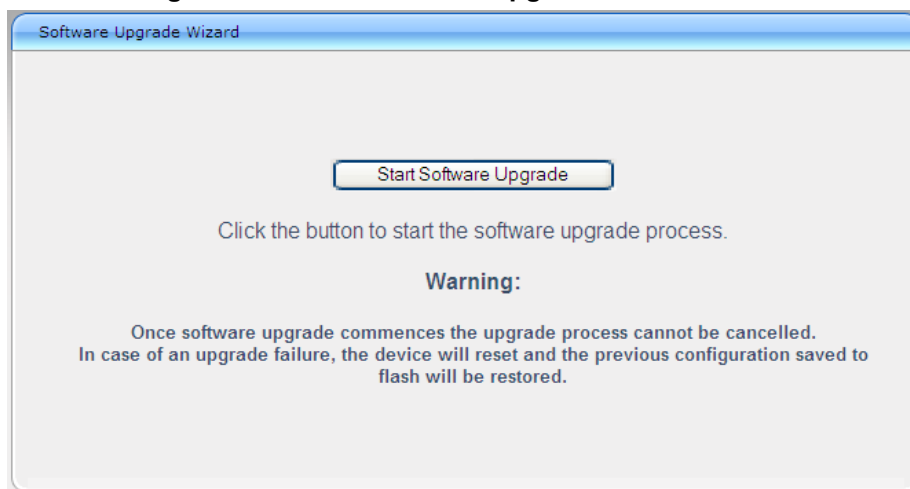
```
# copy firmware from <URL and file name> non-hitless
```

➤ **To load files using the Software Upgrade Wizard:**

1. Make sure that you have installed a new Software License Key (see Software License Key on page 406) that is compatible with the software version to be installed.
2. It is recommended to enable the Graceful Lock feature (see Locking and Unlocking the Device on page 389). The wizard resets the device at the end of the upgrade process, thereby causing current calls to be untimely terminated. To minimize this traffic disruption, the Graceful Lock feature prevents the establishment of new calls.
3. It is recommended to save a copy of the device's configuration to your computer. If an upgrade failure occurs, you can 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 415).
4. 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**.

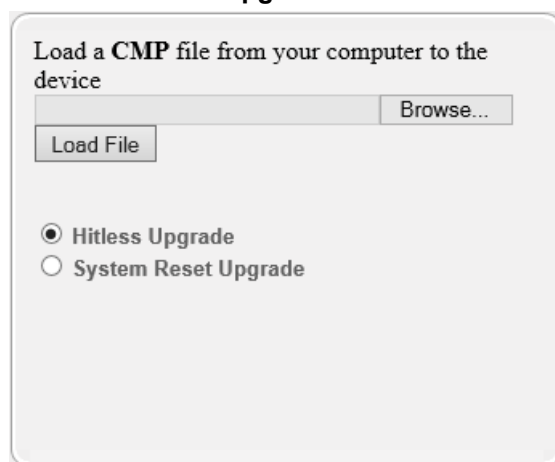
- On the toolbar, click **Device Actions**, and then choose **Software Upgrade Wizard**.

Figure 27-3: Start Software Upgrade Wizard Screen



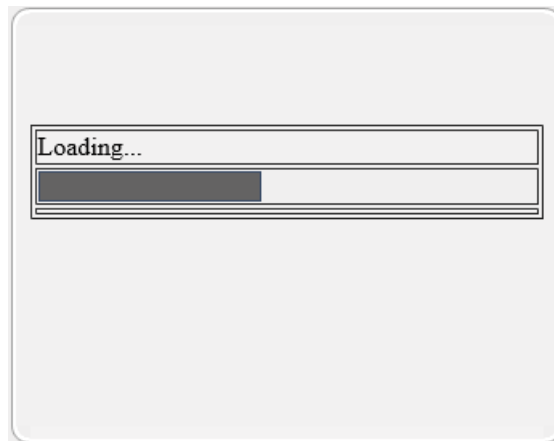
- Click **Start Software Upgrade**; the wizard starts, prompting you to load a .cmp file:

Figure 27-4: Software Upgrade Wizard - Load CMP File



Note: At this stage, you can quit the Software Upgrade Wizard without having to reset the device, by clicking **Cancel** . However, if you continue with the wizard and start loading the cmp file, the upgrade process must be completed with a device reset.

- Click **Browse**, and then navigate to where the .cmp file is located on your computer. Select the file, and then click **Open**.
- Click **Load File**; the device begins to install the .cmp file. A progress bar displays the status of the loading process and a message informs you when file load successfully completes.

Figure 27-5: Software Upgrade Wizard – CMP File Loading Progress Bar

8. If your device is in HA mode, select one of the following upgrade options:

- **Hitless Upgrade** (default)
- **System Reset Upgrade**

See the description of these methods in the beginning of this section.



Note: If you select the **Hitless Upgrade** option, the wizard can only be used to upload a .cmp file; Auxiliary and ini files cannot be uploaded.

9. If you want to load additional files, skip this step and continue with the next step. If you only want to load a .cmp file, click **Reset** ; the device burns the .cmp file to its flash memory and then resets. The device uses the existing configuration (ini) and auxiliary files.



Note: Device reset may take a few minutes (even up to 30 minutes) depending on cmp file version.

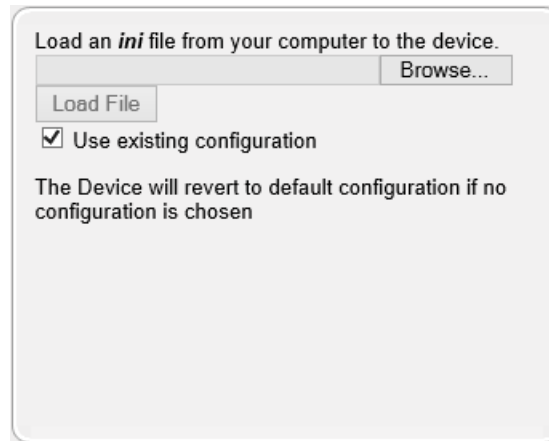
10. To load additional files, use the **Next**  and **Back**  buttons to navigate through the wizard to the desired file-load wizard page. Alternatively, you can navigate to the relevant file-load wizard page by clicking the respective file-name buttons listed in the left pane of the wizard pages.
11. The wizard page for loading an ini file provides you with the following options:
- **Load a new ini file:** In the 'Load an ini file...' field, click **Browse**, and then navigate to where the ini file is located on your computer. Select the file, and then click **Load File**; the device loads the ini file.



Note: 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) and thereby, overwrite values previously configured for these parameters.

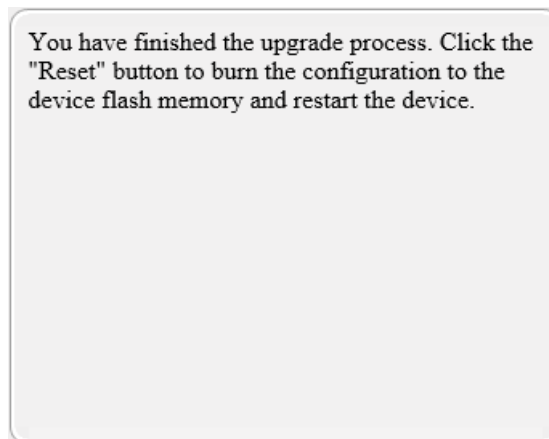
- **Retain the existing configuration (default):** Select the 'Use existing configuration' check box to use the current configuration (and do not select an ini file).
- **Restore configuration to factory defaults:** Clear the 'Use existing configuration' check box (and do not select an ini file).

Figure 27-6: Software Upgrade Wizard – Load INI File



12. When you have completed loading all the desired files, click Next  until the last wizard page appears (the **FINISH** button is highlighted in the left pane):

Figure 27-7: Software Upgrade Wizard – Files Loaded



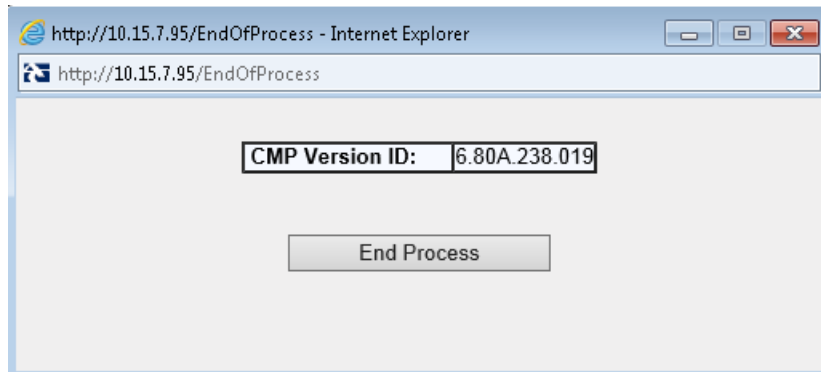
13. Click **Reset**  to burn the files to the device's flash memory; the "Burn and reset in progress" message is displayed and the device 'burns' the newly loaded files to flash memory and then resets.



Note: Device reset may take a few minutes (even up to 30 minutes), depending on .cmp file version.

When the device finishes the installation process and resets, the following wizard page is displayed, showing the installed software version and other files (ini file and auxiliary files) that you may also have installed:

Figure 27-8: Software Upgrade Process Completed Successfully



14. Click **End Process** to close the wizard; the Web Login dialog box appears.
15. Enter your login username and password, and then click **Login**; a message box appears informing you of the new .cmp file version.
16. Click **OK**; the Web interface becomes active, reflecting the upgraded device.

27.4 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
```

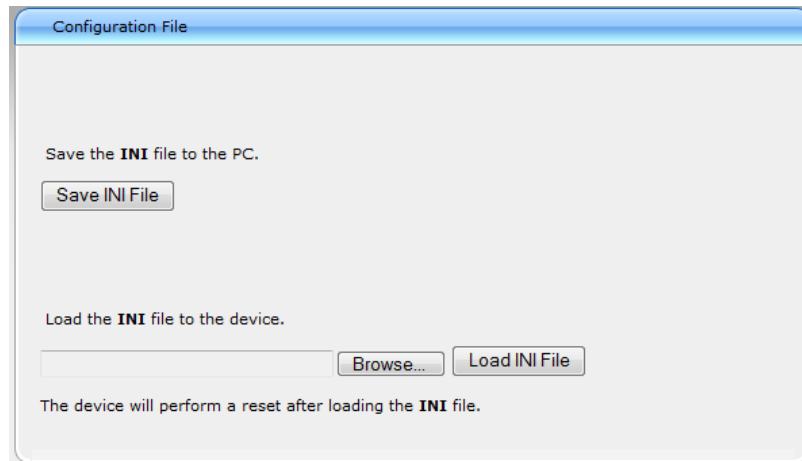


Note: When loading an *ini* file using the Configuration File page, parameters not included in the *ini* file are reset to default settings.

➤ **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 27-9: Configuration File Page



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.

28 Automatic Update Mechanism

This chapter describes the device's Automatic Updated mechanism.

28.1 Automatic Configuration Methods

The table below summarizes the automatic provisioning methods supported by the device:

Automatic Provisioning Methods

BootP / TFTP	DHCP		Automatic Update Methods				SNMP (EMS)
	67	66	HTTP/S	TFTP	FTP	NFS	
No	No	No	Yes	Yes	Yes	No	Yes

28.1.1 DHCP-based Provisioning

A 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.

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:

- 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.

➤ **To enable DHCP:**

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

Figure 10: Enabling DHCP - Application Settings Page

2. From the 'Enable DHCP' drop-down list, select **Enable**.
3. Click **Submit**.
4. To activate the DHCP process, reset device.

The following shows an example of a configuration file for a 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;
    }
}
```



Notes:

- 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.

28.1.2 HTTP-based Provisioning

An HTTP or HTTPS server can be located in the network in which the device is deployed, storing configuration and software files for the device to download. This does not require additional servers and is NAT-safe.

For example, assume the core network HTTPS server is <https://www.corp.com>. A master configuration ini file can be stored 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 device can be configured to periodically check the HTTP server for file updates. 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 only configuration required is to preconfigure the device(s) with the URL of the initial (master) ini file. This can be done using one of the following methods:

- DHCP as described in 'DHCP-based Provisioning' on page 417 or via TFTP at a staging warehouse. The URL is configured using the IniFileURL parameter.
- Private labeling (preconfigured during the manufacturing process).
- 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`

For more information on HTTP-based provisioning, see 'HTTP/S-Based Provisioning using the Automatic Update Feature' on page 420.

28.1.3 FTP-based Provisioning

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 do not 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 Provisioning' on page 418 is that the protocol in the URL is "ftp" (instead of "http").

28.1.4 Provisioning using AudioCodes EMS

AudioCodes EMS server functions as a core-network provisioning server. The device's SNMP Manager should be configured with the IP address of the EMS server, using one of the methods detailed in the previous sections. As soon as a registered device contacts the EMS server through SNMP, the EMS server handles all required configuration automatically, upgrading software as needed. This alternative method doesn't require additional servers at the customer premises, and is NAT-safe.

28.2 HTTP/S-Based Provisioning using the Automatic Update Feature

The Automatic Update feature can be used for automatic provisioning of the device through HTTP/S. 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 device 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.



Warnings: If you use the IniFileURL parameter for the Automatic Update feature, do not use the Web interface to configure the device. If you do configure the device through the Web interface 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 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.



Note: For a description of all the Automatic Update parameters, see Automatic Update Parameters or refer to the CLI Reference Guide.

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>

28.2.1 Files Provisioned by Automatic Update

You can use the Automatic Update feature to update the device with any of the following files:

- Software file (*cmp*)
- Auxiliary files (e.g., Call Progress Tones, SSL Certificates, SSL Private Key)
- Configuration file (ini file)

28.2.2 File Location for Automatic Update

The files for updating the device can be stored on any standard Web (HTTP/S), FTP, or TFTP server. The files can be loaded periodically to the device using HTTP, HTTPS, FTP, or TFTP. This mechanism can be used even when the device is installed behind NAT and firewalls.

The Automatic Update feature is done per file and configured by specifying the file name and URL address of the provisioning server where the file is located. For a description of the parameters used to configure URLs per file, see Automatic Update Parameters. Below are examples for configuring the file names and their URLs 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)# cli-script https://company.com/cli/<MAC>
(automatic-update)# startup-script https://company.com/startup/<MAC>
(automatic-update)# voice-configuration http://www.company.com/configuration.ini
(automatic-update)# call-progress-tones http://www.company.com/call_progress.dat
(automatic-update)# auto-firmware http://www.company.com/SIP_F6.80A.008.cmp
```



Note: For configuration files (ini), the file name in the URL can automatically contain the device's MAC address for enabling the device to download a file unique to the device. For more information, see 'MAC Address Automatically Inserted in Configuration File Name' on page 425.

28.2.3 Triggers for Automatic Update

The Automatic Update feature can be triggered by the following:

- Upon device startup (reset or power up). To disable this trigger, run the following CLI command:

```
(config-system)# automatic-update
(automatic-update)# run-on-reboot off
```

- Periodically:

- Specified time of day (e.g., 18:00), configured by the ini file parameter AutoUpdatePredefinedTime or CLI command `configure system > automatic-update > predefined-time`.
- Interval between Automatic Updates (e.g., every 60 minutes), configured by the ini file parameter AutoUpdateFrequency or CLI command `configure system > automatic-update > update-frequency`.

- Centralized provisioning server request:

- Upon receipt of an SNMP request from the provisioning server.
- Upon receipt of a special SIP NOTIFY message from the provisioning server. The NOTIFY message 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 through the Web interface:

- a. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).
- b. Under the **Misc Parameters** group, set the 'SIP Remote Reset' parameter to **Enable**.
- c. Click **Submit**.

To enable through CLI: `configure voip > sip-definition advanced-settings > sip-remote-reset`.

28.2.4 Access Authentication with HTTP Server

You can configure the device to authenticate itself with the HTTP/S server. The device authenticates itself by providing the HTTP/S server with its authentication username and password. You can configure one of the following HTTP authentication schemes:

- **Basic Access Authentication:** The device provides its username and password to the HTTP server. The username and password is configured in the URL that you define for downloading the file:

- ini file:

```
AutoCmpFileUrl = 'https://<username>:<password>@<IP address or domain name>/<file name>'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# auto-firmware
https://<username>:<password>@<IP address or domain name>/<file name>
```

- **Digest Access Authentication:** The authentication username and password is negotiated between the device and HTTP/S server, using digest MD5 cryptographic hashing. This method is safer than basic access authentication. The digest authentication username and password are configured using the AUPDDigestUsername and AUPDDigestPassword parameters, respectively.

28.2.5 Querying Provisioning Server for Updated Files

Each time the Automatic Update feature is triggered, for each file and its configured URL the device does the following:

1. If you have configured the device to authenticate itself to the HTTP/S server for secure access, the device sends the access authentication username and password to the HTTP/S server (for more information, see Access Authentication with HTTP Server on page 422). If authentication succeeds, Step 2 occurs.
2. 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.
3. The device queries the provisioning server for the requested file by sending an HTTP Get request. This request contains the HTTP User-Agent Header, which identifies the device to the provisioning server. By default, the header includes the device's model 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 further on in this section).

You can configure the information that is sent in the User-Agent header, using the AupdHttpUserAgent parameter or CLI command, configure system > http-user-agent. The information can include any user-defined 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. By default, the device sends the following in the User-Agent header:

```
User-Agent: Mozilla/4.0 (compatible; AudioCodes;
<NAME>;<VER>;<MAC>;<CONF>)
```

For example, if you set `AupdHttpUserAgent = MyWorld-<NAME>;<VER>(<MAC>)`, the device sends the following User-Agent header:

```
User-Agent: MyWorld-Mediant;6.8.200.001(00908F1DD0D3)
```



Note: If you configure the `AupdHttpUserAgent` parameter with the `<CONF>` variable tag, you must reset the device with a burn-to-flash for your settings to take effect.

4. If the provisioning server has relevant files available for the device, the following occurs, depending on file type and configuration:

- **File Download upon each Automatic Update process:** This is applicable to software (.cmp), ini files. In the sent HTTP Get request, the device uses the HTTP If-Modified-Since header to determine whether to download these files. The header contains the date and time (timestamp) of when the device last downloaded the file from the specific URL. This date and time is regardless of whether the file was installed or not on the device. An example of an If-Modified-Since header is shown below:

```
If-Modified-Since: Mon, 1 January 2014 19:43:31 GMT
```

If the file on the provisioning server was unchanged (modified) since the date and time specified in the header, the server replies with an HTTP 304 response and the file is not downloaded. If the file was modified, the provisioning server sends an HTTP 200 OK response with the file in the body of the HTTP response. The device downloads the file and compares the version of the file 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 the automatic software (.cmp) file download method based on this timestamp method, use the ini file parameter, `AutoCmpFileUrl` or CLI command, `configure system > automatic-update > auto-firmware <URL>`. The device uses the same configured URL to download the .cmp file for each subsequent Automatic Update process.



Notes:

- 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.
- The Automatic Update feature assumes that the 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 ini file parameter `AutoUpdateFrequency` or CLI command `configure system > automatic update > update-frequency`.

- **One-time File Download:** This is applicable to software (.cmp) and Auxiliary (e.g., call progress tone / CPT) files. The device downloads these files only **once**, regardless of how many times the device may repeat the Automatic Update process. Once they are downloaded, the device discards their configured URLs. To update these files again, you need to configure their URL addresses and filenames again. Below is an example of how to configure URLs for some of these files:

Auxiliary Files:

◆ **ini:**

```
CptFileUrl = 'https://www.company.com/call_progress.dat'
```

◆ **CLI:**

```
(config-system)# automatic-update
(automatic-update)# call-progress-tones
http://www.company.com/call_progress.dat
(automatic-update)# tls-root-cert https://company.com/root.pem
```

Software (.cmp) File:

◆ **ini:**

```
CmpFileUrl =
'https://www.company.com/device/v.6.80A.227.005.cmp'
```

◆ **CLI:**

```
(config-system)# automatic-update
(automatic-update)# firmware
https://www.company.com/device/v.6.80A.227.005.cmp
```



Notes:

- For one-time file download, the HTTP Get request sent by the device does not include the If-Modified-Since header. Instead, the HTTP-User-Agent header can be used in the HTTP Get request to determine whether firmware update is required.
- When downloading SSL certificates (Auxiliary file), it is recommended to use HTTPS with mutual authentication for secure transfer of the SSL Private Key.

5. If the device receives an HTTP 301/302/303 redirect response from the provisioning server, it establishes a connection with the new server at the redirect URL and re-sends the HTTP Get request.

28.2.6 File Download Sequence

Whenever the Automatic Update feature is triggered (see 'Triggers for Automatic Update' on page 421), the device attempts to download each file from the configured URLs, in the following order:

1. ini file
2. Periodic software file (.cmp) download
3. One-time software file (.cmp) download
4. Auxiliary file(s)

The following files automatically instruct the device to reset:

- Periodic software file (.cmp)
- One-time software file (.cmp)

When multiple files requiring a reset are downloaded, the device resets only **after** it has downloaded and installed **all** the files. However, you can explicitly instruct the device to immediately reset for the following files:

- ini file: Use the ResetNow in file parameter



Warning: If you use the ResetNow parameter in an ini file for periodic automatic provisioning with non-HTTP (e.g., TFTP) and without CRC, the device resets after every file download. Therefore, use this parameter with caution and only if necessary for your deployment requirements.

**Notes:**

- For ini file downloads, by default, parameters not included in the file are set to defaults. To retain the current settings of these parameters, set the SetDefaultOnINIFileProcess parameter to 0.
- If you have configured one-time software file (.cmp) download (configured by the ini file parameter CmpFileURL or CLI command `configure system > automatic-update > firmware`), the device will only apply the file if one-time software updates are enabled. This is disabled by default to prevent unintentional software upgrades. To enable one-time software upgrades, set the ini file parameter AutoUpdateCmpFile to 1 or CLI command `configure system > automatic-update > update-firmware on`.
- If you need to update the device's software and configuration, it is recommended to first update the software. This is because the current ("old") software (before the upgrade) may not be compatible with the new configuration. However, if both files are available for download on the provisioning server(s), the device first downloads and applies the new configuration, and only then does it download and install the new software. Therefore, this is a very important issue to take into consideration.

28.2.7 Cyclic Redundancy Check on Downloaded Configuration Files

You can enable the device to perform cyclic redundancy checks (CRC) on downloaded configuration files (ini) during the Automatic Update process. The CRC checks whether the content (raw data) of the downloaded file is different to the content of the previously downloaded file from the previous Automatic Update process. The device compares the CRC check value (code) result with the check value of the previously downloaded file. If the check values are identical, it indicates that the file has no new configuration settings, and the device discards the file. If the check values are different, it indicates that the downloaded file is different (i.e., includes updates), and the device installs the downloaded file and applies the new configuration settings.

CRC is useful, for example, when the service provider replaces a file, on the provisioning server, with another file whose contents are the same. When the device sends an HTTP Get request during the Automatic Update process, the provisioning server sends the new file to the device. This occurs as the timestamp between the previously downloaded file and this new file is different (determined by the HTTP If-Modified-Since header in the Get request). Therefore, the CRC feature can be used to prevent the device from installing such files.

For enabling CRC, use the ini file parameter AUPDCheckIfIniChanged or CLI command `configure system > automatic-update > crc-check regular`. By default, CRC is disabled. For more information on the parameter, see Automatic Update Parameters.

28.2.8 MAC Address Automatically Inserted in Configuration File Name

You can configure the file name of the configuration file (ini) in the URL to automatically include the MAC address of the device. As described in 'File Location for Automatic Update' on page 420, the file name is included in the configured URL of the provisioning server where the file is located.

Including the MAC address in the file name is useful if you want the device to download a file that is unique to the device. This feature is typically implemented in mass provisioning of devices where each device downloads a specific configuration file. In such a setup, the provisioning server stores configuration files per device, where each file includes the MAC address of a specific device in its file name.

To support this feature, you need to include the MAC address placeholder string, "<MAC>" anywhere in the configured file name of the URL, for example:

```
IniFileURL = 'https://www.company.com/config_<MAC>.ini'
```

The device automatically replaces the string with its hardware MAC address, resulting in a file name request that contains the device's MAC address, for example, *config_00908F033512.ini*. Therefore, you can configure all the devices with the same URL and file name.



Note: If you write the MAC address placeholder string in lower case (i.e., "<mac>"), the device adds the MAC address in lower case to the file name (e.g., *config_<mac>.ini* results in *config_00908f053736e*); if in upper case (i.e., "<MAC>"), the device adds the MAC address in upper case to the file name (e.g., *config_<MAC>.ini* results in *config_00908F053736E*).

28.2.9 Automatic Update Configuration Examples

This section provides a few examples on configuring the Automatic Update feature.

28.2.9.1 Automatic Update for Single Device

This simple example describes how to configure the Automatic Update feature for updating a single device. In this example, the device queries the provisioning server for software, configuration and auxiliary files every 24 hours.

➤ **To set up Automatic Provisioning for single device (example):**

1. Set up an HTTP Web server (e.g., <http://www.company.com>) and place all the required configuration files on this server.
2. Configure the device with the IP address of the DNS server for resolving the domain name (e.g., <http://www.company.com>) that is used in the URL of the provisioning server. You configure this in the Interface table:

- ini File:

```
[ InterfaceTable ]
FORMAT InterfaceTable_Index =
InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress,
InterfaceTable_UnderlyingDevice;
InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
"Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
[ \InterfaceTable ]
```

- CLI:

```
# configure voip
(config-voip)# interface network-if 0
(network-if-0)# primary-dns 80.179.52.100
```

3. Configure the device with the following Automatic Update settings:

- a. Automatic Update is done every 24 hours (1440 minutes):

- ♦ ini File:

```
AutoUpdateFrequency = 1440
```

- ◆ CLI:


```
# configure system
(config-system)# automatic update
(automatic-update)# update-frequency 1440
```
 - b. Automatic Update of software file (.cmp):
 - ◆ ini File:


```
AutoCmpFileUrl = 'https://www.company.com/sw.cmp'
```
 - ◆ CLI:


```
# configure system
(config-system)# automatic update
(automatic-update)# auto-firmware 'http://www.company.com/sw.cmp'
```
 - c. Automatic Update of Call Progress Tone file:
 - ◆ ini File:


```
CptFileURL = 'https://www.company.com/call_progress.dat'
```
 - ◆ CLI:


```
# configure system
(config-system)# automatic update
(automatic-update)# call-progress-tones
'http://www.company.com/call_progress.dat'
```
 - d. Automatic Update of ini configuration file:
 - ◆ ini File:


```
IniFileURL = 'https://www.company.com/config.ini'
```
 - ◆ CLI:


```
# configure system
(config-system)# automatic update
(automatic-update)# voice-configuration 'http://www.company.com/config.ini'
```
 - e. Enable Cyclical Redundancy Check (CRC) on downloaded ini file:
 - ◆ ini File:


```
AUPDCheckIfIniChanged = 1
```
 - ◆ CLI:


```
# configure system
(config-system)# automatic update
(automatic-update)# crc-check regular
```
4. Power down and then power up the device.

28.2.9.2 Automatic Update from Remote Servers

This example describes how to configure the Automatic Update feature where files are stored and downloaded from different file server types. The example scenario includes the following:

- FTPS server at ftpserver.corp.com for storing the Voice Prompts (VP) file. The login credentials to the server are username "root" and password "wheel".
 - HTTP server at www.company.com for storing the ini configuration file.
 - DNS server at 80.179.52.100 for resolving the domain names of the provisioning servers (FTPS and HTTP).
- **To set up Automatic Provisioning for files stored on different server types (example):**
1. **VP file:**
 - a. Set up an FTPS server and copy the VP file to the server.

- b. Configure the device with the URL path of the VP file:

```
VPFileUrl = 'ftps://root:wheel@ftpserver.corp.com/vp.dat'
```

2. **Software (.cmp) and ini files:**

- a. Set up an HTTP Web server and copy the .cmp and ini files to the server.

- b. Configure the device with the URL paths of the .cmp and ini files:

```
AutoCmpFileUrl = 'http://www.company.com/device/sw.cmp'
```

```
IniFileURL = 'http://www.company.com/device/inifile.ini'
```

3. Configure the device with the IP address of the DNS server for resolving the domain names of the FTPS and HTTP servers:

```
[ InterfaceTable ]
FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress,
InterfaceTable_UnderlyingDevice;
InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
"Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
[ \InterfaceTable ]
```

4. Configure the device to perform the Automatic Update process daily at 03:00 (3 a.m):

```
AutoUpdatePredefinedTime = '03:00'
```

28.2.9.3 Automatic Update for Mass Deployment

This example describes how to configure the Automatic Update feature for updating multiple devices (i.e., mass deployment) using an HTTP provisioning server. In this example, all the devices are configured to download the same "master" configuration file. This file serves as the configuration template and instructs the devices which files to download and how often to perform the Automatic Update process. In addition, the master file also instructs each device to download an ini configuration file whose file name contains the MAC address of the device.

The example scenario is as follows:

- All devices download a "master" configuration file that contains the following:
 - Common configuration shared by all device's.
 - Specific configuration that instructs each device to download a specific configuration file based on the device's MAC address, using the special string "<MAC>" in the URL, as described in 'MAC Address Automatically Inserted in Configuration File Name' on page 425.
- Device queries the provisioning server daily at 24:00 (midnight) for software, configuration and auxiliary files.
- HTTP-based provisioning server at www.company.com for storing the files.
- DNS server at 80.179.52.100 for resolving the domain name of the provisioning server.

➤ **To set up automatic provisioning for mass provisioning (example):**

1. Create an ini file named "master_configuration.ini" with the following settings:
 - a. Common configuration for all devices:


```
# Check for updates daily at 24:00
AutoUpdatePredefinedTime = '24:00'
# CPT file update:
CptFileURL = 'https://www.company.com/call_progress.dat'
# Software (.cmp) file update:
AutoCmpFileUrl = 'https://www.company.com/sw.cmp'
```
 - b. Configuration per device based on MAC address:


```
IniFileURL = 'http://www.company.com/config_<MAC>.ini'
```
2. Copy the *master_configuration.ini* file as well as the CPT and .cmp files to the HTTP-based provisioning server.
3. Configure **each** device with the following:
 - a. URL of the *master_configuration.ini* file for Automatic Update:
 - ◆ ini File:


```
IniFileURL =
'http://www.company.com/master_configuration.ini'
```
 - ◆ CLI:


```
# configure system
(config-system)# automatic update
(automatic-update)# voice-configuration
'http://www.company.com/master_configuration.ini'
```
 - b. Configure the device with the IP address of the DNS server for resolving the domain name (e.g., http://www.company.com) that is used in the URL for the provisioning server. This is done in the Interface table:
 - ◆ ini File:


```
[ InterfaceTable ]
FORMAT InterfaceTable_Index =
InterfaceTable_ApplicationTypes,
InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
InterfaceTable_PrefixLength, InterfaceTable_Gateway,
InterfaceTable_VlanID, InterfaceTable_InterfaceName,
InterfaceTable_PrimaryDNSServerIPAddress,
InterfaceTable_SecondaryDNSServerIPAddress,
InterfaceTable_UnderlyingDevice;
InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
"Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
[ \InterfaceTable ]
```
 - ◆ CLI:


```
# configure voip
(config-voip)# interface network-if 0
(network-if-0)# primary-dns 80.179.52.100
```
4. Power down and then power up the device.

29 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 431)
- Loading an empty *ini* file (see "Restoring Defaults using an ini File" on page 432)

29.1 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

29.2 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 415). 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.

30 Saving Current Configuration to a File and Sending it to Remote Destination

You can save (create) the current configuration as a configuration file on the device's flash memory and then have it sent to a user-defined URL of a remote server (TFTP or HTTP/S). The configuration settings in the file are based only on CLI commands. This is done through CLI:

- Creating a Configuration file and saving it on a remote server:

```
# write-and-backup to <URL path with file name>
```

For example:

```
# write-and-backup to tftp://192.168.0.3/config-device1.txt
```

This page is intentionally left blank.

Part VIII

Status, Performance Monitoring and Reporting

31 System Status

This section describes how to view various system statuses.

31.1 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
▼ Loaded Files	

➤ **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 387).

31.2 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 44)

	Active	Speed	Duplex Mode	State	Group Member
1	Yes	1000 Mbps	Full Duplex	Forwarding	GROUP_1
2	Yes	100 Mbps	Half Duplex	Forwarding	GROUP_2

Table 31-1: 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.

32 Carrier-Grade Alarms

This section describes how to view the following types of alarms:

- Active alarms - see "Viewing Active Alarms" on page 439
- Alarm history - see "Viewing Alarm History" on page 439

32.1 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 44).



Note:

- The alarms in the table are deleted upon a device reset.
- To configure the maximum number of active alarms that can be displayed in the table, see the ini file parameter, `ActiveAlarmTableMaxSize`.
- For more information on SNMP alarms, refer to the *SNMP Reference Guide* document.

➤ **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.

32.2 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.

33 Performance Monitoring

This section describes how to view performance monitoring.

33.1 Viewing MOS per Media Realm

The MOS Per Media Realm page displays statistics on Media Realms (configured in "Configuring Media Realms" on page 227). 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 33-1: MOS Per Media Realm Graph



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.

33.2 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 33-2: Quality Of Experience Graph



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**.

33.3 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 33-3: Average Call Duration Graph



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.

This page is intentionally left blank.

34 VoIP Status

This section describes how to view VoIP status and statistics.

34.1 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 112).

➤ **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

34.2 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 110.

➤ **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

34.3 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 119) 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 34-1: 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	

34.4 Viewing Registered Users

You can view SBC users listed in the device's Users Registration database. The list shows each Address of Record (AOR) and its corresponding contact. The contact's registration status is also shown:

- "Active status:1" indicates that the contact has been successfully registered and thus, calls can be routed to it.
- "Active status:0" indicates that the device has recently received a REGISTER request from the contact, but the contact has yet to be registered. The device removes the contact from the database if no response is received within a few seconds from the proxy/registrar server.

An 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. A contact is a SIP URI that can be used to contact that specific instance of the user agent for subsequent requests.

- **To view registered SBC users in the Users Registration database:**
 - **Web:** SAS/SBC Registered Users page (**Status & Diagnostics** tab > **VoIP Status** menu > **SAS/SBC Registered Users**).

Figure 34-2: 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

- **CLI:**
 - SBC users:
show voip register db sbc list
 - SBC contacts of a specified AOR:
show voip register db sbc user <Address Of Record>

34.5 Viewing Registration Status

The Registration Status page displays the registration status of the device's SIP Accounts, which are configured in the Accounts table (see "Configuring Registration Accounts" on page 259).

➤ **To view the registration status:**

- Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registration Status**).

Figure 34-3: Registration Status Page

Registered Per Gateway				NO
▼ Accounts Registration Status				
Index	Group Type	Group Name	Status	

- **Accounts Registration Status:**
 - ◆ Group Type: served IP Group
 - ◆ Group Name: name of the served IP Group, if applicable
 - ◆ Status: "Registered" or "Unregistered"

34.6 Viewing Proxy Set Status

You can view the status of Proxy Sets that are used in your call routing topology. Proxy Sets that are not associated with any routing rule are not displayed.

To configure proxy Sets, see Configuring Proxy Sets on page 251.

➤ **To view Proxy Set status:**

- Open the Active Proxy Set Status page (**Monitor** menu > **Monitor** tab > **VoIP Status** folder > **Proxy Sets Status**).

Figure 34-4: Viewing Proxy Sets Status

Active Proxy Sets Status								
Proxy Set ID	Mode	Keep Alive	Address	Priority	Weight	Success Count	Failure Count	Status
0	Parking	Enabled	abc.com(199.181.132.250)	-	-	0	11	OFFLINE
1	Homing	Enabled	ipbx2.com	-	-	0	0	NOT RESOLVED
2	Parking	Disabled	10.8.6.77(*)	-	-	0	0	ONLINE
3	Load Balancing	Enabled	10.8.6.88	-	-	0	45	OFFLINE
			10.8.6.89(*)	-	-	4	0	ONLINE
4	Parking	Enabled	10.8.6.66	-	-	0	45	OFFLINE
5	Parking	Enabled	ipbx3.com	-	-	0	0	NOT RESOLVED
6	Parking	Enabled	ipbx3.com(10.8.8.1)(*)	-	-	0	0	NOT RESOLVED
			ipbx3.com(10.8.8.2)	-	-	0	0	NOT RESOLVED

Table 34-1: Proxy Sets Status Table Description

Parameter	Description
Proxy Set ID	Displays the Proxy Set ID.
Mode	Displays the Proxy Sets' operational mode: ▪ "Parking" or "Homing": Redundancy mode, as configured by the ProxySet_ProxyRedundancyMode parameter. ▪ "Load Balancing": Proxy load balancing mode, as configured by the ProxySet_ProxyRedundancyMode parameter. For more information, see Configuring Proxy Sets.
Keep Alive	Displays whether the Proxy Keep-Alive feature is enabled ("Enabled") or disabled ("Disabled"), as configured by the ProxySet_EnableProxyKeepAlive parameter (see Configuring Proxy Sets).
Address	Displays the IP address of the proxy server. This can be the IP address as configured in dotted-decimal notation for the Proxy Set, or the resolved IP address of a DNS query if an FQDN is configured for the Proxy Set. IP addresses resolved from FQDNs are displayed as "<FQDN name>(<resolved IP address>)", for example, "abc.com(10.8.6.80)". The IP address that is currently used for routing is indicated with an asterisk, for example, "10.8.6.89(*)". If the FQDN failed to be resolved, only the FQDN name is displayed (e.g., "abc.com").
Priority	Displays the priority of IP addresses resolved from FQDNs. Note: The field is applicable only to Proxy Sets configured with

Parameter	Description
	FQDNs.
Weight	Displays the weight of IP addresses resolved from FQDNs. Note: The field is applicable only to Proxy Sets configured with FQDNs.
Success Count	Displays the total number of successful keep-alive messages (by SIP OPTIONS) sent by the device to the proxy.
Failure Count	Displays the total number of failed keep-alive messages (by SIP OPTIONS) sent by the device to the proxy.
Status	Displays the status of the Proxy Set and its' proxy servers. ▪ "ONLINE": ✓ Proxy Set ID row: At least one proxy is online as determined by the device's keep-alive feature. The status is also "ONLINE" for IP addresses resolved from DNS queries even if keep-alive is disabled. ✓ Proxy server rows (if multiple addresses): The proxy server is online as determined by the device's keep-alive feature. ▪ "OFFLINE": The proxy is offline as determined by the device's keep-alive feature and the Proxy Set is configured for Homing ('Redundancy Mode' parameter) or enabled for load balancing ('Proxy Load Balancing Method' parameter): ✓ Homing: The proxy is the main proxy, but the keep-alive has failed. ✓ Load balancing: The keep-alive for the proxy has failed. ▪ "NOT RESOLVED": Proxy address is configured as an FQDN, but the DNS resolution has failed. ▪ Empty field: Keep-alive for the proxy is disabled or the device has yet to send a keep-alive to the proxy.

This page is intentionally left blank.

35 Reporting Information to External Party

This section describes features for reporting various information to an external party.

35.1 Configuring 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.



Notes:

- The following standard RADIUS Attributes cannot be customized: 1 through 6, 18 through 20, 22, 23, 27 through 29, 32, 34 through 39, 41, 44, 52, 53, 55, 60 through 85, 88, 90, and 91.
- If the RTCP XR feature is unavailable (not licensed or disabled), the R-factor VoIP metrics are not provided in CDRs (CDR fields, Local R Factor and Remote R Factor) generated by the device. Instead, these CDR fields are sent with the value 127, meaning that information is unavailable.

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 35-1: 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

Group	Metric Name
	Packet Loss Concealment
	Silence Suppression State
Jitter Buffer	Jitter Buffer Adaptive
	Jitter Buffer Rate
	Jitter Buffer Nominal
	Jitter Buffer Max
	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, SUB
SCRIBE, 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=13746175212842012155835@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:13746175212842012155835@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 35-1: RTCP XR Parameters in RTP/RTCP Settings Page

RTCP XR Settings	
Enable RTCP XR	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' (*VQMonBurstTHR*) - 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:
 - '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.

35.2 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 3164 and is identified by Facility 17 (local1) and Severity 6 (Informational).

For CDR in RADIUS format, see "Configuring RADIUS Accounting" on page 461.

35.2.1 Configuring CDR Reporting

The following procedure 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 "Enabling Syslog" on page 473.
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 35-2: CDR Parameters in Advanced Parameters Page

CDR and Debug	
CDR Server IP Address	10.8.6.55
CDR Report Level	Start & End Call
Media CDR Report Level	End Media
CDR Syslog Sequence Number	Enable

3. Configure the parameters as required. For a description of the parameters, see "Syslog, CDR and Debug Parameters" on page 534.
4. (Optional) Disable the inclusion of the Sequence Number in Syslog messages, by setting the 'CDR Session ID' parameter to **Disable**.
5. Click **Submit**.



Note:

- If you do not configure an IP address for a CDR server, the device sends CDRs to the Syslog server, as configured in 'Enabling Syslog' on page 473.
- The device sends CDRs only for dialog-initiating INVITE messages (call start), 200 OK responses (call connect) and BYE messages (call end). If you want to enable the generation of CDRs for non-call SIP dialogs (such as SUBSCRIBE, OPTIONS, and REGISTER), use the EnableNonCallCdr parameter.
- To configure the time zone string (e.g., GMT+1) that is displayed with the timestamp in CDRs ("Connect Time", "Release Time", and "Setup Time" CDR fields), use the TimeZoneFormat parameter.

35.2.2 CDR Field Description

This section describes the CDR fields that are generated by the device.

35.2.2.1 CDR Fields for SBC Signaling

The CDR fields for SBC signaling are listed in the table below.

A typical SBC session consists of two SBC legs. Each leg generates its own signaling CDRs. Each leg generates three CDR types: at call start (SBCReportType=CALL_START), connect time (SBCReportType=CALL_CONNECT) and when the call ends (SBCReportType=CALL_END). CDRs belonging to the same SBC session (both legs) have the same Session ID (SessionId CDR field). CDRs belonging to the same SBC leg have the same SIP Call ID (SIPCallId CDR field).

For billing applications, the CDR that is sent when the call ends (CALL_END) is usually sufficient. Billing may be based on the following:

- Call ID (SIPCallId CDR field)
- Source URI (SrcURI CDR field)
- Destination URI (DstURI CDR field)
- Call originator (Orig CDR field) - indicates the call direction (caller)
- Call duration (Durat CDR field) - call duration (elapsed time) from call connect
- Call time is based on SetupTime, ConnectTime and ReleaseTime CDR fields

Table 35-2: CDR Fields for SBC Signaling

CDR Field Name	Description	Format
SBCReportType	Report Type: ■ "CALL_START" ■ "CALL_CONNECT" ■ "CALL_END" ■ "DIALOG_START" ■ "DIALOG_END"	String
EPTyp	Endpoint type: ■ "SBC"	String
SIPMethod	SIP message type	String of up to 10 characters
SIPCallId	Unique ID of call	String of up to 50 characters
SessionId	Unique Session ID	String of up to 10 characters
Orig	Call originator: ■ "LCL" - local ■ "RMT" - remote	String
SourceIp	Source IP address	String of up to 20 characters
SourcePort	Source UDP port	String of up to 10 characters
DestIp	Destination IP address	String of up to 20 characters
DestPort	Destination UDP port	String of up to 10 characters
TransportType	Transport type: ■ "UDP" ■ "TCP" ■ "TLS"	String

CDR Field Name	Description	Format
SrcURI	Source URI	String of up to 41 characters
SrcURIBeforeMap	Source URI before manipulation	String of up to 41 characters
DstURI	Destination URI	String of up to 41 characters
DstURIBeforeMap	Destination URI before manipulation	String of up to 41 characters
Durat	Call duration (in seconds)	String of up to 5 characters
TrmSd	Termination side: ▪ "LCL" – local ▪ "RMT" - remote	String
TrmReason	Termination reason	String of up to 40 characters
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_RESOURCE_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_FORWARDED" ▪ 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	String of up to 17 characters
SetupTime	Call setup time	String of up to 35 characters

CDR Field Name	Description	Format
	Note: To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter.	
ConnectTime	Call connect time Note: To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter.	String of up to 35 characters
ReleaseTime	Call release time Note: To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter.	String of up to 35 characters
RedirectReason	Redirect reason	String of up to 15 characters
RedirectURINum	Redirection URI	String of up to 41 characters
RedirectURINumBeforeMap	Redirect URI number before manipulation	String of up to 41 characters
TxSigIPDiffServ	Signaling IP DiffServ	String of up to 15 characters
IPGroup	IP Group ID and name	String of up to 40 characters
SrdId	SRD ID and name	String of up to 29 characters
SIPInterfaceId	SIP Interface ID	String of up to 15 characters
ProxySetId	Proxy Set ID	String of up to 15 characters
IpProfileId	IP Profile ID and name	String of up to 34 characters
MediaRealmId	Media Realm ID and name	String of up to 55 characters
DirectMedia	Direct media or traversing SBC: ▪ "yes" ▪ "no"	String
SIPTrmReason	SIP call termination reason (BYE, CANCEL, or SIP error codes, e.g., 404)	String of up to 12 characters.
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.	String of up to 26 characters

CDR Field Name	Description	Format
Caller	Name of caller	String of up to 36 characters
Callee	Name of called party	String of up to 36 characters

Below shows an example of an SBC signaling CDR sent at the end of a call (call was terminated normally):

```
[S=40] |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 |SIPTermDesc |Caller |Callee

[S=41] |CALL_END |SBC |20767593291410201017029@10.33.45.80
|1871197419|LCL |10.33.45.80 |5060 |10.33.45.72 |5060 |UDP
|9001@10.8.8.10 |9001@10.8.8.10 |6001@10.33.45.80
|6001@10.33.45.80 |15 |LCL |GWAPP_NORMAL_CALL_CLEAR
|NORMAL_CALL_CLEAR |17:00:29.954 UTC Thu Oct 14 2014
|17:00:49.052 UTC Thu Oct 14 2014 |17:01:04.953 UTC Thu Oct 14
2014 |-1 | | |40 |1 |0 (SRD_GW) |1 |1 |1 () |0 (MR_1) |no |BYE
|Q.850 ;cause=16 ;text="loc |user 9928019 |
```

35.2.2.2 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 35-3: 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

CDR Field Name	Description
RTPdelay	RTP delay
RTPjitter	RTP jitter
TxRTPssrc	Tx RTP SSRC
RxRTPssrc	Local RTP SSRC
LocalRFactor	Local conversation quality Note: If the RTCP XR feature is unavailable (not licensed or disabled), this R-factor VoIP metric is not provided. Instead, the device sends the CDR field with the value 127, meaning that information is unavailable.
RemoteRFactor	Remote conversation quality Note: If the RTCP XR feature is unavailable (not licensed or disabled), this R-factor VoIP metric is not provided. Instead, the device sends the CDR field with the value 127, meaning that information is unavailable.
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

35.3 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 35-3: 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 552.
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 35-4: Supported RADIUS Accounting CDR Attributes

Attribute ID	Attribute Name	Vendor-Specific Attribute (VSA) ID	Description	Value Format	Example	AAA
Request Attributes						
1	user-name	(Standard)	Account number or calling party number or blank	String up to 15 digits long	5421385747	Start Acc Stop Acc
4	nas-ip-address	(Standard)	IP address of the requesting device	Numeric	192.168.14.43	Start Acc Stop Acc
6	service-type	(Standard)	Type of service requested	Numeric	1: login	Start Acc Stop Acc
26	h323-incoming-conf-id	1	SIP call identifier	Up to 32 octets	h323-incoming-conf-id=38393530	Start Acc Stop Acc

Attribute ID	Attribute Name	Vendor-Specific Attribute (VSA) ID	Description	Value Format	Example	AAA
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	h323-setup-time=09:33:26.621 Mon Dec 2014	Start Acc Stop Acc
26	h323-call-origin	26	Originator of call: "answer": Call originated from the IP side (Gateway) or incoming leg (SBC) "originate": Call originated from the Tel side (Gateway) or outgoing leg (SBC)	String	h323-call-origin=answer	Start Acc Stop Acc
26	h323-call-type	27	Protocol type or family used on this leg of the call	String	h323-call-type=VOIP	Start Acc Stop Acc
26	h323-connect-time	28	Connect time in NTP format	String	h323-connect-time=09:33:37.657 UTC Mon Dec 08 2015	Stop Acc
26	h323-disconnect-time	29	Disconnect time in NTP format	String	-	Stop Acc
26	h323-disconnect-cause	30	Disconnect cause code (Q.850)	Numeric	h323-disconnect-cause=16	Stop Acc
26	h323-gw-id	33	Name of the gateway	String	h323-gw-id=<SIP ID string>	Start Acc Stop Acc
26	sip-call-id	34	SIP Call ID	String	sip-call-id=abcde@ac.com	Start Acc Stop Acc
26	call-terminator	35	Terminator of the call:	String	call-terminator=yes	Stop Acc

Attribute ID	Attribute Name	Vendor-Specific Attribute (VSA) ID	Description	Value Format	Example	AAA
			"yes": Call terminated by the Tel side (Gateway) or outgoing leg (SBC) "no": Call terminated by the IP side (Gateway) or incoming leg (SBC)			
26	terminator	37	Terminator of the call: "answer": Call originated from the IP side (Gateway) or incoming leg (SBC) "originate": Call originated from the Tel side (Gateway) or outgoing leg (SBC)	String	terminator=originate	Stop Acc
30	called-station-id	(Standard)	Destination phone number (Gateway call) or Destination URI (SBC call)	String	8004567145	Start Acc
31	calling-station-id	(Standard)	Calling Party Number (ANI) (Gateway call) or Source URI (SBC call)	String	5135672127	Start Acc Stop Acc
40	acct-status-type	(Standard)	Account Request Type - start (1) or stop (2) Note: 'start' isn't supported on the Calling Card application.	Numeric	1	Start Acc Stop Acc
41	acct-delay-time	(Standard)	No. of seconds tried in sending a particular record	Numeric	5	Start Acc Stop Acc

Attribute ID	Attribute Name	Vendor-Specific Attribute (VSA) ID	Description	Value Format	Example	AAA
42	acct-input-octets	(Standard)	Number of octets received for that call duration (for SBC calls, applicable only if media anchoring)	Numeric	-	Stop Acc
43	acct-output-octets	(Standard)	Number of octets sent for that call duration (for SBC calls, applicable only if media anchoring)	Numeric	-	Stop Acc
44	acct-session-id	(Standard)	A unique accounting identifier - match start & stop	String	34832	Start Acc Stop Acc
46	acct-session-time	(Standard)	For how many seconds the user received the service	Numeric	-	Stop Acc
47	acct-input-packets	(Standard)	Number of packets received during the call	Numeric	-	Stop Acc
48	acct-output-packets	(Standard)	Number of packets sent during the call	Numeric	-	Stop Acc
61	nas-port-type	(Standard)	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	(Standard)	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
```

This page is intentionally left blank.

Part IX

Diagnostics

36 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.

36.1 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 "Enabling Syslog" on page 473).

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 36-1: 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): <pre>18:38:14. 52 : 10.33.45.72 : NOTICE:</pre>

Message Item	Description
	<pre>[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: [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]</pre> You can disable the inclusion of the message sequence number in Syslog messages, by setting the 'CDR Session ID' parameter to Disable (see Configuring CDR Reporting on page 455).
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.

36.1.1 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 36-2: 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

Error Abbreviation	Error Name Description
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
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

36.1.2 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 36-3: Syslog Facility Levels

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)
```

36.1.3 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 36-4: Syslog Message Severity

ITU Perceived Severity (SNMP Alarm's Severity)	AudioCodes' Syslog Severity
Critical	RecoverableMsg
Major	RecoverableMsg
Minor	RecoverableMsg
Warning	Notice
Indeterminate	Notice

ITU Perceived Severity (SNMP Alarm's Severity)	AudioCodes' Syslog Severity
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:

36.2 Enabling Syslog

The following procedure describes how to enable and configure Syslog.



Notes:

- For configuring CDR reporting, see "Configuring CDR Reporting" on page [455](#).
- For viewing Syslog messages in the Web interface, see "Viewing Syslog Messages" on page [479](#).
- For a detailed description on the Syslog parameters, see "Syslog, CDR and Debug Parameters" on page [534](#).

➤ To enable Syslog:

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

Figure 36-1: Syslog Settings Page

Syslog Settings	
Enable Syslog	Enable
Syslog Server IP Address	10.15.50.1
Syslog Server Port	514
Syslog CPU Protection	Enabled
Syslog Optimization	Enabled
Debug Level	Detailed

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. Configure the debug level using the 'Debug Level' parameter. This determines the level of messages that the device sends to the Syslog server. If set to Basic or Detailed, you can also configure related features using the following parameters:
 - 'Syslog Optimization' (SyslogOptimization): Enables the device to accumulate and bundle multiple debug messages into a single UDP packet and then send it to a Syslog server. The benefit of this feature is that it reduces the number of UDP Syslog packets, thereby improving (optimizing) CPU utilization. The size of the bundled message is configured by the MaxBundleSyslogLength parameter.
 - 'Syslog CPU Protection' (SyslogCpuProtection): Enables the protection of the device's CPU resources during debug reporting, ensuring voice traffic is unaffected. If CPU resources drop (i.e., high CPU usage) to a critical level (threshold), the device automatically lowers the debug level to free up CPU resources that were required for the previous debug-level functionality. When CPU resources become available again, the device increases the debug level. The threshold is configured by the DebugLevelHighThreshold parameter (see below).

- DebugLevelHighThreshold: Defines the threshold (in percentage) for automatically switching to a different debug level, depending on CPU usage. For more information about this functionality, refer to the parameter's description in Syslog, CDR and Debug Parameters on page 534.

5. Click **Submit**.

36.3 Configuring Web Operations to Report to Syslog

You can define the operations (activities) in the Web interface that must be reported to the Syslog server. The following procedure describes how to configure this in the Web interface. You can also configure this using the ini file parameter, ActivityListToLog or CLI command, config-system > logging > activity-log.

➤ **To define Web activities to report to Syslog server:**

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).
2. Under the Activity Types to Report via Activity Log Messages group, select the Web actions to report to the Syslog server. For more information, see "Syslog, CDR and Debug Parameters" on page 534.

Figure 36-2: Web Activities to Report to Syslog

▼ 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	☐

3. Click **Submit**.

36.4 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 36-3: 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 534.

36.5 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.

The following procedure 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.

➤ **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 36-4: Logging Filters Table - Add Record Dialog Box

The dialog box titled 'Add Record' contains the following fields and options:

- Index:** A text input field containing the value '1'.
- Type:** A dropdown menu with 'Any Filter' selected.
- Value:** An empty text input field.
- Syslog:** A dropdown menu with 'Enable' selected.
- Capture Type:** A dropdown menu with 'Signaling + Media' selected.
- Buttons:** 'Submit' and 'Cancel' buttons at the bottom right.

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 "Enabling Syslog" on page 473).

Table 36-5: 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.

Parameter	Description
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 username@hostname in the Request-URI of the SIP Request-Line. For example, "2222@10.33.45.201", representing the following INVITE: <pre>INVITE sip:2222@10.33.45.201;user=phone SIP/2.0.</pre> ▪ [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 478.
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 478
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.

36.5.1 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 36-6: 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.

36.6 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 following procedure 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 "Enabling Syslog" on page 473).
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 36-5: 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 - Stopped (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, Is
11d:14h:43m:9s ( lgr_psbrdif) (2668 ) #1:OpenChannel:on Trunk -1 BChannel:1 CID=1 with Voice
11d:14h:43m:9s ( lgr_psbrdif) (2669 ) #1:OpenChannel VoiceVolume= 0, DTMFVolume = -11, Input
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.

36.7 Collecting Debug Recording Messages

To collect debug recording packets, use the open source program Wireshark. AudioCodes proprietary plug-in files for Wireshark are required.



Notes:

- The default debug recording port is 925. You can change the port in Wireshark (**Edit** menu > **Preferences** > **Protocols** > **AC DR**).
- The plug-in files are per major software release of Wireshark. For more information, contact your AudioCodes sales representative.
- The plug-in files are applicable only to Wireshark 32-bit for Windows.

➤ 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. Download the proprietary plug-in files from <https://www.audiocodes.com/library/firmware/>.
3. Copy the plug-in files to the directory in which you installed Wireshark, as follows:

Copy this file	To this folder on your PC
...\dtds\cdr.dtd	Wireshark\dtds\
...\plugins\<Wireshark ver.>*.dll	Wireshark\plugins\<Wireshark ver.>
...\tpncp\tpncp.dat	Wireshark\tpncp

4. Start Wireshark.
5. 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 — — →

Packet 2091: 11:49:43.822149 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (lgr_psbrdif)(471) InsertCfEvent

Packet 2092: 11:49:43.822180 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (lgr_psbrdif)(472) InsertCfEvent

Packet 2297: 11:49:58.091115 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(473) \nUdpRtdIngr::D1

Packet 2298: 11:49:58.091159 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(474) SIPTransaction

Packet 2299: 11:49:58.096041 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(475) UdpRtdIngr::D1

Packet 2300: 11:49:58.121103 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(476) New SIPMessage

Packet 2301: 11:49:58.121153 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (lgr_flow)(477) | (SIPTr

Packet 2302: 11:49:58.121185 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(478) SIPCall(#246)

Packet 2303: 11:49:58.121204 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(479) Resource SIPHe

Packet 2304: 11:49:58.121243 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(480) New SIPMessage

Packet 2305: 11:49:58.121269 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (lgr_flow)(481) | Tran

Packet 2306: 11:49:58.121306 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (sip_stack)(482) SIPCall(#246)

Packet 2307: 11:49:58.121330 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (lgr_flow)(483) |

Packet 2308: 11:49:58.126234 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (lgr_stk_ses)(484) <SESSION #116>

Packet 2309: 11:49:58.126266 10.33.1.5 10.33.2.30 SyslogLOCAL0.NOTICE: (lgr_flow)(485) |

Packet 2306: 11:49:58.121306 10.33.1.5 10.33.2.30 SyslogLOCAL0.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]

Packet 2306: 11:49:58.121306 10.33.1.5 10.33.2.30 SyslogLOCAL0.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]

Frame 2306: 163 bytes

Packets: 2367 Displayed: 263 Marked: 0 Dropped: 0

Profile: Default

36.8 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.

This page is intentionally left blank.

37 Creating Core Dump and Debug Files upon Device Crash

For debugging purposes, you can create a core dump file and/or debug file. These files may help you easily identify the cause of the crash. The core dump can either be included in or excluded from the debug file, or alternatively, sent separately to a TFTP server. The files can then be sent to AudioCodes support team for troubleshooting.

- **Core Dump File:** You can enable the device to send a core dump file to a remote destination upon a device crash. The core dump is a copy of the memory image at the time of the crash. It provides a powerful tool for determining the root cause of the crash. When enabled, the core dump file is sent to a user-defined TFTP server (IP address). If no address is configured, the core dump file is saved to the device's flash memory (if it has sufficient memory). The core dump file is saved as a binary file in the following name format: "core_<device name>_ver_<firmware version>_mac_<MAC address>_<date>_<time>", for example, core_acMediant_ver_680-8-4_mac_00908F099096_1-11-2014_3-29-29.

- **Debug File:** You can manually retrieve the debug file from the device and save it to a folder on your local PC. The 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 (only if enabled, no IP address has been defined, and the device has sufficient memory on its flash).
- May include additional application-proprietary debug information.

The debug file is saved as a zipped file in the following name format: "debug_<device name>_ver_<firmware version>_mac_<MAC address>_<date>_<time>", for example, debug_acMediant_ver_680-8-4_mac_00908F099096_1-11-2014_3-29-29.

The following procedure describes how to configure core dump file creation in the Web interface.

➤ **To enable core dump creation:**

1. Set up a TFTP server to where you want to send the core dump file.
2. Open the Debug Utilities page (**Maintenance** tab > **Maintenance** menu > **Debug Utilities**).

Figure 37-1: 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.

3. From the 'Enable Core Dump' drop-down list, select **Enable**.
4. 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).
5. Click **Submit**, and then reset the device with a save-to-flash for your settings to take effect.

The following procedure describes how to retrieve the debug file from the device in the Web interface.

➤ **To save the debug file from the device:**

- In the Debug Utilities page, click the **Save Debug File** button.

38 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).

38.1 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 following procedure 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 38-1: General Tab of Test Call Table

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 38-1: 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.

Parameter	Description
Route By CLI: route-by [Test_Call_RouteBy]	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].
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 216.
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 221.
Authentication Tab	

Parameter	Description
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 .
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

Parameter	Description
	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 494. ▪ [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 399). Note: 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).
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 .

38.2 Starting and Stopping Test Calls

The following procedure 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 492).

38.3 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 491), 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 38-2: 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 491).
- **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.

38.4 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 487). 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 38-3: 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**.

38.5 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.

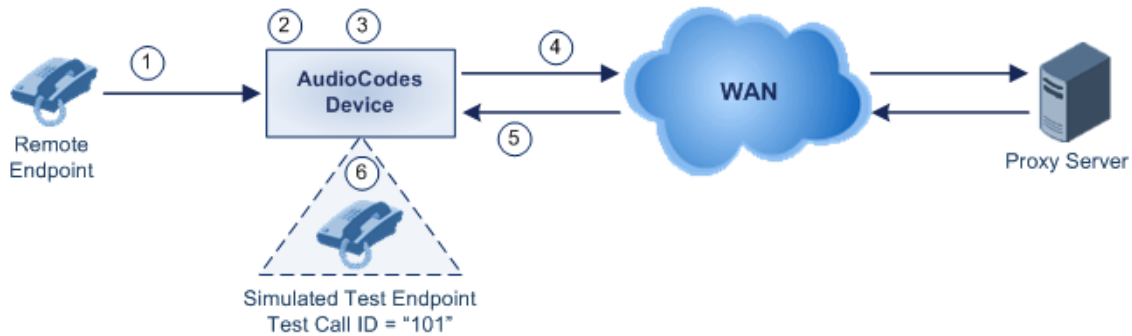
As this test call type involves an SBC call, you need to configure regular SBC rules such as classification and IP-to-IP routing. Therefore, this test call also allows you to verify correct SBC configuration.

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 38-4: 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 533):
 - a. Open the Test Call Settings page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Settings**).

Figure 38-5: Test Call Settings Page

Test Call ID	
SBC Test ID	

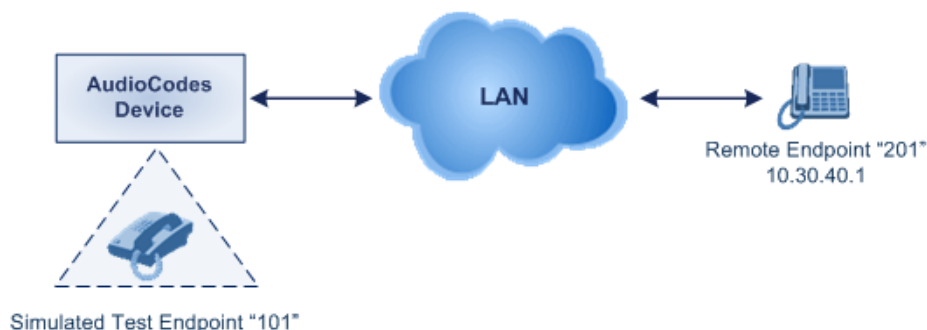
- b. In the 'Test Call ID' field, enter a prefix number for the simulated test endpoint on the device.
 - c. In the 'SBC Test ID' field, enter a called prefix number for identifying the call as an SBC test call.
 - d. Click **Submit**.
2. Configure regular SBC call processing rules for called number prefix "101", such as classification and IP-to-IP routing through a proxy server.

38.6 Test Call Configuration Examples

Below are a few examples of test call configurations.

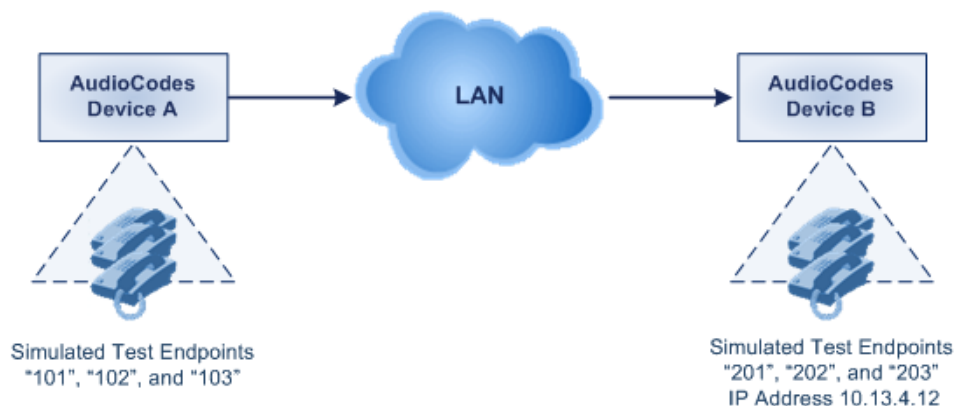
- **Single Test Call Scenario:** This example describes the configuration of a simple test call scenario that includes a single test call between a simulated test endpoint on the device and a remote endpoint.

Figure 38-6: 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.

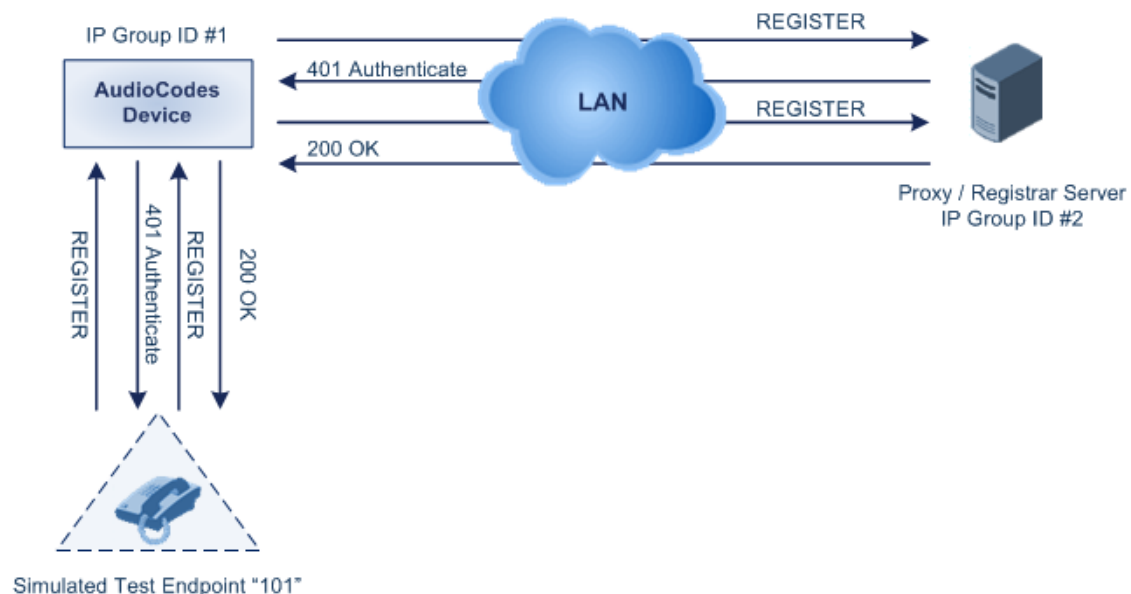
Figure 38-7: Batch Test Call Example



- 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 password) 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 38-8: 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**

This page is intentionally left blank.

Part X

Appendix

39 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 39-1: 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 5-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]).	

Notation	Description
[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 <i>n</i> must be less than the value <i>m</i>. - Only numerical ranges are supported (not alphabetical letters). - For suffix ranges, the starting (<i>n</i>) and ending (<i>m</i>) 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 <i>d[n,m]e</i> and <i>d[n-m]e</i> 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.

Notation	Description												
Special ASCII Characters	The device does not support the use of ASCII characters in manipulation rules and therefore, for LDAP-based queries, the device can use the hexadecimal (HEX) format of the ASCII characters for phone numbers instead. The HEX value must be preceded by a backslash “\”. For example, you can configure a manipulation rule that changes the number +49 (7303) 165-xxxxx to +49 \287303\29 165-xxxxx, where \28 is the ASCII HEX value for “(” and \29 is the ASCII HEX value for “)”. The manipulation rule in this example would denote the parenthesis in the destination number prefix using “x” wildcards (e.g., xx165xxxxx#); the prefix to add to the number would include the HEX values (e.g., +49 \287303\29 165-). Below is a list of common ASCII characters and their corresponding HEX values: <table><thead><tr><th>ASCII Character</th><th>HEX Value</th></tr></thead><tbody><tr><td>*</td><td>\2a</td></tr><tr><td>(</td><td>\28</td></tr><tr><td>)</td><td>\29</td></tr><tr><td>\</td><td>\5c</td></tr><tr><td>/</td><td>\2f</td></tr></tbody></table>	ASCII Character	HEX Value	*	\2a	(	\28	)	\29	\	\5c	/	\2f
ASCII Character	HEX Value												
*	\2a												
(	\28												
)	\29												
\	\5c												
/	\2f												



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.

This page is intentionally left blank.

40 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.

40.1 Management Parameters

This section describes the device's management-related parameters.

40.1.1 General Parameters

The general management parameters are described in the table below.

Table 40-1: General Management Parameters

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 55.
[CustomerSN]	Defines a serial number (S/N) for the device. Note: The device's original S/N is automatically added at the end of the configured S/N. For example, if the original S/N is 8906721 and the configured S/N is "abc123", the resultant S/N is "abc1238906721".

40.1.2 Web Parameters

The Web parameters are described in the table below.

Table 40-2: 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.

Parameter	Description
	[1] = Enable - Web access is through any network interface. Note: For the parameter to take effect, a device reset is required.
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 inactivity of a logged-in user in the Web interface, after which the user is automatically logged off the Web session. In other words, the session expires when the user has not performed any operations (activities) in the Web interface for the configured duration. The valid value is 0-100000, where 0 means no timeout. The default is 15. Note: You can also configure the functionality per user in the Web Users table (see Advanced User Accounts Configuration on page 49) which overrides this global setting.
Web: Deny Access On Fail Count [DenyAccessOnFailCount]	Defines the maximum number of failed login attempts, after which the 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.

Parameter	Description
	▪ [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.

Parameter	Description
[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]	Enables the device to restore the default management users: ▪ Security Administrator user (username "Admin"; password "Admin") ▪ Monitor user (username "User"; password "User") In addition, all other users that may have been configured (in the Web Users table) are deleted. ▪ [0] = (Default) Disabled. Currently configured users (usernames and passwords) are retained. ▪ [1] = Enabled. Default users are restored (see description above) and all other configured users are deleted. Notes: ▪ For the parameter to take effect, a device reset is required. ▪ In addition to the ini file (see above), you can also restore the default user accounts through the following management platforms: ✓ SNMP (restores default users and retains other configured users: 1) Set acSysGenericINILine to WEBPasswordControlViaSNMP = 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: <pre>Username acSysWEBAccessUserName: old/pass/new Password acSysWEBAccessUserCode: username/old/new</pre>
[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: <pre>[WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text [WelcomeMessage]</pre>

Parameter	Description
	For Example: <pre> FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "*****" ; WelcomeMessage 2 = "***** This is a Welcome message ***" ; WelcomeMessage 3 = "*****" ; </pre> 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 [HAAUnitIdName]	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.

40.1.3 Telnet Parameters

The Telnet parameters are described in the table below.

Table 40-3: Telnet Parameters

Parameter	Description
Web: Embedded Telnet Server CLI: telnet [TelnetServerEnable]	Enables the device's embedded Telnet server. [0] Disable [1] Enable Unsecured (default) [2] Enable Secured Note: Only management users with Security Administrator level, Administrator level, or Master level can access the device through Telnet (see "Configuring Web User Accounts" on page 46).
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 Timeout CLI: idle-timeout [TelnetServerIdleDisconnect]	Defines the timeout (in minutes) for disconnection of an idle Telnet session. When 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

Parameter	Description
	will not take effect.
[CLIPrivPass]	Defines the password to access the Enable configuration mode in the CLI. The valid value is a string of up to 50 characters. The default is "Admin". Note: The password is case-sensitive.

40.1.4 ini File Parameters

The parameters relating to ini-file management are described in the table below.

Table 40-4: 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 (*).

40.1.5 SNMP Parameters

The SNMP parameters are described in the table below.

Table 40-5: SNMP Parameters

Parameter	Description
Web: Enable SNMP CLI: disable [DisableSNMP]	Enables and disables SNMP. [0] Enable = (Default) SNMP is enabled. [1] Disable = SNMP is disabled and no traps are sent. Note: For this parameter to take effect, a device reset is required.
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.
[ChassisPhysicalAlias]	Defines the 'alias' name object for the physical entity as specified by a network manager, and provides a non-volatile 'handle' for the physical entity. The valid range is a string of up to 255 characters.
[ChassisPhysicalAssetID]	Defines the user-assigned asset tracking identifier object for the device's chassis as specified by an EMS, and provides non-volatile storage of this information. The valid range is a string of up to 255 characters.
[ifAlias]	Defines the textual name of the interface. The value is equal to the ifAlias SNMP MIB object. The valid range is a string of up to 64 characters.

Parameter	Description
[SendKeepAliveTrap]	Enables the device to send NAT keep-alive traps to the port of the SNMP network management station (e.g., AudioCodes EMS). This is used for NAT traversal, and allows SNMP communication with AudioCodes EMS management platform, located in the WAN, when the device is located behind NAT. It is needed to keep the NAT pinhole open for the SNMP messages sent from EMS to the device. The device sends the trap periodically - every 9/10 of the time configured by the NATBindingDefaultTimeout parameter. The trap that is sent is ackKeepAlive. For more information on the SNMP trap, refer to the <i>SNMP Reference Guide</i>. ▪ [0] = (Default) Disable ▪ [1] = Enable For configuring the port number, use the KeepAliveTrapPort parameter. Note: For this parameter to take effect, a device reset is required.
[KeepAliveTrapPort]	Defines the port of the SNMP network management station to which the device sends keep-alive traps. The valid range is 0 to 65534. The default is 1161. To enable NAT keep-alive traps, use the SendKeepAliveTrap parameter.
[PM_EnableThresholdAlarms]	Enables the sending of the SNMP trap event, acPerformanceMonitoringThresholdCrossing which is sent every time the threshold (high and low) of a Performance Monitored object (e.g., acPMMediaRealmAttributesMediaRealmBytesTxHighThreshold) is crossed. ▪ [0] = (Default) Disable ▪ [1] = Enable
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.
[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.

Parameter	Description
[ActiveAlarmTableMaxSize]	Defines the maximum number of currently active alarms that can be displayed in the Active Alarms table. When the table reaches this user-defined maximum capacity (i.e., full), the device sends the SNMP trap event, acActiveAlarmTableOverflow. If the table is full and a new alarm is raised by the device, the new alarm is not displayed in the table. The valid range is 100 to 1000. The default is 250. For more information on the Active Alarms table, see Viewing Active Alarms on page 439. Note: - For the parameter to take effect, a device reset is required. - To clear the acActiveAlarmTableOverflow trap, you must reset the device. The reset also deletes all the alarms in the Active Alarms table.
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.

Parameter	Description
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 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 - Read Only configure system > snmp > ro-community-string [SNMPReadOnlyCommunityString_x]	Defines a read-only SNMP community string. Up to five read-only community strings can be configured. The valid value is a string of up to 19 characters that can include only the following: - Upper- and lower-case letters (a to z, and A to Z) - Numbers (0 to 9) - Hyphen (-) - Underline (_) For example, "Public-comm_string1". The default is "public".
Community String - Read / Write configure system > snmp > rw-community-string [SNMPReadWriteCommunityString_x]	Defines a read-write SNMP community string. Up to five read-write community strings can be configured. The valid value is a string of up to 19 characters that can include only the following: - Upper- and lower-case letters (a to z, and A to Z) - Numbers (0 to 9) - Hyphen (-) - Underline (_) For example, "Private-comm_string1". The default is "private".
Trap Community String configure system > snmp trap > community-string [SNMPTrapCommunityString]	Defines the community string for SNMP traps. The valid value is a string of up to 19 characters that can include only the following: - Upper- and lower-case letters (a to z, and A to Z) - Numbers (0 to 9) - Hyphen (-) - Underline (_) For example, "Trap-comm_string1". The default 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

Parameter	Description
	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 <i>parameter</i> 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 74.

40.1.6 Serial Parameters

The RS-232 serial parameters are described in the table below.

Table 40-6: Serial Parameters

Parameter	Description
[DisableRS232]	Enables the device's RS-232 (serial) port. ▪ [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 <i>Installation Manual</i>. 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.

40.1.7 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 395.

Table 40-7: 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 <i>ini</i> file. ▪ [0] = Disable - parameters not included in the downloaded <i>ini</i> 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 <i>ini</i> file upload (not applicable if the <i>ini</i> file is loaded using BootP).
[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 <i>DConvert Utility User's Guide</i>. 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 <i>DConvert Utility User's Guide</i>).
[UserInfoFileName]	Defines the name (and path) of the file containing the User Information data.

40.1.8 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

Table 40-8: Automatic Update of Software and Configuration Files Parameters

Parameter	Description
General Automatic Update Parameters	
CLI: configure system/automatic-update/update-firmware [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.
CLI: configure system > automatic-update > update-frequency [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.
CLI: configure system > automatic-update > predefined-time [AutoUpdatePredefinedTime]	Defines schedules (time of day) for automatic updates. The format syntax of this parameter is 'hh:mm', where hh denotes the hour and mm the minutes. The value must be enclosed in single apostrophes. 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.

Parameter	Description
CLI: automatic-update > http-user-agent [AupdHttpUserAgent]	Defines the information sent in the HTTP User-Agent 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. The information can include any user-defined string or the following 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 by the ini file parameter, INIFileVersion or CLI command, configuration-version The device automatically populates these tag variables with actual values in the sent header. By default, the device sends the following in the User-Agent header: <pre>User-Agent: Mozilla/4.0 (compatible; AudioCodes; <NAME>;<VER>;<MAC>;<CONF>)</pre> For example, if you set AupdHttpUserAgent = MyWorld-<NAME>;<VER>(<MAC>), the device sends the following User-Agent header: <pre>User-Agent: MyWorld- Mediant;6.80.200.001(00908F1DD0D3)</pre> Notes: - The variable tags are case-sensitive. - If you configure the parameter with the <CONF> variable tag, you must reset the device with a burn-to-flash for your settings to take effect. - The tags can be defined in any order.
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.
[AUPDDigestUsername]	Defines the username for digest (MD5 cryptographic hashing) access authentication with the HTTP server used for the Automatic Update feature. The valid value is a string of up to 50 characters. By default, no value is defined.
[AUPDDigestPassword]	Defines the password for digest (MD5 cryptographic hashing) access authentication with the HTTP server used for the Automatic Update feature. The valid value is a string of up to 50 characters. By default, no value is defined.
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

Parameter	Description
CLI: configure system > automatic-update > crc-check regular [AUPDCheckIfIniChanged]	Enables the device to perform cyclic redundancy checks (CRC) on downloaded configuration files (.ini) during the Automatic Update process. The CRC checks whether the content (raw data) of the downloaded file is different to the content of the previously downloaded file from the previous Automatic Update process. The device compares the CRC check value (code) result with the check value of the previously downloaded file. If the check values are identical, it indicates that the file has no new configuration settings, and the device discards the file. If the check values are different, the device installs the downloaded file and applies the new configuration settings. ▪ [0] = (Default) Disable - the device does not perform CRC and installs the downloaded file regardless. ▪ [1] = Enable CRC for the entire file, including line order (i.e., same text must be on the same lines). If there are differences between the files, the device installs the downloaded file. If there are no differences, the device discards the newly downloaded file. ▪ [2] = Enable CRC for individual lines only. Same as option [1], except that the CRC ignores the order of lines (i.e., same text can be on different lines).
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 IniFileName. ▪ [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. Note: If you use this parameter in an ini file for periodic automatic provisioning with non-HTTP (e.g., TFTP) and without CRC, the device resets after every file download.

Parameter	Description
Software/Configuration File URL Path for Automatic Update Parameters	
CLI: firmware [CmpFileURL]	Defines the name of the <i>cmp</i> file and the path to the server (IP address or FQDN) from where the device can load the <i>cmp</i> file and update itself. The <i>cmp</i> file can be loaded using HTTP/HTTPS. For example: <code>http://192.168.0.1/filename</code> Notes: - For this parameter to take effect, a device reset is required. - When this parameter is configured, the device always loads the <i>cmp</i> file after it is reset. - The <i>cmp</i> file is validated before it's burned to flash. The checksum of the <i>cmp</i> file is also compared to the previously burnt checksum to avoid unnecessary resets. - The maximum length of the URL address is 255 characters.
CLI: voice-configuration [IniFileURL]	Defines the name of the <i>ini</i> file and the path to the server (IP address or FQDN) on which it is located. The <i>ini</i> file can be loaded using HTTP/HTTPS. For example: <code>http://192.168.0.1/filename</code> <code>http://192.8.77.13/config<MAC></code> <code>https://<username>:<password>@<IP address>/<file name></code> Notes: - For this parameter to take effect, a device reset is required. - When using HTTP or HTTPS, the date and time of the <i>ini</i> file are validated. Only more recently dated <i>ini</i> files are loaded. - The case-sensitive string, "<MAC>" can be used in the file name for instructing the device to replace it with the device's MAC address. For more information, see MAC Address Automatically Inserted in Configuration File Name on page 425. - 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: <code>http://server_name/file</code>, <code>https://server_name/file</code>. 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: <code>http://server_name/file</code>, <code>https://server_name/file</code>. 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.

Parameter	Description
[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.

40.2 Networking Parameters

This subsection describes the device's networking parameters.

40.2.1 Ethernet Parameters

The Ethernet parameters are described in the table below.

Table 40-9: Ethernet Parameters

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 105.
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 Ethernet Port Groups on page 107. Note: For this parameter to take effect, a device reset is required.

40.2.2 Multiple VoIP Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

Table 40-10: 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: <pre>[InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName, InterfaceTable_PrimaryDNSServerIPAddress, InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingDevice; [/InterfaceTable]</pre> For a detailed description of this table, see "Configuring IP Network Interfaces" on page 112. 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) ▪ [0] = Control Note: For this parameter to take effect, a device reset is required.

40.2.3 Routing Parameters

The IP network routing parameters are described in the table below.

Table 40-11: 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: Send and Receive ICMP Redirect 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.

Parameter	Description
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: <pre>[StaticRouteTable] FORMAT StaticRouteTable_Index = StaticRouteTable_DeviceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [\StaticRouteTable]</pre> For a description of this parameter, see "Configuring Static IP Routes" on page 119.

40.2.4 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

Table 40-12: 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: <pre>[DiffServToVlanPriority] FORMAT DiffServToVlanPriority_Index = DiffServToVlanPriority_DiffServ, DiffServToVlanPriority_VlanPriority; [\DiffServToVlanPriority]</pre> For example: <pre>DiffServToVlanPriority 0 = 46, 6; DiffServToVlanPriority 1 = 40, 6; DiffServToVlanPriority 2 = 26, 4; DiffServToVlanPriority 3 = 10, 2;</pre> For a description of this table, see Configuring Quality of Service on page 123. 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 275. 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: 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 275. 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.

40.2.5 NAT Parameters

The Network Address Translation (NAT) parameters are described in the table below.

Table 40-13: NAT Parameters

Parameter	Description
Web/EMS: NAT Mode CLI: disable-NAT-traversal [NATMode]	Enables the NAT feature for media when the device communicates with UAs located behind NAT. [0] Auto-Detect = NAT is performed only if necessary. If the UA is identified as being located behind NAT, the device sends the media packets to the public IP address:port obtained from the source address of the first media packet received from the UA. Otherwise, the packets are sent using the IP address:port obtained from the address in the first received SIP message. Note that if the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA, does it determine whether the UA is behind NAT. [1] NAT Is Not Used = (Default) NAT feature is disabled. The device always sends the media packets to the remote UA using the IP address:port obtained from the first received SIP message. [2] NAT Is Used = NAT is always performed. The device always sends the media packets to the remote UA using the source address obtained from the first media packet from the UA. In this mode, the device does not send any packets until it receives the first packet from the UA (in order to obtain the IP address). For more information on handling calls from UAs behind NAT, see "First Incoming Packet Mechanism" on page 133.
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.
EMS: Binding Life Time [NATBindingDefaultTime out]	The device sends SNMP keep-alive traps periodically - every 9/10 of the time configured by this parameter (in seconds). Therefore, the parameter is applicable only if the SendKeepAliveTrap parameter is set to 1. The parameter is used to allow SNMP communication with AudioCodes EMS management platform, located in the WAN, when the device is located behind NAT. It is needed to keep the NAT pinhole open for the SNMP messages sent from EMS to the device. The valid range is 0 to 2,592,000. The default is 30. Note: For this parameter to take effect, a device reset is required.

40.2.6 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table 40-14: DNS Parameters

Parameter	Description
Internal DNS Table	
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. 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, ; For a detailed description of this table, see "Configuring the Internal DNS Table" on page 126.
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;

Parameter	Description
	[\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 127 .

40.2.7 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

Table 40-15: DHCP Parameters

Parameter	Description
Web: Enable DHCP [DHCPEnable]	Enables DHCP client 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 417. - 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 <i>ini</i> file.
[DHCPSpeedFactor]	Defines the device's DHCP renewal speed for a leased IP address from a DHCP server. [0] = Disable [1] = (Default) Normal [2] to [10] = Fast When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is 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.
DHCP Servers Table	
Web: DHCP Servers Table CLI: configure voip > dhcp server <index> [DhcpServer]	Defines the device's embedded DHCP server. The format of the ini file table parameter is as follows: <pre>[DhcpServer] FORMAT DhcpServer_Index = DhcpServer_InterfaceName, DhcpServer_StartIPAddress, DhcpServer_EndIPAddress, DhcpServer_SubnetMask, DhcpServer_LeaseTime, DhcpServer_DNSServer1, DhcpServer_DNSServer2, DhcpServer_NetbiosNameServer, DhcpServer_NetbiosNodeType, DhcpServer_NTPServer1, DhcpServer_NTPServer2, DhcpServer_TimeOffset, DhcpServer_TftpServer, DhcpServer_BootFileName, DhcpServer_ExpandBootfileName, DhcpServer_OverrideRouter, DhcpServer_SipServer, DhcpServer_SipServerType; [\DhcpServer]</pre> For a detailed description of this table, see Configuring the Device's DHCP Server on page 157.

Parameter	Description
DHCP Vendor Class Table	
Web: DHCP Vendor Class table CLI: configure voip > dhcp vendor-class [DhcpVendorClass]	Defines Vendor Class Identifier (VCI) names (DHCP Option 60) for the device's DHCP server. Only if the DHCPDiscover request message, received from the DHCP client, contains this value does the device provide DHCP services. The format of the ini file table parameter is as follows: <pre>[DhcpVendorClass] FORMAT DhcpVendorClass_Index = DhcpVendorClass_DhcpServerIndex, DhcpVendorClass_VendorClassId; [\DhcpVendorClass]</pre> For a detailed description of this table, see Configuring the Vendor Class Identifier on page 161.
DHCP Option Table	
Web: DHCP Option table CLI: configure voip > dhcp option [DhcpOption]	Defines additional DHCP Options that the device's DHCP server can use to service its DHCP clients. The format of the ini file table parameter is as follows: <pre>[DhcpOption] FORMAT DhcpOption_Index = DhcpOption_DhcpServerIndex, DhcpOption_Option, DhcpOption_Type, DhcpOption_Value, DhcpOption_ExpandValue; [\DhcpOption]</pre> For a detailed description of this table, see Configuring Additional DHCP Options on page 162.
DHCP Static IP Table	
Web: DHCP Static IP table CLI: configure voip > dhcp static-ip <index> [DhcpStaticIP]	Defines static "reserved" IP addresses that the device's DHCP server allocates to specific DHCP clients defined by MAC address. The format of the ini file table parameter is as follows: <pre>[DhcpStaticIP] FORMAT DhcpStaticIP_Index = DhcpStaticIP_DhcpServerIndex, DhcpStaticIP_IPAddress, DhcpStaticIP_MACAddress; [\DhcpStaticIP]</pre> For a detailed description of this table, see Configuring Static IP Addresses for DHCP Clients on page 164.

40.2.8 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

Table 40-16: 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 99.	
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. Note: The offset setting is applied only on the hour. For example, if you configure this parameter at 15:42, the device applies the setting only at 16:00.
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

Parameter	Description
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 - <i>mm:dd:hh:mm</i>, where: ✓ <i>mm</i> denotes month ✓ <i>dd</i> denotes date of the month ✓ <i>hh</i> denotes hour ✓ <i>mm</i> denotes minutes For example, "05:01:08:00" denotes daylight saving starting from May 1 at 8 A.M. - Day of month - <i>mm:day/wk:hh:mm</i>, where: ✓ <i>mm</i> denotes month (e.g., 04) ✓ <i>day</i> denotes day of week (e.g., FRI) ✓ <i>wk</i> denotes week of the month (e.g., 03) ✓ <i>hh</i> denotes hour (e.g., 23) ✓ <i>mm</i> 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. Note: The offset setting is applied only on the hour. For example, if you configure this parameter at 15:42, the device applies the setting only at 16:00.

40.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

40.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

Table 40-17: 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

40.3.2 SIP Test Call Parameters

The SIP Signaling Test Call parameters are described in the table below.

Table 40-18: 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 (<i>ID</i>) 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 487.

40.3.3 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table 40-19: 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.

Parameter	Description
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.
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 enabled.
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 3164 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).

Parameter	Description
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.
<pre>configure voip > sip- definition settings > time-zone-format</pre> [TimeZoneFormat]	Defines the time zone that is displayed with the timestamp in CDRs. The timestamp appears in the CDR fields "Setup Time", "Connect Time", and "Release Time". The valid value is a string of up to six characters. The default is UTC. For example, if you configure the parameter TimeZoneFormat = GMT+11, the timestamp in CDRs are generated with the following time zone display: <pre>17:47:45.411 GMT+11 Sun Jan 03 2018</pre> Note: The time zone is only for display purposes; it does not configure the actual time zone.
<pre>configure system > cdr > non- call-cdr-rprt</pre> [EnableNonCallCdr]	Enables creation of CDR messages for non-call SIP dialogs (such as SUBSCRIBE, OPTIONS, and REGISTER). [0] = (Default) Disable [1] = Enable
Web/EMS: Debug Level CLI: configure system/logging/debug-level [GwDebugLevel]	Enables Syslog debug reporting and logging level. [0] No Debug = (Default) Debug is disabled. [1] Basic = Sends debug logs of incoming and outgoing SIP messages. [5] Detailed = Sends debug logs of incoming and outgoing SIP message as well as many other logged processes. Note: When debug reporting is enabled, in order to view Syslog messages with Wireshark, you need to install AudioCodes Wireshark plug-in (acsyslog.dll). Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and are displayed using the 'acsyslog' filter instead of the regular 'syslog' filter.
Web: Syslog Optimization CLI: configure system/logging/syslog- optimization [SyslogOptimization]	Enables the device to accumulate and bundle multiple debug messages into a single UDP packet and then send it to a Syslog server. The benefit of this feature is that it reduces the number of UDP Syslog packets, thereby improving (optimizing) CPU utilization. [0] Disable (default) [1] Enable Note: The size of the bundled message is configured by the

Parameter	Description
	MaxBundleSyslogLength parameter.
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 enabled.
Web: Syslog CPU Protection CLI: configure system/logging/syslog-cpu- protection [SyslogCpuProtection]	Enables the protection of the device's CPU resources during debug reporting, ensuring voice traffic is unaffected. If CPU resources drop (i.e., high CPU usage) to a critical level (threshold), the device automatically lowers the debug level to free up CPU resources that were required for the previous debug-level functionality. When sufficient CPU resources become available again, the device increases the debug level. The threshold is configured by the 'Debug Level High Threshold' parameter (see below). ▪ [0] Disable ▪ [1] Enable (default)
Web: Debug Level High Threshold CLI: debug-level-high- threshold [DebugLevelHighThreshold]	Defines the threshold (in percentage) for automatically switching to a different debug level, depending on CPU usage. The parameter is applicable only if the 'Syslog CPU Protection' parameter is enabled. The valid value is 0 to 100. The default is 90. The debug level is changed upon the following scenarios: ▪ CPU usage equals threshold: Debug level is reduced one level. ▪ CPU usage is at least 5% greater than threshold: Debug level is reduced another level. ▪ CPU usage is 5 to 19% less than threshold: Debug level is increased by one level. ▪ CPU usage is at least 20% less than threshold: Debug level is increased by another level. For example, assume that the threshold is set to 70% and the Debug Level to Detailed (5). When CPU usage reaches 70%, the debug level is reduced to Basic (1). When CPU usage increases by 5% or more than the threshold (i.e., greater than 75%), the debug level is disabled - No Debug (0). When the CPU usage decreases to 5% less than the threshold (e.g., 65%), the debug level is increased to Basic (1). When the CPU usage decreases to 20% less than the threshold (e.g., 50%), the debug level changes to Detailed (5). Note: The device does not increase the debug level to a level that is higher than what you configured for the 'Debug Level' parameter.
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)

Parameter	Description
	▪ [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 CLI: config-system > logging > activity-log [ActivityListToLog]	Defines the operations (activities) in the Web interface that are reported to a Syslog server. ▪ [pvc] Parameters Value Change = Changes made on-the-fly to parameters. Note that the <i>ini</i> file parameter, EnableParametersMonitoring can also be used to set this option, using values [0] (disable) or [1] (enable). ▪ [afi] Auxiliary Files Loading = Loading of auxiliary files. ▪ [dr] Device Reset = Resetting of device through the Maintenance Actions page. Note: For this option to take effect, a device reset is required. ▪ [fb] Flash Memory Burning = Saving configuration with burn to flash (in Maintenance Actions page). ▪ [swu] Device Software Update = Software updates (i.e., loading of cmp file) through the Software Upgrade Wizard. ▪ [ard] Access to Restricted Domains = Access to restricted 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 = Attempts to access the Web interface with a false or empty username 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 = Web login and logout attempts. Note: For the <i>ini</i> file parameter, enclose values in single quotation marks, for example: ActivityListToLog = 'pvc', 'afi', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc'.
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.

Parameter	Description
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 (default) ▪ [1] Enable Note: For the parameter to take effect, a device reset is required.
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 Debug Recordings" on page 476.

40.3.4 Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

Table 40-20: RAI Parameters

Parameter	Description
[EnableRAI]	Enables Resource Available Indication (RAI) alarm generation if the device's busy endpoints exceed a user-defined threshold, configured by the RAIHighThreshold parameter. When enabled and the threshold is crossed, the device sends the SNMP trap, acBoardCallResourcesAlarm. ▪ [0] = (Default) Disable ▪ [1] = Enable Note: For this parameter to take effect, a device reset is required.
[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.

Parameter	Description
	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.

40.4 HA Parameters

The High Availability (HA) parameters are described in the table below.

Table 40-21: HA Parameters

Parameter	Description
HA Device Name <pre>configure system > high-availability > unit-id-name</pre> [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. The default value is "Device 1" for the active device and "Device 2" for the redundant device.
HA Remote Address <pre>configure system > high-availability > remote-address</pre> [HARemoteAddress]	Defines the Maintenance interface address of the redundant device in the HA system. By default, no value is defined.
HA Revertive <pre>configure system > high-availability > revertive-mode</pre> [HAREvertiveEnabled]	Enables HA switchover based on HA priority. ▪ [0] Disable (default) = A switchover over to the redundant device is done only if a failure occurs in the currently active device. ▪ [1] Enable = A switchover over to the redundant device is done if a failure occurs in the currently active device. However, a switchover to the device with the highest priority (configured by the HAPriority parameter) occurs whenever the device recovers from a failure. Therefore, whenever possible, the highest priority device is the active one. For more information on the HA switchover mechanism, see Device Switchover upon Failure on page 366.
HA Priority <pre>configure system > high-availability > priority</pre> [HAPriority]	Defines the priority of the device used in the HA Revertive mechanism. The valid value is 1 (lowest priority) to 10 (highest priority). The default is 5. Note: ▪ The parameter is applicable only if you configure the 'HA Revertive' parameter to Enable. ▪ You must configure each device in the HA system with different parameter values (priorities).

Parameter	Description
HA Monitoring Parameters	
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.
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.

40.5 Security Parameters

This subsection describes the device's security parameters.

40.5.1 General Security Parameters

The general security parameters are described in the table below.

Table 40-22: 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 139.

Parameter	Description
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 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.

Parameter	Description
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 SRTCP [TimeoutToRelatchSRTCPMsec]	Defines a period (msec) during which if no packets are received from the current SRTCP 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/SRTCP 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.

40.5.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

Table 40-23: 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.

Parameter	Description
Web/: HTTPS Cipher String CLI: https-cipher-string [HTTSPCipherString]	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 'TLS for Remote Device Management' on page 96.

40.5.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

Table 40-24: SRTP Parameters

Parameter	Description
Web: Media Security CLI: media-security-enable [EnableMediaSecurity]	Enables Secure Real-Time Transport Protocol (SRTP). [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-bhviior	Global parameter that defines the handling of SRTP (when the EnableMediaSecurity parameter

Parameter	Description
[MediaSecurityBehaviour]	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 275. 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 275. 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 275. 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: Offered SRTP Cipher Suites CLI: offer-srtp-cipher [SRTPofferedSuites]	Defines the offered crypto suites (cipher encryption algorithms) for SRTP. ▪ [0] All = (Default) All available crypto suites. ▪ [1] 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] 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	Enables encryption on transmitted RTP packets in

Parameter	Description
Packets CLI: RTP-encryption-disable-tx [RTPEncryptionDisableTx]	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
SRTP Tunneling Authentication for RTP configure voip > media security > srtp-tnl-vld-rtp-auth [SRPTunnelingValidateRTPRxAuthentication]	Enables validation of SRTP tunneling authentication for RTP. ▪ [0] Disable = (Default) The device does not perform any validation and forwards the packets as is. ▪ [1] Enable = The device validates the packets (e.g., sequence number) and if successful, forwards the packets. If validation fails, it drops the packets. Note: The parameter is applicable only to SRTP-to-SRTP calls and when both endpoints use the same authentication keys.
SRTP Tunneling Authentication for RTCP configure voip > media security > srtp-tnl-vld-rtcp-auth [SRPTunnelingValidateRTCPRxAuthentication]	Enables validation of SRTP tunneling authentication for RTCP. ▪ [0] Disable = (Default) The device does not perform any validation and forwards the packets as is. ▪ [1] Enable = The device validates the packets (e.g., sequence number) and if successful, forwards the packets. If validation fails, it drops the packets. Note: The parameter is applicable only to SRTP-to-SRTP calls and when both endpoints use the same authentication keys.
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 275. 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.

40.5.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

Table 40-25: TLS Parameters

Parameter	Description
Web: TLS Contexts Table CLI: configure system > tls # [TLSContexts]	Defines SSL/TLS certificates. The format of the ini file table parameter is as follows: <pre>[TLSContexts] FORMAT TLSContexts_Index = TLSContexts_Name, TLSContexts_TLSVersion, TLSContexts_ServerCipherString, TLSContexts_ClientCipherString, TLSContexts_OcspEnable, TLSContexts_OcspServerPrimary, TLSContexts_OcspServerSecondary, TLSContexts_OcspServerPort, TLSContexts_OcspDefaultResponse; [\TLSContexts]</pre> For a detailed description of this table, see Configuring TLS Certificate Contexts on page 85.
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]	Defines the device's mode of operation regarding mutual authentication and certificate verification for TLS connections. ▪ [0] Disable = (Default) ✓ Device acts as a client: Verification of the server's certificate depends on the VerifyServerCertificate parameter. ✓ Device acts as a server: The device does not request the client certificate. ▪ [1] Enable = ✓ Device acts as a client: Verification of the server certificate is required to establish the TLS connection. ✓ Device acts as a server: The device requires the receipt and verification of the client certificate to establish the TLS connection. Notes: ▪ For this parameter to take effect, a device reset is required. ▪ This feature can be configured per SIP Interface (see Configuring SIP Interfaces on page 236). ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName.

Parameter	Description
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 the device receives a remote certificate and this parameter is not disabled, the IP address from which the certificate is received is compared with the addresses defined for the Proxy Sets. If no Proxy Set with the source address is found, the connection is refused. Otherwise, the value of SubjectAltName field in the certificate is compared with the addresses\ DNS Names of the classified Proxy Set. If a match is found for any of the configured Proxies, the TLS 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 SIPSPRequireClientCertificate 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 extendedKeyUsage) 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

Parameter	Description
Web: TLS Remote Subject Name CLI: tls-rmt-sub-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: 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 (acCertificateExpiryNotification) 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.

40.5.5 SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

Table 40-26: 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	Defines the maximum packet size (in bytes) for SSH packets. The valid value is 582 to 35000. The default is 35000.

Parameter	Description
[SSHMaxBinaryPacketSize]	
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 [SSHMaxLoginAttempts]	Defines the maximum SSH login attempts allowed for entering an incorrect password by an administrator before the SSH session is rejected. The valid range is 1 to 5. The default is 3. Note: The new setting takes effect only for new subsequent SSH connections.

40.5.6 IDS Parameters

The Intrusion Detection System (IDS) parameters are described in the table below.

Table 40-27: 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 [IDSArmClearPeriod]	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 IDSArmClearPeriod is set to 20 sec and the Threshold Window is set to 15 sec, the IDSArmClearPeriod 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 145.

Parameter	Description
IDS Rule Table	
Web: IDS Rule Table [IDSRule]	Defines rules for IDS Policies. The format of the ini file parameter is: <pre>[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]</pre> For a detailed description of this table, see "Configuring IDS Policies" on page 145.
IDS Match Table	
Web: IDS Match Table [IDSMatch]	Defines target rules per IDS Policy. The format of the ini file parameter is: <pre>[IDSMatch] FORMAT IDSMatch_Index = IDSMatch_SIPInterface, IDSMatch_ProxySet, IDSMatch_Subnet, IDSMatch_Policy; [\IDSMatch]</pre> For a detailed description of this table, see "Assigning IDS Policies" on page 149.

40.6 Quality of Experience Parameters

The Quality of Experience (QoE) parameters are described in the table below.

Table 40-28: 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 primary 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: Redundant Server IP CLI: configure voip > qoe configuration > set secondary-server-ip [QOESecondaryServerIp]	Defines the IP address of the secondary SEM server to where the quality experience reports are sent. This is applicable when the SEM/EMS server is in Geographical Redundancy HA mode. Note: For this parameter to take effect, a device reset is required.
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.
QoE Connection by TLS configure voip > qoe configuration > tls-enable [QOEEnableTLS]	Enables a TLS connection with the SEM server. ▪ [0] Disable (default) ▪ [1] Enable Note: For the parameter to take effect, a device reset is required.
Quality of Experience Profile Table	
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 216.
Quality of Experience Color Rules Table	

Parameter	Description
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: <pre>[QOECOLORRules] FORMAT QOECOLORRules_Index = QOECOLORRules_QoeProfile, QOECOLORRules_ColorRuleIndex, QOECOLORRules_monitoredParam, QOECOLORRules_direction, QOECOLORRules_profile, QOECOLORRules_GreenYellowThreshold, QOECOLORRules_GreenYellowHysteresis, QOECOLORRules_YellowRedThreshold, QOECOLORRules_YellowRedHysteresis; [QOECOLORRules]</pre> For a detailed description of this table, see "Configuring Quality of Experience Profiles" on page 216.
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: <pre>[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]</pre> For a detailed description of this table, see "Configuring Bandwidth Profiles" on page 221. 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: <pre>[MediaEnhancementProfile] FORMAT MediaEnhancementProfile_Index = MediaEnhancementProfile_ProfileName; [MediaEnhancementProfile]</pre> For a detailed description of this table, see "Configuring Media Enhancement Profiles" on page 224.
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: <pre>[MediaEnhancementRules] FORMAT MediaEnhancementRules_Index = MediaEnhancementRules_MediaEnhancementProfile, MediaEnhancementRules_RuleIndex, MediaEnhancementRules_Trigger, MediaEnhancementRules_Color, MediaEnhancementRules_ActionRule, MediaEnhancementRules_ActionValue; [MediaEnhancementRules]</pre>

Parameter	Description
	For a detailed description of this table, see "Configuring Media Enhancement Profiles" on page 224 .

40.7 Control Network Parameters

40.7.1 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

Table 40-29: 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: <pre>[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_EnableSBCCClientForking, IPGroup_SourceUriInput, IPGroup_DestUriInput, IPGroup_ContactName, IPGroup_Username, IPGroup_Password, IPGroup_UUIFormat, IPGroup_QOEProfile, IPGroup_BWProfile, IPGroup_MediaEnhancementProfile, IPGroup_AlwaysUseSourceAddr; [/IPGroup]</pre> For a description of this table, see "Configuring IP Groups" on page 240. 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: <pre>[Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroup, Account_ServingIPGroup, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType; [/Account]</pre> For a detailed description of this table, see "Configuring Registration Accounts" on page 259.

Parameter	Description
Proxy Registration Parameters	
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.

Parameter	Description
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. 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.

Parameter	Description
	Notes: - This functionality can be configured per Proxy Set in the Proxy Set table (see "Configuring Proxy Sets" on page 251). - When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Web/EMS: Use Gateway Name for OPTIONS CLI: use-gw-name-for-opt [UseGatewayNameForOptions]	Determines whether the device uses its IP address or string name ("gateway name") in keep-alive SIP OPTIONS messages (host part of the Request-URI). The "gateway name" is configured using the SIPGatewayName parameter. The device uses the OPTIONS request as a keep-alive message with its primary and redundant SIP proxy servers (i.e., the EnableProxyKeepAlive parameter is set to 1). [0] No = (Default) Device's IP address is used in keep-alive OPTIONS messages. [1] Yes = Device's "gateway name" is used in keep-alive OPTIONS messages. [2] Server = Device's IP address is used in the From and To headers in keep-alive OPTIONS messages.
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	
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 251. To configure the Proxy Set attributes (such as Proxy Load Balancing) in the ini file, use the ProxySet parameter.

Parameter	Description
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: <pre>[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_SuccessDetectionRetries, ProxySet_SuccessDetectionInterval, ProxySet_FailureDetectionRetransmissions; [\ProxySet]</pre> For a description of this table, see "Configuring Proxy Sets" on page 251. 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.

Parameter	Description
Web: Registration Time Threshold CLI: registration-time-thres [RegistrationTimeThreshold]	Defines a threshold (in seconds) for re-registration timing. If this 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: 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 [EmptyAuthorizationHeader]	Enables the inclusion of the SIP Authorization header in initial 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: <pre>Authorization: Digest username=alice_private@home1.net, realm="home1.net", nonce="",</pre>

Parameter	Description
	<pre>response="e56131d19580cd833064787ecc"</pre> 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: <pre>Route: <sip:10.10.10.10;lr;transport=udp></pre> or <pre>Route: <sip: pcscf-gm.ims.rr.com;lr;transport=udp></pre>
[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.
[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

Parameter	Description
	between 160 to 200 seconds. This prevents an “avalanche” of keep-alive by multiple SIP UAs to a specific server.
Web: Max Generated Register Rate CLI: configure voip/sip-definition proxy-and-registration/max-gen-reg- rate [MaxGeneratedRegistersRate]	Defines the maximum number of user register requests that the device can send to a proxy (or registrar) server per second. This feature is useful in that it can be used to prevent an overload on the device's CPU caused by the sending of many (for example, 1,000) registration requests to a proxy server at any given time. The valid value is 30 to 300 register requests per second. The default is 100.

40.7.2 Network Application Parameters

The SIP network application parameters are described in the table below.

Table 40-30: 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: <pre>[SRD] FORMAT SRD_Index = SRD_Name, SRD_MediaRealm, SRD_IntraSRDMediaAnchoring, SRD_BlockUnRegUsers, SRD_MaxNumOfRegUsers, SRD_EnableUnAuthenticatedRegistrations; [\SRD]</pre> For a detailed description of this table, see "Configuring SRDs" on page 233.
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: <pre>[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_PreClassificationManSet; [\SIPInterface]</pre> For a detailed description of this table, see "Configuring SIP Interfaces" on page 236.
[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. 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.

Parameter	Description
[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: <pre>[NATTranslation] FORMAT NATTranslation_Index = NATTranslation_SourceIPInterfaceName, NATTranslation_TargetIPAddress, NATTranslation_SourceStartPort, NATTranslation_SourceEndPort, NATTranslation_TargetStartPort, NATTranslation_TargetEndPort; [\NATTranslation]</pre> For a detailed description of this table, see "Configuring NAT Translation per IP Interface" on page 130.
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: <pre>[CpMediaRealm] FORMAT CpMediaRealm_Index = CpMediaRealm_MediaRealmName, CpMediaRealm_IPv4IF, CpMediaRealm_IPv6IF, CpMediaRealm_PortRangeStart, CpMediaRealm_MediaSessionLeg, CpMediaRealm_PortRangeEnd, CpMediaRealm_IsDefault, CpMediaRealm_QoeProfile, CpMediaRealm_BWProfile; [\CpMediaRealm]</pre> For a detailed description of this table, see "Configuring Media Realms" on page 227.

Parameter	Description
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: <pre>[RemoteMediaSubnet] FORMAT RemoteMediaSubnet_Index = RemoteMediaSubnet_Realm, RemoteMediaSubnet_RemoteMediaSubnetIndex, RemoteMediaSubnet_RemoteMediaSubnetName, RemoteMediaSubnet_PrefixLength, RemoteMediaSubnet_AddressFamily, RemoteMediaSubnet_DstIPAddress, RemoteMediaSubnet_QOEProfileName, RemoteMediaSubnet_BWProfileName; [\RemoteMediaSubnet]</pre> For a detailed description of this table, see "Configuring Remote Media Subnets" on page 231.

40.8 General SIP Parameters

The general SIP parameters are described in the table below.

Table 40-31: General SIP Parameters

Parameter	Description
Web: Send reject on overload CLI: configure voip/sip-definition advanced-settings/reject-on-ovrld [SendRejectOnOverload]	Disables the sending of SIP 503 (Service Unavailable) responses upon receipt of new SIP dialog-initiating requests when the device's CPU is overloaded and thus, unable to accept and process new SIP messages. [0] Disable = No SIP 503 response is sent when CPU overloaded. [1] Enable (default) = SIP 503 response is sent when CPU overloaded. Note: Even if this parameter is disabled (i.e., 503 is not sent), the device still discards new SIP dialog-initiating requests when the CPU is overloaded.
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: 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 [VerifyReceeedRequestUri]	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

Parameter	Description
	(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 number of calls).
Web: Number of Calls Limit CLI: [IpProfile_CallLimit,]	Defines the maximum number of concurrent calls per IP Profile (see "Configuring IP Profiles" on page 275).
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: <pre> 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.6.80A.037.009 Reason: Q.850 ;cause=16 ;text="local" Content-Length: 0 </pre>
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

Parameter	Description
	(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 275. 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: 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 new session expiry timeout is calculated by subtracting the configured value from the original timeout as specified in the Session-Expires header. However, the new timeout must be greater than or equal to one-third (1/3) of the Session-Expires value. If the refresher does not send a refresh request within the new timeout, the device disconnects the session (i.e., sends a SIP BYE). For example, if you configure the parameter to 32 seconds and the Session-Expires value is 180 seconds, the session timeout occurs 148 seconds (i.e., 180 minus 32) after the last session refresh. If the Session-Expires header value is 90 seconds, the timeout occurs 60 seconds after the last refresh. This is because 90 minus 32 is 58 seconds, which is less than one third of the Session-Expires value (i.e., 60/3 is 30, and 90 minus 30 is 60). 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' value is used. A special value is 'empty', indicating that no user part in the Request-URI

Parameter	Description
	(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 275. 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 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:

Parameter	Description
	<pre>a=rtpmap:120 NoAudio/8000\r\n</pre> 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.
Web: Display Default SIP Port CLI: display-default-sip-port [DisplayDefaultSIPPort]	Enables the device to add the default SIP port 5060 (UDP/TCP) or 5061 (TLS) to outgoing messages that are received without a port. This condition also applies to manipulated messages where the resulting message has no port number. The device adds the default port number to the following SIP headers: Request-Uri, To, From, P-Asserted-Identity, P-Preferred-Identity, and P-Called-Party-ID. If the message is received with a port number other than the default, for example, 5070, the port number is not changed. An example of a SIP From header with the default port is shown below: <pre>From: <sip:+4000@10.8.4.105:5060;user=phone>;tag=f25419a96a;epid=009FAB8F3E</pre> ▪ [0] Disable (default) ▪ [1] Enable
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.

Parameter	Description
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. [1] Yes = (Default) 'user=phone' string is part of the SIP URI and SIP To header.
Web: Use user=phone in From Header CLI: phone-in-from-hdr [IsUserPhoneInFrom]	Determines whether the 'user=phone' string is added to the From and Contact SIP headers. [0] No = (Default) Doesn't add 'user=phone' string. [1] Yes = 'user=phone' string is part of the From and Contact headers.

Parameter	Description
Web: Use Tel URI for Asserted Identity CLI: uri-for-assert-id [UseTelURIForAssertedID]	Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers. ▪ [0] Disable = (Default) 'sip:' ▪ [1] Enable = 'tel:'
Web: Tel to IP No Answer Timeout CLI: tel2ip-no-ans-timeout [IPAlertTimeout]	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.

Parameter	Description
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: <pre>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)</pre> 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.
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: <pre>User-Agent: myproduct/v.6.80A.037.009</pre> If not configured, the default string, <AudioCodes product-name>/software version' is used, for example:

Parameter	Description
	<pre>User-Agent: Audiocodes-Sip-Gateway-Mediant 9000/v.6.80.010.006</pre> 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: <pre>o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126</pre>
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. ▪ [0] Forward = (Default) Use different call identifiers for a redirected

Parameter	Description
	INVITE message. ▪ [1] Redirect = Use the same call identifiers.
Web: Retry-After Time CLI: retry-afr-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 [EnablePAssociatedURI Header]	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.
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.
[DisableCryptoLifetimeSDP]	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:hhQe10yZRcRcplFPkH5xYY9R1de37ogh9G1MpvNp 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
[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.

Parameter	Description
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 395. ▪ [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.
[EnableSilenceSupplnSDP]	Determines the device's behavior upon receipt of SIP Re-INVITE messages that include the SDP's 'silencesupp:off' attribute. ▪ [0] = (Default) Disregard the 'silencesupp' 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' parameter.
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).

Parameter	Description
	▪ [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).
Network Node ID net-node-id [NetworkNodeId]	Defines the Network Node Identifier of the device for Avaya UCID. The valid value range is 1 to 0x7FFF. The default is 0. Notes: ▪ To use this feature, you must set the 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: 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 URI user part as INVITE sip:622125519100;ext=104@10.1.1.10 (or INVITE tel:622125519100;ext=104). If the parameter EnableMicrosoftExt 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., e622125519100104). 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
CLI: sdp-ecan-frmt	Defines the echo canceller format in the outgoing SDP. The 'ecan'

Parameter	Description
[SDPEcanFormat]	attribute is used in the SDP to indicate the use of echo cancellation. [0] = (Default) The 'ecan' attribute appears on the 'a=gpmnd' 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 [FirstCallIRBTId]	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 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 275.
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

Parameter	Description
	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 CLI: nb-of-rtx-b4-hot- swap [HotSwapRtx]	Defines the number of retransmitted INVITE/REGISTER messages before the call 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: <pre>[MessageManipulations] FORMAT MessageManipulations_Index = MessageManipulations_ManSetID, MessageManipulations_MessageType, MessageManipulations_Condition, MessageManipulations_ActionSubject, MessageManipulations_ActionType, MessageManipulations_ActionValue, MessageManipulations_RowRole; [MessageManipulations]</pre> For example, the below configuration changes the user part of the SIP From header to 200: <pre>MessageManipulations 1 = 0, Invite.Request, , Header.From.Url.User, 2, 200, 0;</pre> For a detailed description of this table, see Configuring SIP Message Manipulation on page 266.
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: <pre>[MessagePolicy]</pre>

Parameter	Description
	FORMAT MessagePolicy_Index = MessagePolicy_Policy, MessagePolicy_MaxMessageLength, MessagePolicy_MaxHeaderLength, MessagePolicy_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.

40.9 Coders and Profile Parameters

The profile parameters are described in the table below.

Table 40-32: 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, 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,

Parameter	Description
	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 275.

40.10 Channel Parameters

This subsection describes the device's channel parameters.

40.10.1 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

Table 40-33: RTP/RTCP and T.38 Parameters

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). For a detailed description of configuring the UDP port range, see Configuring RTP Base UDP Port on page 153. Note: For this parameter to take effect, a device reset is required.
[UdpPortSpacing]	Defines the UDP port spacing within the configured port range. [5] (default) [10] Notes: - A device reset is required for this parameter to take effect. - For more information on configuring the UDP port range, see Configuring RTP Base UDP Port on page 153.
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 RFC 3611. [0] Disable (default) [1] 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] Enable Calculation Only = 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	Defines the voice quality monitoring - minimum gap size (number of frames).

Parameter	Description
[VQMonGMin]	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).
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
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

40.11 SBC Parameters

The SBC parameters are described in the table below.

Table 40-34: 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.
Web: 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 = (Default) Call is rejected if classification fails. [1] Allow = 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: configure voip/sbc general-setting/sbc-dialog-subsc-route-mode	Enables the device to route in-dialog, refresh SIP SUBSCRIBE requests to the "working" (has connectivity) proxy. [0] = (Default) Disable – the device sends in-dialog, refresh

Parameter	Description
[SBCInDialogSubscribeRouteMode]	SUBSCRIBES according to the address in the Contact header of the 200 OK response received from the proxy to which the initial SUBSCRIBE was sent (as per the SIP standard). ▪ [1] = Enable – the device routes in-dialog, refresh SUBSCRIBES to the "working" proxy (regardless of the Contact header). The "working" proxy (address) is determined by the device's keep-alive mechanism for the Proxy Set that was used to route the initial SUBSCRIBE. Note: For this feature to be functional, ensure the following: ▪ Keep-alive mechanism is enabled for the Proxy Set ('Enable Proxy Keep Alive' parameter is set to any value other than Disable). ▪ Load-balancing between proxies is disabled ('Proxy Load Balancing Method' parameter is set to Disable).
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 not limit Session-Expires.

Parameter	Description
Web: SBC Session Refreshing Policy CLI: configure voip/sbc general-setting/sbc-session-refresh-policy [SBCSessionRefreshingPolicy]	Defines the SIP user agent responsible for periodically sending refresh requests for established sessions (active calls). The session refresh allows SIP UAs or proxies to determine the status of the SIP session. When a session expires, the session is considered terminated by the UAs, regardless of whether a SIP BYE was sent by one of the UAs. The SIP Session-Expires header conveys the lifetime of the session, which is sent in re-INVITE or UPDATE requests (session refresh requests). The 'refresher=' parameter in the Session-Expires header (sent in the initial INVITE or subsequent 2xx response) indicates who sends the session refresh requests. If the parameter contains the value 'uac', the device performs the refreshes; if the parameter contains the value 'uas', the remote proxy performs the refreshes. An example of the Session-Expires header is shown below: <pre>Session-Expires: 4000;refresher=uac</pre> Thus, this parameter is useful when a UA does not support session refresh requests or does not support the indication of who performs session refresh requests. In such a scenario, the device can be configured to perform the session refresh requests. [0] Remote Refresher = (Default) The UA (proxy) performs the session refresh requests. The device indicates this to the UA by sending the SIP message with the 'refresher=' parameter in the Session-Expires header set to 'uas'. [1] SBC Refresher = The device performs the session refresh requests. The device indicates this to the UA by sending the SIP message with the 'refresher=' parameter in the Session-Expires header set to 'uac'. Note: The time values of the Session-Expires (session refresh interval) and Min-SE (minimum session refresh interval) headers can be configured using the SBCSessionExpires and SBCMinSE parameters, respectively.
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 275. 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: 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 275. 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 275. 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
[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 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: <pre>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 udpt1 t38</pre> If this parameter is disabled, the only 'm=' line included in the SDP is the newly negotiated media (i.e., image).
Web: SBC Diversion URI Type CLI: sbc-diversion-uri-type (configure voip > sbc general-setting) [SBCDiversionUriType]	Defines the URI type to use in the SIP Diversion header of the outgoing SIP message. [0] Transparent = (Default) The device does not change the URI and leaves it as is. [1] Sip = The "sip" URI is used. [2] Tel = The "tel" URI is used. Note: The parameter is applicable only if the Diversion header is used. The SBCDiversionMode and SBCHistoryInfoMode parameters in the IP Profile table determine the call redirection (diversion) SIP header to use - History-Info or Diversion.
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	Defines the authentication and integrity level of quality of protection

Parameter	Description
of Protection CLI: auth-qop [AuthQOP]	(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 CLI: sbc-usr-rgstr-time [SBCUserRegistrationTime]	Global parameter that defines the duration (in seconds) of the periodic registrations that occur between the user and the device (the device responds 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 275. 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
CLI: config-voip>sbcc general-setting sbc-rand-expire [SBCRandomizeExpires]	Defines a value (in seconds) that is used to calculate a new value for the expiry time in the Expires header of SIP 200 OK responses for user registration and subscription requests from users. The expiry time value appears in the Expires header in REGISTER and SUBSCRIBE SIP messages. When the device receives such a request from a user, it forwards it to the proxy or registrar server. Upon a successful registration or subscription, the server sends a SIP 200 OK response. If the expiry time was unchanged by the server, the device applies this feature and changes the expiry time in the SIP 200 OK response before forwarding it to the user; otherwise, the device does not change the expiry time. This feature is useful in scenarios where multiple users may refresh their registration or subscription simultaneously, thereby causing the device to handle many such sessions at a given time. This may result in an overload of the device (reaching maximum session capacity), thereby preventing the establishment of new calls or preventing the handling of some user registration or subscription requests. When this feature is enabled, the device assigns a random expiry time to each user registration or subscription and thus, ensuring future user registration and subscription requests are more distributed over time (i.e., do not all occur simultaneously). The device takes any random number between 0 and the value configured by this parameter, and then subtracts this random number from the original expiry time value. For example, assume that the original expiry time is 120 and this parameter is set to 10. If the device randomly chooses the number 5 (i.e., between 0 and 10), the resultant expiry time will be 115 (120 minus 5). The valid value is 0 to 20. The default is 10. If set to 0, the device does not change the expiry time. Notes: - The lowest expiry time that the device sends in the 200 OK, regardless of the resultant calculation, is 10 seconds. For example, if the original expiry time is 12 seconds and this parameter is set to 5, theoretically, the new expiry time can be less than 10 (e.g., 12 – 4 = 8). However, the expiry time will be set to 10. - The expiry time received from the user can be changed by the device before forwarding it to the proxy. This is configured by the SBCUserRegistrationTime parameter.
Web: SBC Proxy Registration Time CLI: sbc-prxy-rgstr-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-rgstr-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.

Parameter	Description
[SBCEnableSurvivabilityNotice]	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: <pre>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></pre>
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 324.
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.

Parameter	Description
Web: SBC GRUU Mode CLI: sbc-gruu-mode [SBCGruuMode]	Determines the Globally Routable User Agent (UA) URI (GRUU) support, 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). <pre>Public-GRUU: sip:userA@domain.com;gr=unique-id</pre>
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 315.

Parameter	Description
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 media). Note: For more information on No Media Anchoring, see "No Media Anchoring (Anti Tromboning)" on page 302.
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 275. 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 320.
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 sends it to the other side. ▪ [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.

Parameter	Description
[SBCRemoveSIPFromNonSecuredTransport] configure voip > sbc settings > sbc-remove-sips-non-sec-transp	Defines the SIP headers for which the device replaces “sips:” with “sip:” in the outgoing SIP-initiating dialog request (e.g., INVITE) when the destination transport type is unsecured (e.g., UDP). (The “sips:” URI scheme indicates secured transport, for example, TLS.) [0] = (Default) The device replaces “sips:” with “sip:” for the Request-URI and Contact headers only (and retains “sips:” for all other headers). [1] = The device replaces “sips:” with “sip:” for the Request-URI, Contact, From, To, P-Asserted, P-Preferred, and Route headers.
CLI: sbc-media-sync [EnableSBCMediaSync]	Enables synchronization of media between two SIP user agents when a call is established between them. Media synchronization means that the media is properly negotiated (SDP offer/answer) between the user agents. In some scenarios, the call is established despite the media not being synchronized. This may occur, for example, in call transfer (SIP REFER) where the media between the transfer target and transferee are not synchronized. The device performs media synchronization by sending a re-INVITE immediately after the call is established in order for the user agents to negotiate the media (SDP offer/answer). [0] Disable = (Default) Media synchronization is performed only if the RTP mode (e.g., a=sendrecv, a=sendrecv, a=sendonly, a=recvonly, and a=inactive) between the user agents are different and synchronization is required. [1] Enable = Media synchronization is performed if the media, including RTP mode or any other media such as coders, is different and has not been negotiated between the user agents. [2] Never = Media synchronization is never performed.
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: <pre>[SBCAdmissionControl] FORMAT SBCAdmissionControl_Index = SBCAdmissionControl_AdmissionControlName, SBCAdmissionControl_LimitType, SBCAdmissionControl_IPGroupID, SBCAdmissionControl_SRDID, SBCAdmissionControl_RequestType, SBCAdmissionControl_RequestDirection, SBCAdmissionControl_Limit, SBCAdmissionControl_LimitPerUser, SBCAdmissionControl_Rate, SBCAdmissionControl_MaxBurst, SBCAdmissionControl_Reservation; [\\SBCAdmissionControl]</pre> For a detailed description of this table, see "Configuring Admission Control" on page 326.
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: <pre>[AllowedCodersGroupX] FORMAT AllowedCodersGroup_Index = AllowedCodersGroup_Name; [\\AllowedCodersGroup]</pre>

Parameter	Description
	Where X represents the index number. For a detailed description of this table, see "Configuring Allowed Audio Coder Groups" on page 329 .
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 331 .
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 332 .
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 339 .
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,

Parameter	Description
	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 340.
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 349.
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 353.

Parameter	Description
IP to IP Outbound Manipulation Table	
Web: IP to IP Outbound Manipulation CLI: configure voip > sbc manipulations ip-outbound-manipulation [IPOutboundManipulation]	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 as follows: <pre> 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; </pre> For a detailed description of this table, see "Configuring IP-to-IP Outbound Manipulations" on page 356.

40.12 Services

40.12.1 SIP-based Media Recording Parameters

The SIP-based media recording parameters are described in the table below.

Table 40-35: 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: <pre>[SIPRecRouting] FORMAT SIPRecRouting_Index = SIPRecRouting_RecordedIPGroupID, SIPRecRouting_RecordedSourcePrefix, SIPRecRouting_RecordedDestinationPrefix, SIPRecRouting_PeerIPGroupID, SIPRecRouting_PeerTrunkGroupID, SIPRecRouting_Caller, SIPRecRouting_SRSIPGroupID; [\SIPRecRouting]</pre> For a detailed description of this table, see "Configuring SIP Recording Routing Rules" on page 171.

40.12.2 RADIUS and LDAP Parameters

40.12.2.1 General Parameters

The general RADIUS and LDAP parameters are described in the table below.

General RADIUS and LDAP Parameters

Parameter	Description
Web: Use Local Users Database CLI: configure system > mgmt-auth > use-local-users-db [MgmtUseLocalUsersDatabase]	Defines when the device uses its local user database (Web Users table) for LDAP- or RADIUS-based management-user login authentication. [0] When No Auth Server Defined = (Default) When no LDAP/RADIUS server is configured (or as fallback if the server is inaccessible). [1] Always = Always first verify the user's credentials in the Web Users table, and if not found, then search the LDAP/RADIUS server.
Web: Behavior upon Authentication Server Timeout CLI: configure system > mgmt-auth > timeout-behavior [MgmtBehaviorOnTimeout]	Defines the device's response when a connection timeout occurs with the LDAP/RADIUS server. [0] Deny Access = User is denied access to the management platform. [1] Verify Access Locally = (Default) Device verifies the user's credentials in its Web Users table (local database). Note: The parameter is applicable to LDAP- or RADIUS-based management-user login authentication.
Web: Default Access Level CLI: default-access-level [DefaultAccessLevel]	Defines the default access level for the device when the LDAP/RADIUS response doesn't include an access level attribute for determining the user's management access level. The valid range is 0 to 255. The default is 200 (i.e., Security Administrator). Note: The parameter is applicable to LDAP- or RADIUS-based management-user login authentication and authorization.

40.12.2.2 RADIUS Parameters

The RADIUS parameters are described in the table below.

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	Defines the IP address of the RADIUS accounting server.

Parameter	Description
[RADIUSAccServerIP]	
Web: Accounting Port CLI: accounting-port [RADIUSAccPort]	Defines the port of the RADIUS accounting server. The default is 1646.
Web/EMS: 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. [2] At Setup & Release = Sent at call setup and release.
Web: AAA Indications EMS: 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 EMS: Web Use Radius 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.
Web: RADIUS Authentication Server IP Address EMS: RADIUS Auth Server IP 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 EMS: RADIUS Auth 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 EMS: RADIUS Auth Server Secret CLI: shared-secret	Defines the 'secret' used to authenticate the device to the RADIUS server. This should be a cryptically strong password.

Parameter	Description
[SharedSecret]	
RADIUS Authentication Parameters	
Web: Password Local Cache Mode CLI: local-cache-mode [RadiusLocalCacheMode]	Defines 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, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Web: Password Local 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 0xFFFFFFFF. 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.
EMS: RADIUS Auth Number of Retries [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.

40.12.2.3 LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below.

LDAP Parameters

Parameter	Description
Web: LDAP Service CLI: configure voip/lldap/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 servers. [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.
Web: LDAP Authentication Filter CLI: configure voip > ldap > auth-filter [LDAPAuthFilter]	Defines the LDAP search filter attribute for searching the login username in the directory's subtree for LDAP-based user authentication and authorization. You can use the dollar (\$) sign to represent the username. For example, if this parameter is set to "(sAMAccountName=\$)" and the user logs in with the username "SueM", the LDAP query is run for sAMAccountName=SueM.
Web: Use LDAP for Web/Telnet Login CLI: configure voip > ldap > enable-mgmt-login [MgmtLDAPLogin]	Enables LDAP-based management-user login authentication and authorization. [0] Disable (default) [1] Enable Note: For this parameter to take effect, a device reset is required.
[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.
Web: MS LDAP OCS Number attribute name EMS: LDAP ocs Number Attribute Name CLI: ldap-ocs-nm-attr [MSLDAPOCSNumAttribute]	Defines the name of the attribute that represents the user's Lync 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 [MSLDAPPBXNumAttribute]	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".

Parameter	Description
eName]	
Web: MS LDAP MOBILE Number attribute name CLI: ldap-mobile-nm-attr [MSLDAPMobileNumAttribute]	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 [MSLDAPPrivateNumAttribute]	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 [MSLDAPDisplayNameAttribute]	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 189.
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-timeout [LDAPCacheEntryRemovalTimeout]	Defines the duration (in hours) after which the LDAP entry is removed from the cache. The default is 0.
LDAP Configuration Table	
Web: LDAP Configuration Table CLI: configure voip > ldap > ldap-configuration	Defines the LDAP servers. [LdapConfiguration] FORMAT LdapConfiguration_Index = LdapConfiguration_LdapConfServerIp,

Parameter	Description
[LdapConfiguration]	LdapConfiguration_LdapConfServerPort, LdapConfiguration_LdapConfServerMaxRespondTime, LdapConfiguration_LdapConfServerDomainName, LdapConfiguration_LdapConfPassword, LdapConfiguration_LdapConfBindDn, LdapConfiguration_LdapConfInterfaceType, LdapConfiguration_Type, LdapConfiguration_MngmAuthAtt, LdapConfiguration_ConnectionStatus; [\LdapConfiguration] For a detailed description of this table, see 'Configuring LDAP Servers' on page 181 .
LDAP Server Search DN Table	
Web: LDAP Search DN 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 DN (Base Paths) per LDAP Server' on page 184 .
Management LDAP Groups Table	
Web: Management LDAP Groups Table CLI: configure voip > ldap > mgmt-ldap-groups [MgmntLDAPGroups]	Defines the users group attribute in the AD and corresponding management access level. [MgmntLDAPGroups] FORMAT MgmntLDAPGroups_Index = MgmntLDAPGroups_LdapConfigurationIndex, MgmntLDAPGroups_GroupIndex, MgmntLDAPGroups_Level, MgmntLDAPGroups_Group; [\MgmntLDAPGroups] For a detailed description of this table, see 'Configuring Access Level per Management Groups Attributes' on page 186 .

40.12.3 Least Cost Routing Parameters

The Least Cost Routing (LCR) parameters are described in the table below.

Table 40-36: 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 204.
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 205.
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 206.

40.12.4 Call Setup Rules Parameters

The Call Setup Rules parameters are described in the table below.

Table 40-37: 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 209.

41 SBC Capacity

The table below lists the maximum capacity. This includes SIP signaling sessions, SBC sessions, and registered users.

Table 41-1: Maximum Call Sessions and Registered Users

Signaling Sessions	Media Sessions			Registered Users
	RTP-to-RTP	SRTP-RTP	Codec Transcoding	
6,000	4,000	4,000	-	36,000
16,000	16,000	12,000	-	0

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.



This page is intentionally left blank.

42 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 <https://www.audiocodes.com/library/firmware/>.

Table 42-1: Technical Specifications

Function	Specification
Networking Interfaces	
Ethernet	12 redundant 1 GB Ethernet ports
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
Security	
Access Control	DoS/DDoS line rate protection, bandwidth throttling, Dynamic Blacklisting
VoIP Firewall	RTP pinhole management, Rogue RTP detection and prevention, SIP message policy
Encryption and Authentication	TLS, SRTP, HTTPS, SSH, Client/Server SIP Digest authentication, RADIUS Digest
Privacy	Topology Hiding, User Privacy
Traffic Separation	VLAN/physical interface separation for multiple Media, Control and OAMP interfaces
Intrusion Detection System	Detect and mitigate VoIP attacks, prevent Theft of Service and unauthorized access
Interoperability	
SIP B2BUA	Full SIP transparency, mature & broadly deployed SIP stack
SIP interworking	3xx redirect, REFER, PRACK, Session Timer, Early media, Call hold, Delayed offer
Registration	Registration and authentication on behalf of an IP-PBX
Transport Mediation	SIP over UDP to SIP over TCP or SIP over TLS, IPv4 to IPv6, RTP to SRTP
Header Manipulation	Ability to add/modify/delete headers using advanced regular expressions
URI and Number Manipulations	URI User and Host name manipulations. Ingress & Egress Digit Manipulation
NAT	Local and Far End NAT traversal for support of remote workers
Coder normalization	Coder enforcement and re-prioritization

Function	Specification
Voice Quality and SLA	
Call Admission Control	Based on bandwidth, session establishment rate, number of connections/registrations
Packet Marking	802.1p/Q VLAN tagging, DiffServ, TOS
Standalone Survivability	Maintain local calls in the event of WAN failure
Transparent Media	Low latency, unprocessed payload transfer
Media De-anchoring	Hair-pinning of local calls to avoid unnecessary media delays and bandwidth consumption
Redundancy	High availability with two box redundancy, active calls preserved
Voice Quality Monitoring	AudioCodes Session Experience Manager (SEM)
Quality of Experience	Access control and media quality enhancements based on QoE and bandwidth utilization
Test agent	Remote verification of connectivity, voice quality and SIP message flow between SIP UAs
SIP Routing	
Routing Methods	Request URL, IP Address, FQDN, ENUM, advanced LDAP
Advanced Routing Criteria	QoE, bandwidth, SIP message (SIP request, Coder type etc)
Routing Features	Least cost routing, call forking, load balancing
Redundancy	Detect proxy failures and route to alternative proxies
SIPRec	IETF standard SIP recording interface
Physical / Environmental	
Dimensions	43.45 x 62.23 x 2.97 cm (17.11 x 27.5 x 1.7 in)
Weight	19.2 kg (42.3 lb)
Mounting	19"-rack mount
Power	100-240 VAC power feeds
Environmental	Operational: 10 to 35°C
Regulatory Compliance	
FCC rating	Class A
Normative standards	CISPR 22; EN 55022; EN 55024; FCC CFR 47, Pt 15; ICES-003; CNS13438; GB9254; K22; K24; EN 61000-3-2; EN 61000-3-3; EN 60950-1; IEC 60950-1

This page is intentionally left blank.

International Headquarters

1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.

200 Cottontail Lane,
Suite A101E,
Somerset, NJ 08873
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: <https://www.audiocodes.com/corporate/offices-worldwide>

Website: <https://www.audiocodes.com/>

©2018 AudioCodes Ltd. All rights reserved. AudioCodes, AC, HD VoIP, HD VoIP Sounds Better, IPmedia, Mediant, MediaPack, What's Inside Matters, OSN, SmartTAP, User Management Pack, VMAS, VoIPerfect, VoIPerfectHD, Your Gateway To VoIP, 3GX, VocaNom, AudioCodes One Voice and CloudBond 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.

Document #: LTRT-41996

