

Mediant 2600

Enterprise Session Border Controller (E-SBC)

Version 7.0



Table of Contents

1	Introduction	23
1.1	Product Overview	23
1.2	Typographical Conventions	24
1.3	Getting Familiar with Configuration Concepts and Terminology.....	25
1.3.1	SBC Application.....	25
Getting Started with Initial Connectivity.....		29
2	Introduction	31
3	Default OAMP IP Address.....	33
4	Configuring VoIP LAN Interface for OAMP	35
4.1	Web Interface	35
4.2	CLI	37
Management Tools		39
5	Introduction	41
6	Web-Based Management.....	43
6.1	Getting Acquainted with the Web Interface.....	43
6.1.1	Computer Requirements.....	43
6.1.2	Accessing the Web Interface	44
6.1.3	Areas of the GUI	45
6.1.4	Toolbar Description.....	46
6.1.5	Navigation Tree	47
6.1.5.1	Displaying Navigation Tree in Basic and Full View	47
6.1.5.2	Showing / Hiding the Navigation Pane.....	48
6.1.6	Working with Configuration Pages	49
6.1.6.1	Accessing Pages.....	49
6.1.6.2	Viewing Parameters	49
6.1.6.3	Modifying and Saving Parameters	51
6.1.7	Working with Tables	52
6.1.7.1	Table Toolbar Description	53
6.1.7.2	Toggling Display Mode of Table Dialog Boxes	54
6.1.7.3	Scrolling through Table Pages	54
6.1.7.4	Searching Table Entries.....	55
6.1.7.5	Sorting Tables by Column.....	55
6.1.8	Searching for Configuration Parameters	56
6.1.9	Creating a Login Welcome Message.....	58
6.1.10	Getting Help.....	59
6.1.11	Logging Off the Web Interface.....	60
6.2	Customizing the Web Interface.....	60
6.2.1	Replacing the Corporate Logo.....	60
6.2.1.1	Replacing the Corporate Logo with an Image.....	61
6.2.1.2	Replacing the Corporate Logo with Text.....	61
6.2.2	Customizing the Product Name.....	62
6.2.3	Customizing the Favicon	62
6.2.4	Creating a Login Welcome Message.....	64

6.3	Viewing the Home Page	65
6.4	Configuring Web User Accounts.....	67
6.4.1	Basic User Accounts Configuration	69
6.4.2	Advanced User Accounts Configuration.....	71
6.5	Displaying Login Information upon Login	75
6.6	Configuring Web Security Settings	76
6.6.1	Configuring Secured (HTTPS) Web	76
6.6.2	Configuring Web Session and Access Settings	76
6.7	Web Login Authentication using Smart Cards.....	78
6.8	Configuring Web and Telnet Access List	79
7	CLI-Based Management.....	81
7.1	Getting Familiar with CLI	81
7.1.1	Understanding Configuration Modes	81
7.1.2	Using CLI Shortcuts.....	82
7.1.3	Common CLI Commands	83
7.1.4	Configuring Tables through CLI.....	84
7.1.5	Understanding CLI Error Messages	85
7.2	Enabling CLI.....	86
7.2.1	Enabling Telnet for CLI	86
7.2.2	Enabling SSH with RSA Public Key for CLI.....	86
7.3	Configuring Maximum Telnet/SSH Sessions	89
7.4	Establishing a CLI Session	89
7.5	Viewing Current CLI Sessions	90
7.6	Terminating a User's CLI Session	90
7.7	Configuring Displayed Output Lines in CLI Terminal Window	91
8	SNMP-Based Management	93
8.1	Disabling SNMP	93
8.2	Configuring SNMP Community Strings	94
8.3	Configuring SNMP Trap Destinations	96
8.4	Configuring SNMP Trusted Managers	97
8.5	Configuring SNMP V3 Users	98
9	INI File-Based Management.....	101
9.1	INI File Format.....	101
9.1.1	Configuring Individual ini File Parameters	101
9.1.2	Configuring Table ini File Parameters	101
9.1.3	General ini File Formatting Rules	103
9.2	Configuring an ini File.....	103
9.3	Loading an ini File to the Device.....	104
9.4	Secured Encoded ini File.....	104
9.5	Configuring Password Display in ini File	105
9.6	INI Viewer and Editor Utility	106
General System Settings		107
10	Configuring SSL/TLS Certificates.....	109
10.1	Configuring TLS Certificate Contexts.....	109

10.2	Assigning CSR-based Certificates to TLS Contexts.....	113
10.3	Assigning Externally Created Private Keys to TLS Contexts.....	116
10.4	Generating Private Keys for TLS Contexts	117
10.5	Creating Self-Signed Certificates for TLS Contexts	118
10.6	Importing Certificates and Certificate Chain into Trusted Certificate Store.....	119
10.7	Configuring Mutual TLS Authentication.....	121
10.7.1	TLS for SIP Clients	121
10.7.2	TLS for Remote Device Management	121
10.8	Configuring TLS Server Certificate Expiry Check	123
11	Date and Time.....	125
11.1	Configuring Automatic Date and Time using SNTP	125
11.2	Configuring Date and Time Manually.....	126
11.3	Configuring the Time Zone	127
11.4	Configuring Daylight Saving Time.....	128
General VoIP Configuration.....		129
12	Network.....	131
12.1	Configuring Physical Ethernet Ports	131
12.2	Configuring Ethernet Port Groups.....	133
12.3	Configuring Underlying Ethernet Devices	135
12.4	Configuring IP Network Interfaces	138
12.4.1	Assigning NTP Services to Application Types	141
12.4.2	Multiple Interface Table Configuration Summary and Guidelines	142
12.4.3	Networking Configuration Examples	143
12.4.3.1	One VoIP Interface for All Applications	143
12.4.3.2	VoIP Interface per Application Type.....	143
12.4.3.3	VoIP Interfaces for Combined Application Types	144
12.4.3.4	VoIP Interfaces with Multiple Default Gateways	145
12.5	Configuring Static IP Routes.....	146
12.5.1	Configuration Example of Static IP Routes	147
12.5.2	Troubleshooting the Routing Table	148
12.6	Configuring Quality of Service	149
12.7	Configuring ICMP Messages.....	151
12.8	DNS	152
12.8.1	Configuring the Internal DNS Table.....	152
12.8.2	Configuring the Internal SRV Table.....	153
12.9	Network Address Translation Support	156
12.9.1	Device Located behind NAT	156
12.9.1.1	Configuring a Static NAT IP Address for All Interfaces.....	157
12.9.1.2	Configuring NAT Translation per IP Interface	158
12.9.2	Remote UA behind NAT	159
12.9.2.1	SIP Signaling Messages	159
12.9.2.2	Media (RTP/RTCP/T.38)	160
12.10	Robust Receipt of Media Streams by Media Latching.....	165
12.11	Multiple Routers Support	167

13 Security	169
13.1 Configuring Firewall Settings	169
13.2 Configuring General Security Settings	174
13.3 Intrusion Detection System	175
13.3.1 Enabling IDS	176
13.3.2 Configuring IDS Policies	176
13.3.3 Assigning IDS Policies	180
13.3.4 Viewing IDS Alarms	181
14 Media	185
14.1 Configuring Voice Settings	185
14.1.1 Configuring Voice Gain (Volume) Control	185
14.1.2 Configuring Echo Cancellation	186
14.2 Fax and Modem Capabilities	187
14.2.1 Fax/Modem Operating Modes	188
14.2.2 Fax/Modem Transport Modes	188
14.2.2.1 T.38 Fax Relay Mode	189
14.2.2.2 G.711 Fax / Modem Transport Mode	192
14.2.2.3 Fax Fallback	192
14.2.2.4 Fax/Modem Bypass Mode	193
14.2.2.5 Fax / Modem NSE Mode	194
14.2.2.6 Fax / Modem Transparent with Events Mode	195
14.2.2.7 Fax / Modem Transparent Mode	195
14.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection	196
14.2.3 V.34 Fax Support	197
14.2.3.1 Bypass Mechanism for V.34 Fax Transmission	197
14.2.3.2 Relay Mode for T.30 and V.34 Faxes	197
14.2.4 V.152 Support	198
14.3 Configuring RTP/RTCP Settings	199
14.3.1 Configuring the Dynamic Jitter Buffer	199
14.3.2 Configuring RFC 2833 Payload	200
14.3.3 Configuring RTP Base UDP Port	201
14.4 Event Detection and Notification using X-Detect Header	202
14.4.1 Detecting Answering Machine Beeps	203
14.4.2 SIP Call Flow Examples of Event Detection and Notification	203
14.5 Answering Machine Detection (AMD)	206
14.5.1 Configuring AMD	208
14.6 Automatic Gain Control (AGC)	210
14.7 Configuring Various Codec Attributes	211
14.8 Configuring Media (SRTP) Security	211
14.8.1 SRTP using DTLS Protocol	213
15 Services	215
15.1 DHCP Server Functionality	215
15.1.1 Configuring the DHCP Server	215
15.1.2 Configuring the Vendor Class Identifier	220
15.1.3 Configuring Additional DHCP Options	221
15.1.4 Configuring Static IP Addresses for DHCP Clients	223
15.1.5 Viewing and Deleting DHCP Clients	224
15.2 SIP-based Media Recording	226
15.2.1 Enabling SIP-based Media Recording	229
15.2.2 Configuring SIP Recording Rules	230
15.2.3 Configuring SIP User Part for SRS	231

15.2.4	Interworking SIP-based Media Recording with Third-Party Vendors	232
15.2.4.1	Genesys	232
15.2.4.2	Avaya UCID	232
15.3	RADIUS-based Services	233
15.3.1	Enabling RADIUS Services	233
15.3.2	Configuring RADIUS Servers	233
15.3.3	Configuring Interface for RADIUS Communication	236
15.3.4	Configuring General RADIUS Parameters	236
15.3.5	RADIUS-based Management User Authentication	236
15.3.5.1	Setting Up a Third-Party RADIUS Server	237
15.3.5.2	Configuring RADIUS-based User Authentication	238
15.3.5.3	Securing RADIUS Communication	240
15.3.5.4	RADIUS-based User Authentication in URL	240
15.3.6	RADIUS-based CDR Accounting	240
15.4	LDAP-based Management and SIP Services	241
15.4.1	Enabling the LDAP Service	242
15.4.2	Enabling LDAP-based Web/CLI User Login Authentication and Authorization	243
15.4.3	Configuring LDAP Server Groups	243
15.4.4	Configuring LDAP Servers	245
15.4.5	Configuring LDAP DN (Base Paths) per LDAP Server	250
15.4.6	Configuring the LDAP Search Filter Attribute	251
15.4.7	Configuring Access Level per Management Groups Attributes	251
15.4.8	Configuring the Device's LDAP Cache	253
15.4.8.1	Refreshing the LDAP Cache	255
15.4.8.2	Clearing the LDAP Cache	257
15.4.9	Configuring Local Database for Management User Authentication	257
15.4.10	LDAP-based Login Authentication Example	258
15.4.11	Enabling LDAP Searches for Numbers with Characters	263
15.4.12	Active Directory-based Routing for Microsoft Lync	264
15.4.12.1	Querying the AD and Routing Priority	264
15.4.12.2	Configuring AD-Based Routing Rules	267
15.5	Least Cost Routing	269
15.5.1	Overview	269
15.5.2	Configuring LCR	271
15.5.2.1	Configuring Cost Groups	271
15.5.2.2	Configuring Time Bands for Cost Groups	272
15.5.2.3	Assigning Cost Groups to Routing Rules	274
15.6	HTTP-based Remote Services	275
15.6.1	Configuring HTTP Services	275
15.6.2	Configuring Remote HTTP Hosts	278
15.6.3	Centralized Third-Party Routing Server or ARM	280
15.7	HTTP-based Proxy Services	283
15.7.1	Enabling the HTTP Proxy Application	284
15.7.2	Configuring HTTP Interfaces	284
15.7.3	Configuring HTTP Proxy Services	286
15.7.4	Configuring HTTP Proxy Hosts	287
15.7.5	Configuring an HTTP-based EMS Service	289
15.8	Configuring Call Setup Rules	292
15.8.1	Call Setup Rule Examples	296
15.9	Enhanced 9-1-1 Support for Lync Server	298
15.9.1	About E9-1-1 Services	298
15.9.2	Microsoft Lync Server and E9-1-1	299
15.9.2.1	Gathering Location Information of Lync Clients for 911 Calls	300
15.9.2.2	Adding ELINs to the Location Information Server	301

15.9.2.3	Passing Location Information to the PSTN Emergency Provider	302
15.9.3	AudioCodes ELIN Device for Lync Server E9-1-1 Calls to PSTN.....	304
15.9.3.1	Detecting and Handling E9-1-1 Calls	304
15.9.3.2	Pre-empting Existing Calls for E9-1-1 Calls	306
15.9.3.3	PSAP Callback to Lync Clients for Dropped E9-1-1 Calls	306
15.9.3.4	Selecting ELIN for Multiple Calls within Same ERL	307
15.9.4	Configuring AudioCodes ELIN Device.....	308
15.9.4.1	Enabling the E9-1-1 Feature	308
15.9.4.2	Configuring the E9-1-1 Callback Timeout	308
15.9.4.3	Configuring the SIP Release Cause Code for Failed E9-1-1 Calls.....	308
15.9.4.4	Configuring SBC IP-to-IP Routing Rule for E9-1-1	309
15.9.4.5	Viewing the ELIN Table	309
16	Quality of Experience.....	311
16.1	Reporting Voice Quality of Experience to SEM.....	311
16.1.1	Configuring the SEM Server	311
16.1.2	Configuring Clock Synchronization between Device and SEM	312
16.1.3	Enabling RTCP XR Reporting to SEM	312
16.2	Configuring Quality of Experience Profiles.....	313
16.3	Configuring Bandwidth Profiles.....	317
16.4	Configuring Media Enhancement Profiles	320
17	Control Network	323
17.1	Configuring Media Realms	323
17.1.1	Configuring Remote Media Subnets.....	326
17.1.2	Configuring Media Realm Extensions	329
17.2	Configuring SRDs.....	332
17.2.1	Filtering Tables in Web Interface by SRD	337
17.2.2	Multiple SRDs for Multi-tenant Deployments.....	337
17.2.3	Cloning SRDs	340
17.2.4	Color-Coding of SRDs in Web Interface.....	341
17.2.5	Automatic Configuration based on SRD	341
17.3	Configuring SIP Interfaces.....	342
17.4	Configuring IP Groups	348
17.5	Configuring Proxy Sets.....	360
18	SIP Definitions	369
18.1	Configuring SIP Parameters	369
18.2	Configuring Registration Accounts.....	369
18.2.1	Regular Registration Mode	372
18.2.2	Single Registration for Multiple Phone Numbers using GIN.....	372
18.3	Configuring Proxy and Registration Parameters	373
18.3.1	SIP Message Authentication Example	375
18.4	Configuring SIP Message Manipulation	377
18.5	Configuring SIP Message Policy Rules.....	382
19	Coders and Profiles	385
19.1	Configuring Default Coders	385
19.2	Configuring Coders Groups.....	389
19.3	Configuring IP Profiles.....	391
Session Border Controller Application.....		421

20 SBC Overview.....	423
20.1 Feature List	423
20.2 B2BUA and Stateful Proxy Operating Modes.....	425
20.3 Call Processing of SIP Dialog Requests	428
20.4 User Registration.....	430
20.4.1 Initial Registration Request Processing.....	430
20.4.2 Classification and Routing of Registered Users	431
20.4.3 General Registration Request Processing	432
20.4.4 Registration Refreshes	432
20.4.5 Registration Restriction Control.....	433
20.4.6 Deleting Registered Users.....	433
20.5 Media Handling	434
20.5.1 Media Anchoring.....	435
20.5.2 Direct Media.....	436
20.5.3 Restricting Audio Coders	438
20.5.4 Coder Transcoding	438
20.5.5 Transcoding Mode	441
20.5.6 Prioritizing Coder List in SDP Offer	442
20.5.7 SRTP-RTP and SRTP-SRTP Transcoding	442
20.5.8 Multiple RTP Media Streams per Call Session	443
20.5.9 Interworking Miscellaneous Media Handling	443
20.5.9.1 Interworking DTMF Methods.....	443
20.5.9.2 Interworking RTP Redundancy	443
20.5.9.3 Interworking RTP-RTCP Multiplexing	444
20.5.9.4 Interworking RTCP Attribute in SDP	444
20.5.9.5 Interworking Crypto Lifetime Field.....	444
20.5.9.6 Interworking Media Security Protocols.....	444
20.5.9.7 Interworking ICE Lite for NAT Traversal	444
20.6 Fax Negotiation and Transcoding	445
20.7 Limiting SBC Call Duration	445
20.8 SBC Authentication	445
20.8.1 SIP Authentication Server Functionality	446
20.8.2 User Authentication based on RADIUS.....	446
20.9 Interworking SIP Signaling.....	447
20.9.1 Interworking SIP 3xx Redirect Responses	447
20.9.1.1 Resultant INVITE Traversing Device	447
20.9.1.2 Local Handling of SIP 3xx	449
20.9.2 Interworking SIP Diversion and History-Info Headers	449
20.9.3 Interworking SIP REFER Messages.....	450
20.9.4 Interworking SIP PRACK Messages	450
20.9.5 Interworking SIP Session Timer	451
20.9.6 Interworking SIP Early Media	451
20.9.7 Interworking SIP re-INVITE Messages.....	453
20.9.8 Interworking SIP UPDATE Messages	453
20.9.9 Interworking SIP re-INVITE to UPDATE.....	454
20.9.10 Interworking Delayed Offer	454
20.9.11 Interworking Call Hold.....	454
20.9.12 Interworking SIP Via Headers	455
20.9.13 Interworking SIP User-Agent Headers	455
20.9.14 Interworking SIP Record-Route Headers	455
20.9.15 Interworking SIP To-Header Tags in Multiple SDP Answers	455
20.9.16 Interworking In-dialog SIP Contact and Record-Route Headers.....	455

21	Enabling the SBC Application.....	457
22	Configuring General SBC Settings.....	459
22.1	Interworking Dialog Information in SIP NOTIFY Messages.....	460
23	Configuring Admission Control.....	463
24	Configuring Coder Groups.....	467
24.1	Configuring Allowed Audio Coder Groups	467
24.2	Configuring Allowed Video Coder Groups	469
25	Routing SBC	471
25.1	Configuring Classification Rules	471
25.1.1	Classification Based on URI of Selected Header Example	477
25.2	Configuring Message Condition Rules.....	478
25.3	Configuring SBC IP-to-IP Routing.....	480
25.4	Configuring SIP Response Codes for Alternative Routing Reasons	491
25.5	Configuring SBC Routing Policy Rules	493
26	SBC Manipulations.....	497
26.1	Configuring IP-to-IP Inbound Manipulations	499
26.2	Configuring IP-to-IP Outbound Manipulations.....	504
27	Configuring Dial Plans.....	511
27.1	Importing and Exporting Dial Plans.....	515
27.2	Creating Dial Plan Files	516
27.3	Using Dial Plan Tags for IP-to-IP Routing	516
27.3.1	Dial Plan Backward Compatibility	518
27.4	Using Dial Plan Tags for Outbound Manipulation	520
28	Advanced SBC Features.....	521
28.1	Configuring Call Preemption for SBC Emergency Calls.....	521
28.2	Emergency Call Routing using LDAP to Obtain ELIN	522
28.3	WebRTC	523
28.3.1	SIP over WebSocket.....	526
28.3.2	Configuring WebRTC.....	527
28.4	Configuring Dual Registration.....	528
28.5	Call Forking	531
28.5.1	Initiating SIP Call Forking	531
28.5.2	SIP Forking Initiated by SIP Proxy Server	532
28.5.3	Call Forking-based IP-to-IP Routing Rules.....	533
28.6	Call Survivability	534
28.6.1	Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server for Survivability.....	534
28.6.2	BroadSoft's Shared Phone Line Call Appearance for SBC Survivability.....	535
28.6.3	Call Survivability for Call Centers	536
28.6.4	Survivability Mode Display on Aastra IP Phones	539
28.7	Alternative Routing on Detection of Failed SIP Response	539
Cloud Resilience Package		541

29 CRP Overview	543
30 CRP Configuration	545
30.1 Enabling the CRP Application	545
30.2 Configuring Call Survivability Mode	545
30.3 Pre-Configured IP Groups	547
30.4 Pre-Configured IP-to-IP Routing Rules	547
30.4.1 Normal Mode	547
30.4.2 Emergency Mode	548
30.4.3 Auto Answer to Registrations	550
30.5 Configuring PSTN Fallback	551
High Availability System	553
31 HA Overview	555
31.1 Connectivity and Synchronization between Devices	555
31.2 Device Switchover upon Failure	556
31.3 HA Status on the Home Page	557
32 HA Configuration	559
32.1 Initial HA Configuration	559
32.1.1 Network Topology Types and Rx/Tx Ethernet Port Group Settings	559
32.1.2 Configuring the HA Devices	561
32.1.2.1 Step 1: Configure the First Device	561
32.1.2.2 Step 2: Configure the Second Device	563
32.1.2.3 Step 3: Initialize HA on the Devices	565
32.2 Configuration while HA is Operational	566
32.3 Configuring Firewall Allowed Rules	566
32.4 Monitoring IP Entity and HA Switchover upon Ping Failure	567
33 HA Maintenance	571
33.1 Maintenance of Redundant Device	571
33.2 Replacing a Failed Device	571
33.3 Forcing a Switchover	572
33.4 Software Upgrade	572
Maintenance	573
34 Basic Maintenance	575
34.1 Resetting the Device	575
34.2 Remotely Resetting Device using SIP NOTIFY	577
34.3 Locking and Unlocking the Device	577
34.4 Saving Configuration	578
35 High Availability Maintenance	579
35.1 Initiating an HA Switchover	579
35.2 Resetting the Redundant Unit	579

36 Disconnecting Active Calls	581
37 Software Upgrade.....	583
37.1 Auxiliary Files	583
37.1.1 Loading Auxiliary Files.....	583
37.1.1.1 Loading Auxiliary Files through Web Interface	584
37.1.1.2 Loading Auxiliary Files through CLI	585
37.1.1.3 Loading Auxiliary Files through ini File using TFTP	585
37.1.2 Deleting Auxiliary Files	585
37.1.3 Call Progress Tones File	586
37.1.4 Prerecorded Tones File	588
37.1.5 Dial Plan File.....	589
37.1.5.1 Creating a Dial Plan File.....	589
37.1.5.2 Obtaining IP Destination from Dial Plan File	590
37.1.5.3 Viewing Information of Installed Dial Plan File	590
37.1.6 User Information File	591
37.1.6.1 Enabling the User Info Table.....	591
37.1.6.2 User Information File for SBC User Database	591
37.1.6.3 Viewing the Installed User Info File Name	595
37.1.7 AMD Sensitivity File.....	595
37.2 Software License Key.....	596
37.2.1 Viewing the License Key.....	596
37.2.2 Installing a New Software License Key	596
37.2.2.1 Installing Software License Key through Web Interface	596
37.2.2.2 Installing Software License Key through CLI	598
37.2.3 Viewing the Device's Product Key	598
37.3 Upgrading SBC Capacity Licenses by License Pool Manager Server.....	598
37.4 Software Upgrade Wizard.....	601
38 Backing Up and Loading Configuration File.....	607
39 Automatic Provisioning	609
39.1 Automatic Configuration Methods.....	609
39.1.1 DHCP-based Provisioning	609
39.1.2 HTTP-based Provisioning.....	610
39.1.3 FTP-based Provisioning	611
39.1.4 Provisioning using AudioCodes EMS	611
39.2 HTTP/S-Based Provisioning using the Automatic Update Feature.....	612
39.2.1 Files Provisioned by Automatic Update.....	612
39.2.2 File Location for Automatic Update	612
39.2.3 Triggers for Automatic Update.....	613
39.2.4 Access Authentication with HTTP Server.....	614
39.2.5 Querying Provisioning Server for Updated Files	614
39.2.6 File Download Sequence.....	617
39.2.7 Cyclic Redundancy Check on Downloaded Configuration Files	618
39.2.8 MAC Address Placeholder in Configuration File Name.....	618
39.2.9 File Template for Automatic Provisioning	619
39.2.10 Automatic Update Configuration Examples.....	621
39.2.10.1 Automatic Update for Single Device	621
39.2.10.2 Automatic Update from Remote Servers	622
39.2.10.3 Automatic Update for Mass Deployment.....	624
40 Restoring Factory Defaults	627
40.1 Restoring Factory Defaults through CLI.....	627
40.2 Restoring Factory Defaults through Web Interface	628

40.3 Restoring Defaults through ini File	628
Status, Performance Monitoring and Reporting	629
41 System Status	631
41.1 Viewing Device Information	631
41.2 Viewing Ethernet Port Information	632
41.3 Viewing Hardware Components Status	633
41.4 Reporting DSP Utilization through SNMP MIB.....	635
42 Carrier-Grade Alarms.....	637
42.1 Viewing Active Alarms	637
42.2 Viewing History Alarms.....	637
43 Performance Monitoring.....	639
43.1 Viewing MOS per Media Realm.....	639
43.2 Viewing Quality of Experience	640
43.3 Viewing Average Call Duration	641
44 VoIP Status	643
44.1 Viewing Active IP Interfaces	643
44.2 Viewing Ethernet Device Status.....	643
44.3 Viewing Static Routes Status.....	643
44.4 Viewing Registered Users	644
44.5 Viewing Registration Status.....	645
44.6 Viewing Proxy Set Status	645
45 Reporting Information to External Party	649
45.1 Configuring RTCP XR	649
45.2 Generating Call Detail Records	653
45.2.1 CDR Field Description	653
45.2.1.1 CDR Fields for SBC Signaling	653
45.2.1.2 CDR Fields for SBC Media.....	657
45.2.1.3 CDR Fields for Locally Stored SBC	659
45.2.2 Customizing CDRs for SBC Calls.....	660
45.2.3 Storing CDRs on the Device.....	663
45.2.4 Configuring CDR Reporting	666
45.3 Configuring RADIUS Accounting	667
Diagnostics	675
46 Syslog and Debug Recording	677
46.1 Configuring Log Filter Rules	677
46.1.1 Filtering IP Network Traces	681
46.2 Configuring Syslog	682
46.2.1 Syslog Message Format.....	682
46.2.1.1 Event Representation in Syslog Messages.....	684
46.2.1.2 Identifying AudioCodes Syslog Messages using Facility Levels.....	685
46.2.1.3 Syslog Fields for Answering Machine Detection (AMD)	686

46.2.1.4	SNMP Alarms in Syslog Messages.....	686
46.2.2	Configuring Web User Activities to Report to Syslog	687
46.2.3	Configuring Syslog Debug Level	689
46.2.4	Configuring Address of Syslog Server.....	690
46.2.5	Enabling Syslog	690
46.2.6	Viewing Syslog Messages	690
46.2.7	Viewing Web User Activity Logs	692
46.3	Configuring Debug Recording	693
46.3.1	Configuring Address of Debug Recording Server	693
46.3.2	Collecting Debug Recording Messages	693
46.3.3	Debug Capturing on Physical VoIP Interfaces	695
47	Creating Core Dump and Debug Files upon Device Crash	697
48	Testing SIP Signaling Calls	699
48.1	Configuring Test Call Endpoints	699
48.2	Starting and Stopping Test Calls	703
48.3	Viewing Test Call Statistics.....	704
48.4	Configuring DTMF Tones for Test Calls.....	706
48.5	Configuring Basic Test Calls.....	706
48.6	Test Call Configuration Examples.....	708
49	Pinging a Remote Host or IP Address.....	711
Appendix		713
50	Dialing Plan Notation for Routing and Manipulation.....	715
51	Configuration Parameters Reference	719
51.1	Management Parameters	719
51.1.1	General Parameters	719
51.1.2	Web Parameters.....	720
51.1.3	Telnet Parameters	724
51.1.4	ini File Parameters.....	724
51.1.5	SNMP Parameters.....	725
51.1.6	Serial Parameters	729
51.1.7	Auxiliary and Configuration File Name Parameters	729
51.1.8	Automatic Update Parameters	731
51.2	Networking Parameters	735
51.2.1	Ethernet Parameters.....	735
51.2.2	Multiple VoIP Network Interfaces and VLAN Parameters	736
51.2.3	Routing Parameters.....	736
51.2.4	Quality of Service Parameters.....	737
51.2.5	NAT Parameters	738
51.2.6	DNS Parameters.....	739
51.2.7	DHCP Parameters	740
51.2.8	NTP and Daylight Saving Time Parameters.....	742
51.3	Debugging and Diagnostics Parameters.....	743
51.3.1	General Parameters	744
51.3.2	SIP Test Call Parameters	744
51.3.3	Syslog, CDR and Debug Parameters.....	745
51.3.4	Resource Allocation Indication Parameters.....	751
51.4	HA Parameters.....	752
51.5	Security Parameters	754

51.5.1	General Security Parameters	754
51.5.2	HTTPS Parameters	756
51.5.3	SRTP Parameters.....	757
51.5.4	TLS Parameters.....	759
51.5.5	SSH Parameters.....	761
51.5.6	IDS Parameters	762
51.5.7	OCSP Parameters	763
51.6	Quality of Experience Parameters	763
51.7	Control Network Parameters	767
51.7.1	IP Group, Proxy, Registration and Authentication Parameters	767
51.7.2	Network Application Parameters	775
51.8	General SIP Parameters	778
51.9	Coders and Profile Parameters.....	793
51.10	Channel Parameters.....	795
51.10.1	Voice Parameters	795
51.10.2	Coder Parameters	798
51.10.3	DTMF Parameters	798
51.10.4	RTP, RTCP and T.38 Parameters.....	800
51.11	SBC Parameters	804
51.11.1	Supplementary Services.....	818
51.12	IP Media Parameters.....	820
51.13	Services	824
51.13.1	SIP-based Media Recording Parameters.....	824
51.13.2	RADIUS and LDAP Parameters	825
51.13.2.1	General Parameters	825
51.13.2.2	RADIUS Parameters	826
51.13.2.3	LDAP Parameters	828
51.13.3	Least Cost Routing Parameters	830
51.13.4	Call Setup Rules Parameters	831
51.13.5	HTTP-based Services.....	831
51.13.6	HTTP Proxy Parameters.....	832
52	SBC and DSP Channel Capacity	835
52.1	Signaling-Media Sessions & User Registrations	835
52.2	Channel Capacity and Capabilities	836
53	Technical Specifications	837

This page is intentionally left blank.

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: March-25-2018

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>.

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 2600 SBC Hardware Installation Manual
Complementary Guides
CLI Reference Guide
SNMP User's Guide
SBC Design Guide
Recommended Security Guidelines Configuration Note
SIP Message Manipulations Quick Reference Guide
SAS Application Configuration Guide
Utility Guides
INI Viewer & Editor Utility User's Guide
AcBootP Utility User's Guide
CLI Wizard User's Guide

Notes and Warnings



Note: The device is an indoor unit and therefore, must be installed only **INDOORS**. In addition, Ethernet port interface cabling must be routed only indoors and must not exit the building.



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 (eyay@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
41780	Initial document release for Version 7.0.
41783	New parameters: IPGroup_UIFormat; PM_EnableThresholdAlarms; NetworkNodeId; HTTPProxyApplication; HTTPInterface; HTTPProxyService; HTTPProxyHost; EMSService. New features: HTTP-based Proxy Services; Viewing Information of Installed Dial Plan File; Viewing the Installed User Info File Name; Upgrading SBC Capacity Licenses by License Pool Manager Server; CDR Fields for Locally Stored SBC; Storing CDRs on the Device. Updated parameter descriptions: Session Limit; Session Timeout; LoggingFilters; PhysicalPortsTable_GroupStatus; LdapConfiguration_Interface; SRD_SBCOperationMode (option 2); SRD_EnableUnAuthenticatedRegistrations; SRD_SBCRegisteredUsersClassificationMethod (option 2); SIPInterface_EnableUnAuthenticatedRegistrations; IPGroup_SBCOperationMode (default and option 3 removed); ProxySet_DNSResolveMethod (new option 3); SBCAdmissionControl_Reservation; SBCAdmissionControl_Rate; IP2IPRouting_DestType; IP2IPRouting_DestIPGroupName; IP2IPRouting_DestSIPInterfaceName; WebSessionTimeout; EnableSBCMediaSync; SSHMaxSessions (default). Updated sections: Technical Specifications; Configuring Web Session and Access Settings; Configuring Automatic Date and Time using SNTP; Configuring Date and Time Manually; Configuring the Time Zone; Configuring Daylight Saving Time; Media (RTP/RTCP/T.38); Answering Machine Detection (AMD); SIP-based Media Recording (warning added, etc.); Avaya UCID; Enhanced 9-1-1 Support for Lync Server (note added); Cloning SRDs; Configuring SIP Message Manipulation; Configuring Log Filter Rules; Configuring Address of Debug Recording Server; Starting and Stopping Debug

LTRT	Description
	Recording (removed); Configuring Ethernet Port Groups. Maximum table rows of following tables were updated: TLS Contexts table; SBC Routing Policy table; SBC User Info table. Miscellaneous: maximum IP addresses per host name was updated for internal DNS (Internal DNS table); statement added that R-factor VoIP metrics unavailable if RTCP XR feature is unavailable (not licensed or disabled); statement removed that IP Group ID 0 cannot be used; SBCEnableAASTRASurvivabilityNotice parameter modified to SBCEnableSurvivabilityNotice; maximum concurrent WebRTC sessions added; WAN parameters were removed; CLI for Web Users table.
41784	New sections: Enabling LDAP Searches for Numbers with Characters; File Template for Automatic Provisioning; Configuring Dial Plans New parameters: LdapConfiguration_VerifyCertificate; GW Group Registered IP Address; GW Group Registered Status; CDRLocalMaxNomOfFiles; CDR Syslog Sequence Number; HAUinitIdName; HARemoteAddress; HARevertiveEnabled; HAPriority;LDAPNumericAttributes; FeatureKeyURL; TemplateUrl; AupdFilesList; IP2IPRouting_SrcTags; IP2IPRouting_DestTags; IPOutboundManipulation_SrcTags; IPOutboundManipulation_DestTags; DialPlans; DialPlanRule; IP2IPRouting_Trigger; SIPChallengeCachingMode; MaxCallDuration G.727 removed; NFS removed; Web browser requirement for Web interface; LDAP cache size. Parameters removed: SRD_SBCRegisteredUsersClassificationMethod; IpProfile_RemoteBaseUDPPort Updated parameter descriptions: IpProfile_DisconnectOnBrokenConnection; TLSContexts_TLSVersion; DeviceTable_Tagging; BaseUDPPort; HTTPRemoteServices_Policy; HTTPRemoteServices_VerifyCertificate; HTTPInterface_VerifyCert; HTTPProxyHost_VerifyCert; CpMediaRealm_PortRangeEnd; CpMediaRealm_PortRangeStart; SIPInterface_UDPPort; IPGroup_Type; IPGroup_SIPGroupName; IPGroup_ClassifyByProxySet; IPGroup_InboundManSet; IPGroup_OutboundManSet; IPGroup_SBCPSAPMode; MessageManipulations_RowRole; CodersGroup0_CoderSpecific; CodersGroupX_CoderSpecific; Classification_SrcAddress; IP2IPRouting_GroupPolicy; BaseUDPPort; RTCPXREscIP; SBCUserRegistrationGraceTime; Coder payload types; RejectCancelAfterConnect Updated sections: Configuring Underlying Ethernet Devices; Configuring RTP Base UDP Port; Enabling the E9-1-1 Feature; Configuring IP Groups (matching methodology); Configuring SIP Message Manipulation; Configuring Classification Rules; Configuring SBC IP-to-IP Routing; Network Topology Types and Rx/Tx Ethernet Port Group Settings; Initial HA Configuration; Configuring RTCP XR; Configuring CDR Reporting; Customizing CDRs for SBC Calls; Storing CDRs on the Device; Configuring RADIUS Accounting.

LTRT	Description
41788	- Updated sections: Accessing the Web Interface (autocompletion); Advanced User Accounts Configuration (password definition); CLI-Based Management (permissible user accounts); Understanding Configuration Modes (Note typo); Configuring TLS Certificate Contexts (TLS versions); Configuring Underlying Ethernet Devices (max. VLANs); First Incoming Packet Mechanism (NAT by Signaling); Configuring Firewall Settings (Note); Centralized Third-Party Routing Server or ARM (Configure routing); Configuring the SEM Server (port removed); Registration Refreshes; Configuring Dial Plans; Configuring SBC User Info Table in Loadable Text File (syntax); MAC Address Placeholder in Configuration File Name (Note); Configuring RTCP XR (PublicationIPGroupID); Storing CDRs on the Device (ftp removed, Web names); Maintaining Same Syslog SID/BID over Multiple Devices (removed); - New sections: Clearing the LDAP Cache - Updated parameters: CallSetupRules_ActionType (typo value); IpProfile_MediaIPVersionPreference; IPProfile_SBCIceMode (value name); ConditionTable_Name (max. chars); CLIPrivPass; NATMode (values): Max Generated Register Rate; SendAcSessionIDHeader (removed); RTPOnlyMode (removed); SBCUserRegistrationGraceTime - New parameters: WebLoginBlockAutoComplete; EnforcePasswordComplexity; CDRLocalMaxFileSize; CDRLocalMaxNomOfFiles; CDRLocalInterval; GeneratedRegistersInterval; PublicationIPGroupID
41791	- Updated sections: Product Overview; CLI; Customizing the Web Interface; Configuring TLS Certificate Contexts; Creating a Login Welcome Message; Configuring Proxy Sets; Configuring Multi-Line Extensions and Supplementary Services; Configuring TLS Certificate Contexts; Configuring Proxy Sets; Direct Media; Pre-Configured IP Groups; Pre-Configured IP-to-IP Routing Rules; WebRTC; Pre-Configured IP Groups; Pre-Configured IP-to-IP Routing Rules; Initial HA Configuration; Software License Key; Viewing Device Information; Viewing Proxy Set Status; Customizing CDRs for SBC Calls; Configuring CDR Reporting; Configuring RADIUS Accounting; Technical Specifications - New sections: Customizing the Web Interface; Viewing the Device's Product Key - Updated parameters: TLSContexts_ServerCipherString; TLSContexts_ClientCipherString; NATTranslation_SourceStartPort; NATTranslation_SourceEndPort; NATTranslation_TargetStartPort; NATTranslation_TargetEndPort; SIPInterface_SBCDirectMedia; IpProfile_SBCUseSilenceSupp; IPProfile_SBCDirectMediaTag; IPOutboundManipulation_PrivacyRestrictionMode; DialPlans_Name; DialPlanRule_RuleIndex; DialPlanRule_Name; HA Configuration; Software License Key; WelcomeMessage; EnableCoreDump; ProxySet; PrackMode; SessionExpiresDisconnectTime; RADIUSRetransmission; RadiusTO; UserInfoFileUrl; - New parameters: ProxySet_SuccessDetectionRetries; ProxySet_SuccessDetectionInterval; ProxySet_FailureDetectionRetransmissions; ProxySet_MinActiveServersLB; UseProductName; UserProductName; UseWebLogo; WebLogoText; LogoWidth; WebFaviconFileUrl; EnableNonCallCdr; UseProductName; LogoFileName
41793	Updates sections: CLI; Configuring Web User Accounts (typo); Disabling Enabling SNMP; Configuring TLS Certificate Contexts; Assigning CSR-based Certificates to TLS Contexts (SHAI); Generating Private Keys for TLS Contexts (4096); Configuring IP Network Interfaces (NOTE); Configuring Firewall Settings; Assigning IDS Policies; Configuring LDAP Servers (max); Configuring Call Setup Rules; Configuring SIP Response Codes for Alternative Routing Reasons; Configuring Dial Plans (priority); Creating Core Dump and Debug Files upon Device Crash (reset); Configuring DTMF Tones for Test Calls; Configuring Basic Test Calls; Configuring SBC Test Call with

LTRT	Description
	External Proxy (removed) - Updated parameters: TLSContexts_ServerCipherString; TLSContexts_ClientCipherString; AccessList_Start_Port; AccessList_End_Port; SIPInterface_InterfaceName; ProxySet_ProxyName; MessageManipulations_ManipulationName; MessagePolicy_Name; IpProfile_SBCUseSilenceSupp; [ManipulationName; IpProfile_SBCPlayHeldTone; SBCAdmissionControl_AdmissionControlName; SBCAdmissionControl_Rate; Classification_ClassificationName; IP2IPRouting_RouteName; SBCRoutingPolicy_Name; IPInboundManipulation_ManipulationName; IPOutboundManipulation_ManipulationName; Test_Call_Play (tone type); EnableWebAccessFromAllInterfaces; ResetWebPassword; DisableSNMP; KeepAliveTrapPort (default); SBCtestID (removed); EnableCoreDump; SSHMaxLoginAttempts; IgnoreAlertAfterEarlyMedia; ECNLPMODE; ProxySet_IsProxyHotSwap; EnablePChargingVector (removed) - New parameters: TLSContexts_DTLSVersion; TLSContexts_DHKeySize; CustomerSN
41796	- Updated parameters: IP Profile; IPv4 Interface Name; IPv6 Interface Name; SBC Operation Mode; SBC Operation Mode; Name; Proxy Keep-Alive; Silence Suppression (Coders Group); Media Security Method; SDP Ptime Answer; Preferred Ptime; SetupTime; ConnectTime; Release Time; Embedded Telnet Server; Disable SNMP; Syslog Optimization; RTPAuthenticationDisableTx; RTPEncryptionDisableTx, RTCPEncryptionDisableTx; SBCKeepContactUserinRegister; CallSetupRules - Updated sections; Disabling SNMP; Network Address Translation Support; Configuring Voice Settings; Fax/Modem Transport Modes; Configuring Media (SRTP) Security; SIP-based Media Recording; Configuring SIP Recording Rules; Configuring Call Setup Rules; Configuring Media Realm Extensions; Configuring SIP Message Manipulation; Interworking Media Security Protocols; Interworking SIP Early Media; Configuring Call Preemption for SBC Emergency Calls; DHCP-based Provisioning; Automatic Update from Remote Servers; Viewing Active Alarms; CDR Fields for Locally Stored SBC; Configuring CDR Reporting; Configuring RADIUS Accounting - New sections: Configuring Dual Registration - New parameters: Query Target; Adapt RFC2833 BW to Voice coder; ActiveAlarmTableMaxSize; TimeZoneFormat; SRTPTunnelingValidateRTPRxAuthentication Deleted parameters: Silence Suppression (IPProfile_SCE); SBCRemoveSIPSFfromNonSecuredTransport; SIPRecTimeStamp - Deleted parameters: EnableSIPRemoteReset; IsCiscoSCEMode; EnableSilenceCompression

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 Introduction

This User's Manual is intended for the professional person responsible for installing, configuring and managing the AudioCodes product (hereafter, referred to as *device*). The document provides the information you need to configure and manage the device.

1.1 Product Overview

AudioCodes' Mediant 2600 E-SBC (hereafter referred to as *device*) is a member of AudioCodes family of Enterprise Session Border Controllers (E-SBC), enabling connectivity and security between small medium businesses (SMB) and service providers' VoIP networks. The device is a fully featured enterprise-class SBC provides a secured voice network deployment based on a Back-to-Back User Agent (B2BUA) implementation. The SBC functionality provides perimeter defense for protecting the enterprise from malicious VoIP attacks; mediation for allowing the connection of any IP PBX to any service provider; and service assurance for service quality and manageability.

The device also offers call "survivability" solutions using its Stand Alone Survivability (SAS) or Cloud Resilience Package applications, ensuring service continuity to enterprises served by a centralized SIP-based IP-Centrex server or branch offices of distributed enterprises. Call survivability enables internal office communication between SIP clients in the case of disconnection from the centralized SIP IP-Centrex server or IP-PBX.

The device is a single full-height (FH) AMC module running the session border controller application, consisting of the following:

- 1.25 GHz multi-core CPU
- Eight Ethernet 10/100/1000Base-T ports, grouped into pairs where one port is active while the other is standby (1+1), auto-negotiation, half- and full-duplex modes, and straight-through and crossover cable detection
- 1+1 power load-sharing and redundancy by providing two Power Supply modules
- 1+1 High Availability by deploying two Mediant 2600 devices, each connected to a different network (in the same subnet). In case of a network failure in one network, a switch over of traffic to the standby Mediant 2600 occurs.

The device supports an optional Media Processing Module (MPM), providing resources for additional IP-to-IP transcoding sessions.

The device allows full management through its HTTP/S-based embedded Web server. This user-friendly Web interface allows remote configuration using any standard Web browser (such as Microsoft™ Internet Explorer™). The device can also be managed through AudioCodes Element Management System (EMS), and SNMP v2/v3.



Note: For maximum call capacity figures, see "SBC and DSP Channel Capacity" on page 835.

1.2 Typographical Conventions

This document uses the following typographical conventions to convey information:

Table 1-1: Typographical Conventions

Convention	Description	Example
Boldface font	- Buttons in the Web interface. - Optional parameter values in the Web interface. - Navigational path in the Web interface. - Toolbar buttons in the Web interface.	Click the Add button.
Text enclosed by double apostrophe (" ")	Text that you need to type.	Enter the value "10.10.1.1".
Courier font	CLI commands.	At the prompt, type the following: # configure system
Text enclosed by square brackets ([])	Ini file parameter.	Configure the [GWDebugLevel] parameter to 1.
Text enclosed by single apostrophe (' ')	Web parameters.	From the 'Debug Level' drop-down list, select Basic .
	Notes highlight important or useful information.	-
	Warnings alert you to potentially serious problems if a specific action is not taken.	-

1.3 Getting Familiar with Configuration Concepts and Terminology

Before using your device, it is recommended that you familiarize yourself with the basic configuration concepts and terminology. An understanding of the basic concepts and terminology will help you configure and manage your device more effectively and easily.

1.3.1 SBC Application

The objective of your configuration is to enable the device to forward calls between telephony endpoints in the SIP-based Voice-over-IP (VoIP) network. The endpoints (SIP entities) can be servers such as SIP proxy servers and IP PBXs, or end users such as IP phones. In the SIP world, the endpoints are referred to as SIP user agents (UA). The UA that initiates the call is referred to as the user agent client (UAC); the UA that accepts the call is referred to as the user-agent server (UAS).

The following table describes the main configuration concepts and terminology.

Table 1-2: Configuration Concepts and Terminology

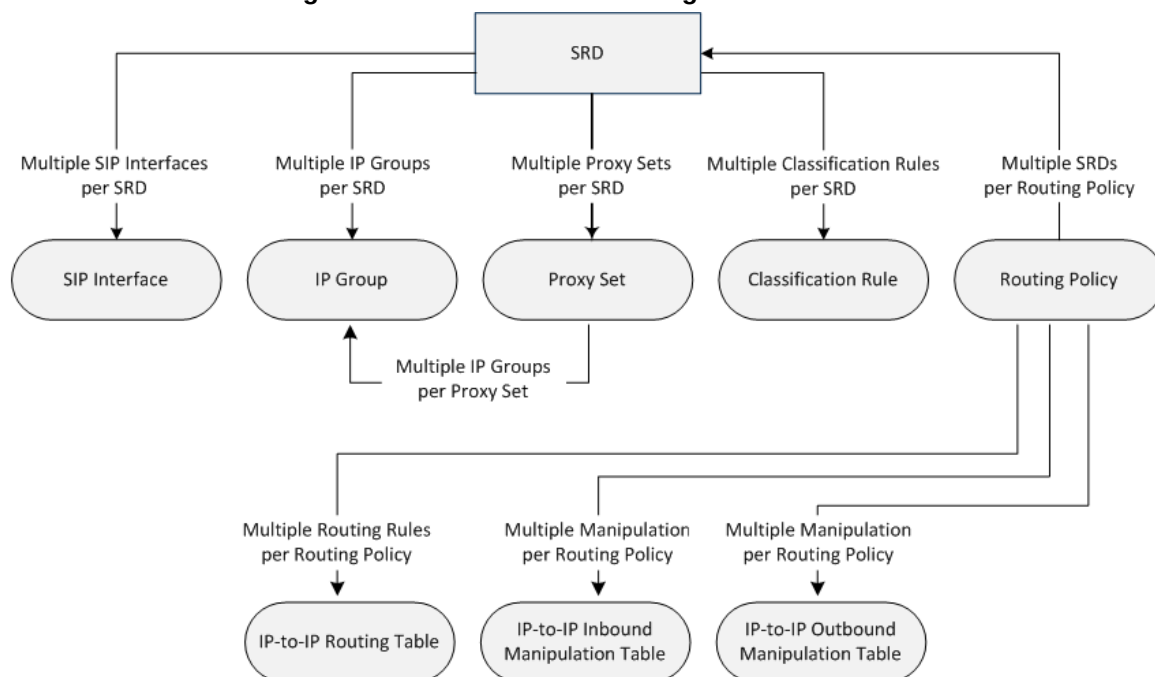
Configuration Terms	Description
IP Group	The IP Group is a logical representation of the SIP entity (UA) with which the device receives and sends calls. The SIP entity can be a server (e.g., IP PBX or SIP Trunk) or it can be a group of users (e.g., LAN IP phones). For servers, the IP Group is typically used to define the address of the entity (by its associated Proxy Set). IP Groups are used in IP-to-IP routing rules to denote the source and destination of the call.
Proxy Set	The Proxy Set defines the actual address (IP address or FQDN) of SIP entities that are servers (e.g., IP PBX). As the IP Group represents the SIP entity, to associate an address with the SIP entity, the Proxy Set is assigned to the IP Group. You can assign the same Proxy Set to multiple IP Groups (belonging to the same SRD).
SIP Interface	The SIP Interface represents a Layer-3 network. It defines a local listening port for SIP signaling traffic on a local, logical IP network interface. The term <i>local</i> implies that it's a logical port and network interface on the device. The SIP Interface is used to receive and send SIP messages with a specific SIP entity (IP Group). Therefore, you can create a SIP Interface for each SIP entity in the VoIP network with which your device needs to communicate. For example, if your VoIP network consists of three SIP entities -- a SIP Trunk, a LAN IP PBX, and remote WAN users -- a SIP Interface can be created for each of these Layer-3 networks. The SIP Interface is associated with the SIP entity, by assigning it to an SRD that is in turn, assigned to the IP Group of the SIP entity.
Media Realm	The Media Realm defines a local UDP port range for RTP (media) traffic on any one of the device's logical IP network interfaces. The Media Realm is used to receive and send media traffic with a specific SIP entity (IP Group). The Media Realm can be associated with the SIP entity, by assigning the Media Realm to the IP Group of the SIP entity, or by assigning it to the SIP Interface associated with the SIP entity.
SRD	The SRD is a logical representation of your entire SIP-based VoIP

Configuration Terms	Description
	network (Layer 5) containing groups of SIP users and servers. The SRD is in effect, the foundation of your configuration to which all other previously mentioned configuration entities are associated. For example, if your VoIP network consists of three SIP entities -- a SIP Trunk, a LAN IP PBX, and remote WAN users -- the three SIP Interfaces defining these Layer-3 networks would all assigned to the same SRD. Typically, only a single SRD is required and this is the recommended configuration topology. As the device provides a default SRD, in a single SRD topology, the device automatically assigns the SRD to newly created configuration entities. Thus, in such scenarios, there is no need to get involved with SRD configuration. Multiple SRDs are required only for multi-tenant deployments, where it "splits" the device into multiple logical devices. For multiple SRDs, the SRD can be configured with a Sharing Policy. The Sharing Policy simply means whether the SRD's resources (SIP Interfaces, IP Groups, and Proxy Sets) can be used by other SRDs. For example, if all tenants route calls with the same SIP Trunking service provider, the SRD of the SIP Trunk would be configured as a <i>Shared</i> Sharing Policy. SRDs whose resources are not shared, would be configured with an <i>Isolated</i> Sharing Policy.
IP Profile	The IP Profile is an optional configuration entity that defines a wide range of call settings for a specific SIP entity (IP Group). The IP Profile includes signaling and media related settings, for example, jitter buffer, voice coders, fax signaling method, SIP header support (local termination if not supported), and media security method. The IP Profile is in effect, the interoperability "machine" of the device, enabling communication between SIP endpoints that "speak" different call "languages". The IP Profile is associated with the SIP entity, by assigning the IP Profile to the IP Group of the SIP entity.
Classification	Classification is the process that identifies the incoming call (SIP dialog request) as belonging to a specific SIP entity (IP Group). There are three chronological classification stages, where each stage is done only if the previous stage fails. The device first attempts to classify the SIP dialog by checking if it belongs to a user that is already registered in the device's registration database. If this stage fails, the device checks if the source IP address is defined for a Proxy Set and if yes, it classifies it to the IP Group associated with the Proxy Set. If this fails, the device classifies the SIP dialog using the Classification table, which defines various characteristics of the incoming dialog that if matched, classifies the call to a specific IP Group. The main characteristics of the incoming call is the SIP Interface that is associated with the SRD for which the Classification rule is configured.
IP-to-IP Routing	IP-to-IP routing rules define the routes for routing calls between SIP entities. As the SIP entities are represented by IP Groups, the routing rules typically employ IP Groups to denote the source and destination of the call. For example, to route calls from the IP PBX to the SIP Trunk, the routing rule can be configured with the IP PBX as the source IP Group and the SIP Trunk as the destination IP Group. Instead of IP Groups, various other source and destination methods can be used. For example, the source can be a source host name while the destination can be an IP address or based on an LDAP query.
IP-to-IP Inbound and	IP-to-IP inbound and outbound manipulation lets you manipulate the user

Configuration Terms	Description
Outbound Manipulation	part of the SIP URI in the SIP message for a specific entity (IP Group). Inbound manipulation is done on messages received from the SIP entity; outbound manipulation is done on messages sent to the SIP entity. Inbound manipulation lets you manipulate the user part of the SIP URI for source (e.g., in the SIP From header) and destination (e.g., in the Request-URI line) in the incoming SIP dialog request. Outbound manipulation lets you manipulate the user part of the Request-URI for source (e.g., in the SIP From header) or destination (e.g., in the SIP To header) or calling name, in outbound SIP dialog requests. The IP-to-IP inbound and outbound manipulation are associated with the SIP entity, by configuring the rules with incoming characteristics such as source IP Group and destination host name. The manipulation rules are also assigned an SBC Routing Policy, which in turn, is assigned to IP-to-IP routing rules. As most deployments require only one SBC Routing Policy, the default Routing Policy is automatically assigned to the manipulation rules and to the routing rules.
SBC Routing Policy	SBC Routing Policy logically groups routing and manipulation (inbound and outbound) rules to a specific SRD. It also enables Least Cost Routing (LCR) for routing rules and associates an LDAP server for LDAP-based routing. However, as multiple Routing Policies are required only for multi-tenant deployments, for most deployments only a single Routing Policy is required. When only a single Routing Policy is required, handling of this configuration entity is not required as a default Routing Policy is provided, which is automatically associated with all relevant configuration entities.
Call Admission Control	Call Admission Control (CAC) lets you configure the maximum number of permitted concurrent calls (SIP dialogs) per IP Group, SIP Interface, SRD, or user.
Accounts	Accounts are used to register or authenticate a "served" SIP entity (e.g., IP PBX) with a "serving" SIP entity (e.g., a registrar or proxy server). The device does this on behalf of the "served" IP Group. Authentication (SIP 401) is typically relevant for INVITE messages forwarded by the device to a "serving" IP Group. Registration is for REGISTER messages, which are initiated by the device on behalf of the "serving" SIP entity.

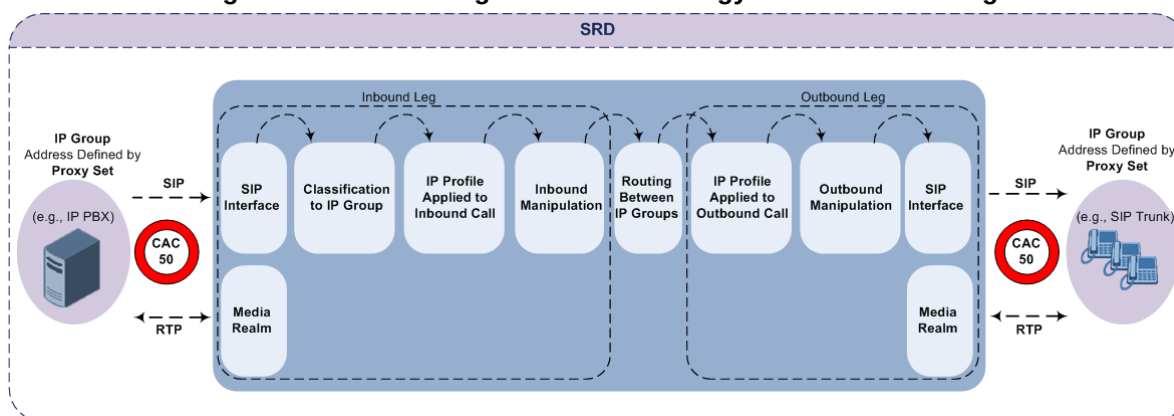
The associations between the configuration entities are summarized in the following figure:

Figure 1-1: Association of Configuration Entities



The main configuration entities and their involvement in the call processing is summarized in following figure. The figure is used only as an example to provide basic understanding of the configuration terminology. Depending on configuration and network topology, the call process may include additional stages or a different order of stages.

Figure 1-2: SBC Configuration Terminology for Call Processing



1. The device determines the SIP Interface on which the incoming SIP dialog is received and thus, determines its associated SRD.
2. The device classifies the dialog to an IP Group (origin of dialog), using a specific Classification rule that is associated with the dialog's SRD and that matches the incoming characteristics of the incoming dialog defined for the rule.
3. IP Profile and inbound manipulation can be applied to incoming dialog.
4. The device routes the dialog to an IP Group (destination), using the IP-to-IP Routing table. The destination SRD (and thus, SIP Interface and Media Realm) is the one assigned to the IP Group. Outbound manipulation can be applied to the outgoing dialog.

Part I

Getting Started with Initial Connectivity

2 Introduction

This part describes how to initially access the device's management interface and change its default IP address to correspond with your networking scheme.

This page is intentionally left blank.

3 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 3-1: Default VoIP LAN IP Address for OAMP

IP Address	Value
Application Type	OAMP + Media + Control
IP Address	192.168.0.2 - this IP address is assigned to the first Ethernet Port Group (top-left ports 1 & 2)
Prefix Length	255.255.255.0 (24)
Default Gateway	192.168.0.1
Underlying Device	vlan 1
Interface Name	O+M+C

This page is intentionally left blank.

4 Configuring VoIP LAN Interface for OAMP

You can change the IP address of the VoIP-LAN interface for OAMP, using any of the following methods:

- Embedded HTTP/S-based Web server - see "Web Interface" on page 35
- Embedded command line interface (CLI) - see "CLI" on page 37



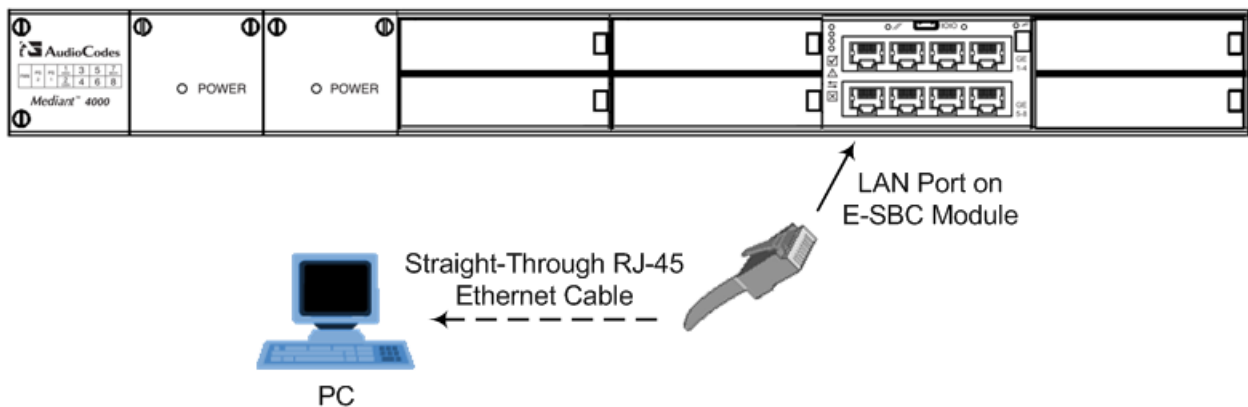
Note: If you are implementing the High Availability feature, see also HA Overview on page 555 for initial setup.

4.1 Web Interface

The following procedure describes how to change the IP address of the OAMP on the VoIP-LAN interface, using the Web-based management tool (Web interface). The default IP address is used to initially access the device.

➤ **To configure the VoIP-LAN IP Address for OAMP, using the Web interface:**

1. Connect the first Ethernet port group (top-left ports 1 and 2) located on the front panel directly to the network interface of your computer, using a straight-through Ethernet cable.



2. Change the IP address and subnet mask of your computer to correspond with the default OAMP IP address and subnet mask of the device.
3. Access the Web interface:
 - a. On your computer, start a Web browser and in the URL address field, enter the default IP address of the device; the Web interface's Web Login screen appears:

Figure 4-1: Web Login Screen

- b. In the 'Username' and 'Password' fields, enter the case-sensitive, default login username ("Admin") and password ("Admin").
 - c. Click **Login**.
4. Open the Physical Ports Settings page (Configuration tab > VoIP menu > Network > Physical Ports Table) and then configure the device's physical Ethernet port-pair (group) that you want to later assign to the OAMP interface. For more information, see Configuring Physical Ethernet Ports on page 131.
5. Open the Interface table (**Configuration** tab > **VoIP** menu > **Network** > **IP Interfaces Table**).

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 + Media	IPv4 Manual	192.168.0.2	24	192.168.0.1	Voice	0.0.0.0	0.0.0.0	vlan 1

6. Select the index row corresponding to the **OAMP + Media + Control** application type, and then click **Edit**.
7. Change the IP address to correspond with your network IP addressing scheme, for example:
 - IP Address: 10.8.6.86
 - Prefix Length: 24 (for 255.255.255.0)
 - Gateway: 10.8.6.85
 - Underlying Device: Select the Ethernet Device (VLAN and associated Ethernet port group) for OAMP
8. Click **Add**.
9. Save your settings by resetting the device with a flash burn (see "Resetting the Device" on page 575).
10. Disconnect the device from the PC and cable the device to your network. You can now access the management interface using the new OAMP IP address.



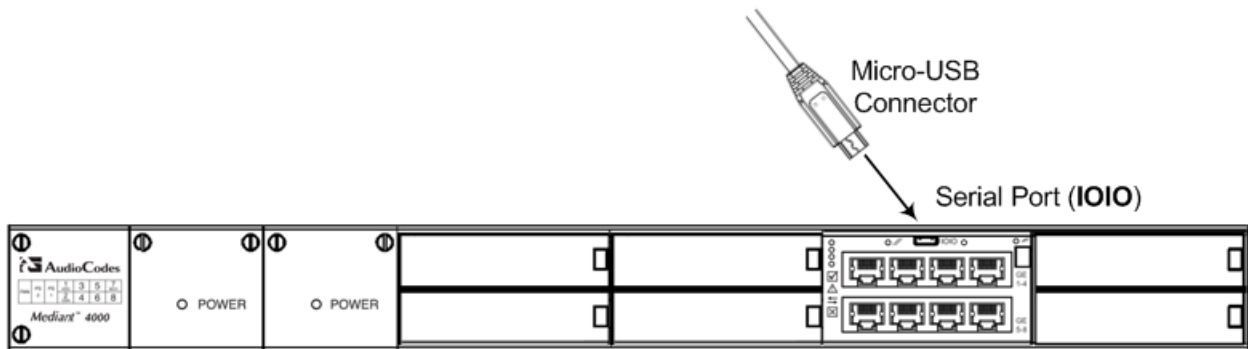
Note: When you complete the above procedure, change your PC's IP address to correspond with your network requirements.

4.2 CLI

This procedure describes how to configure the VoIP-LAN IP address for OAMP using the device's 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 configure the OAMP IP address in the CLI:**

1. Connect the RS-232 port of the device to the serial communication port on your computer. For more information, refer to the Hardware Installation Manual.



2. Establish serial communication with the device using a terminal emulator 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
3. At the CLI prompt, type the username (default is "Admin" - case sensitive):


```
Username: Admin
```
4. At the prompt, type the password (default is "Admin" - case sensitive):


```
Password: Admin
```
5. At the prompt, type the following:


```
enable
```
6. At the prompt, type the password again:


```
Password: Admin
```
7. Access the VoIP configuration mode:


```
# configure voip
```
8. Access the Interface table:


```
(config-voip)# interface network-if 0
```
9. Configure the IP address:


```
(network-if-0)# ip-address <IP address>
```
10. Configure the prefix length:


```
(network-if-0)# prefix-length <prefix length / subnet mask, e.g., 16>
```

11. Configure the Default Gateway address:
`(network-if-0)# gateway <IP address>`
12. Apply your settings:
`(network-if-0)# activate`
13. Cable the device to your network. You can now access the device's management interface using this new OAMP IP address.

Part II

Management Tools

5 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 43
- Command Line Interface (CLI) - see "CLI-Based Management" on page 81
- Simple Network Management Protocol (SNMP) - see "SNMP-Based Management" on page 93
- Configuration *ini* file - see "INI File-Based Management" on page 101

**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 719.

This page is intentionally left blank.

6 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 596).

6.1 Getting Acquainted with the Web Interface

This section provides a description of the Web interface.

6.1.1 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 11.0.13 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.

6.1.2 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 43).
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 6-1: Web Login Screen

The image shows a web login form 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 65.

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 67.
- 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**.
- By default, autocompletion of the login username is enabled whereby the 'Username' field offers previously entered usernames. To disable autocompletion, use the WebLoginBlockAutoComplete ini file parameter.
- 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
```



6.1.3 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 6-2: Main Areas of the Web Interface GUI

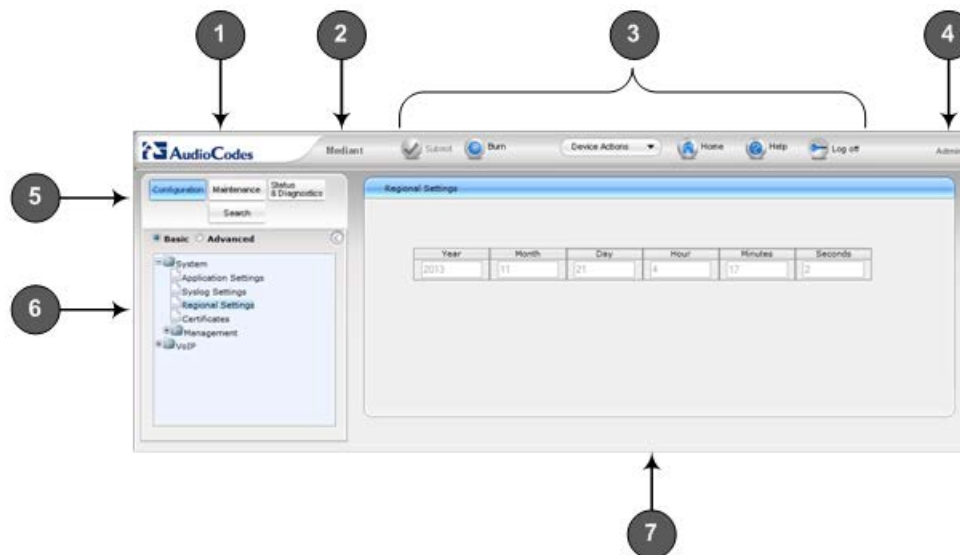


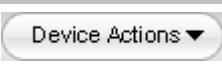
Table 6-1: Description of the Web GUI Areas

Item #	Description
1	AudioCodes company logo.
2	Product name.
3	Toolbar, providing frequently required command buttons. For more information, see "Toolbar Description" on page 46.
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 49) ■ Search tab: Enables a search engine for searching configuration parameters (see "Searching for Configuration Parameters" on page 56)
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 47.
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 49.

6.1.4 Toolbar Description

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

Table 6-2: Description of Toolbar Buttons

Icon	Button Name	Description
	Submit	Applies parameter settings to the device (see "Saving Configuration" on page 578). 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 578).
	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 607). ▪ 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 607). ▪ Reset: Opens the Maintenance Actions page for performing various maintenance procedures such as resetting the device (see "Resetting the Device" on page 575). ▪ Software Upgrade Wizard: Starts the Software Upgrade Wizard for upgrading the device's software (see "Software Upgrade Wizard" on page 601). ▪ Switch Over: Opens the High Availability Maintenance page for switching between Active and Redundant devices (see High Availability Maintenance on page 579). ▪ Reset Redundant: Opens the High Availability Maintenance page for resetting the Redundant device (see High Availability Maintenance on page 579).
	Home	Opens the Home page (see "Viewing the Home Page" on page 65).
	Help	Opens the Online Help topic of the currently opened configuration page (see "Getting Help" on page 59).
	Log off	Logs off a session with the Web interface (see "Logging Off the Web Interface" on page 60).
-	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.

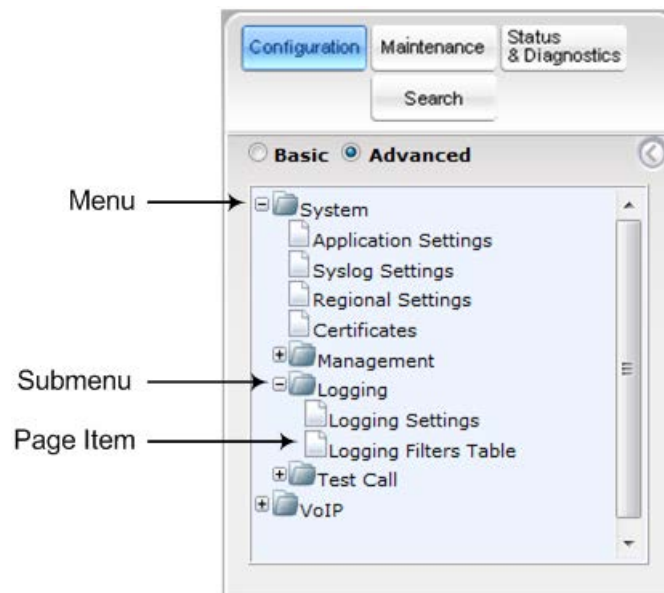
6.1.5 Navigation Tree

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 6-3: 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.

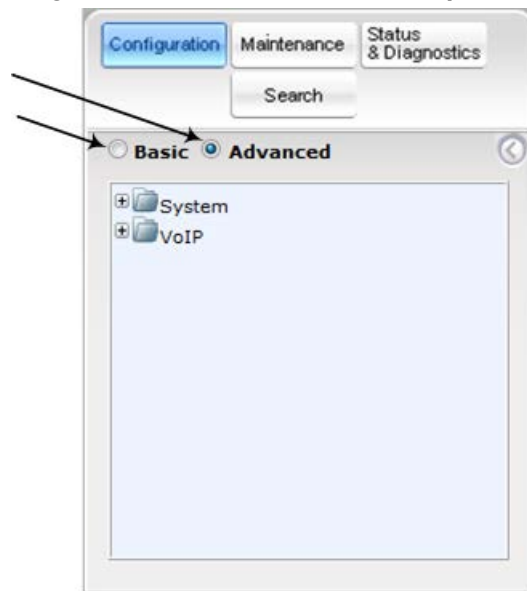
6.1.5.1 Displaying Navigation Tree in Basic and Full View

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 6-4: Basic and Full View Options



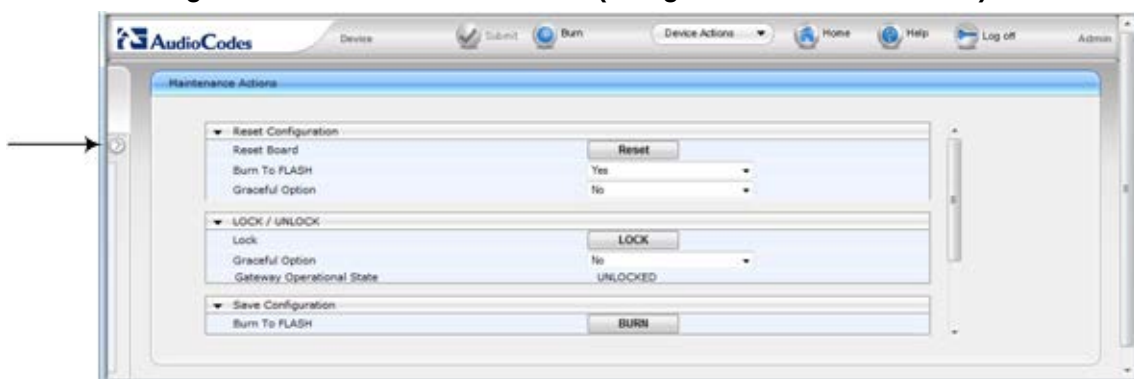
Note: After you reset the device, the Web GUI is displayed in **Basic** view.

6.1.5.2 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 6-5: Show and Hide Button (Navigation Pane in Hide View)



6.1.6 Working with Configuration Pages

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

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

6.1.6.2 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 49
- Displaying parameter groups - see "Showing / Hiding Parameter Groups" on page 50

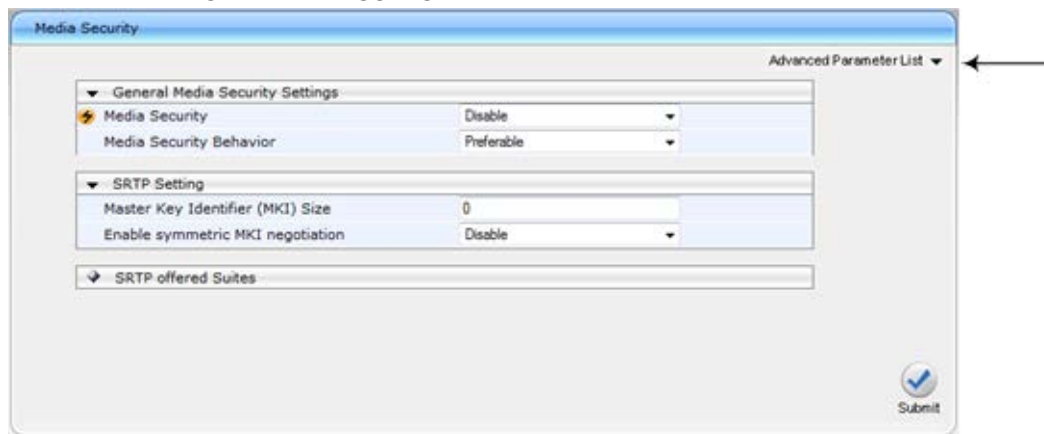
6.1.6.2.1 Displaying Basic and Advanced Parameters

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

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

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

Figure 6-6: Toggling between Basic and Advanced View



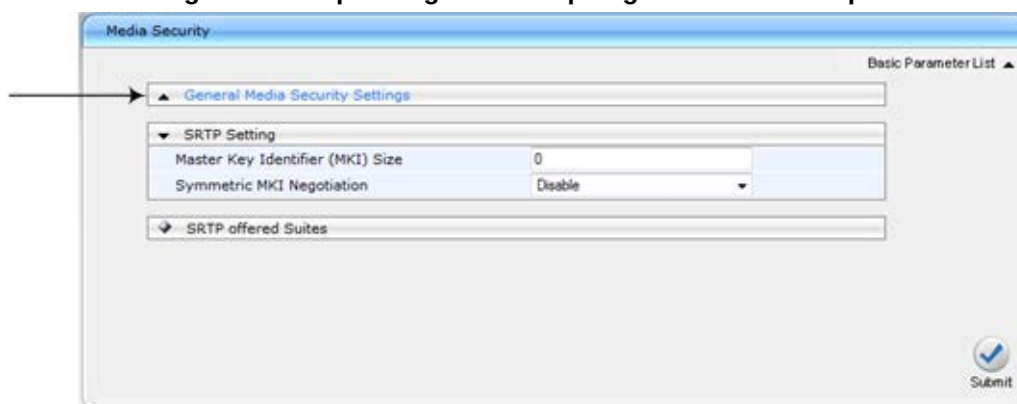
Notes:

- When the Navigation tree is in **Advanced** display mode (see "Navigation Tree" on page 47), 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.

6.1.6.2.2 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.

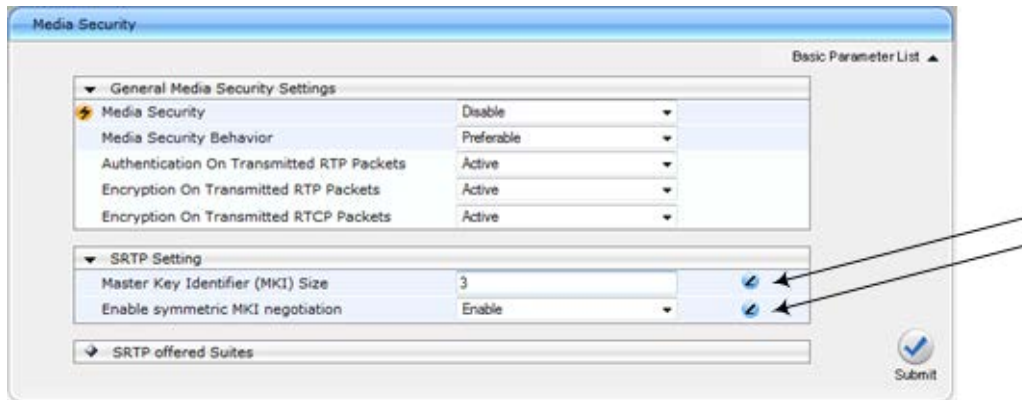
Figure 6-7: Expanding and Collapsing Parameter Groups



6.1.6.3 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 6-8: 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.

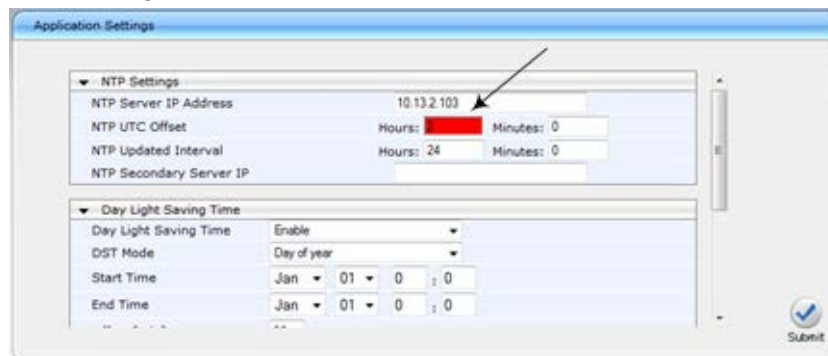
When you click **Submit**, modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect. Parameters displayed on the page with the lightning  icon take effect only after a device reset. For resetting the device, see "Resetting the Device" on page 575.



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 578).

If you enter an invalid 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 6-9: Value Reverts to Previous Valid Value



6.1.7 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 6-10: Displayed Details Pane

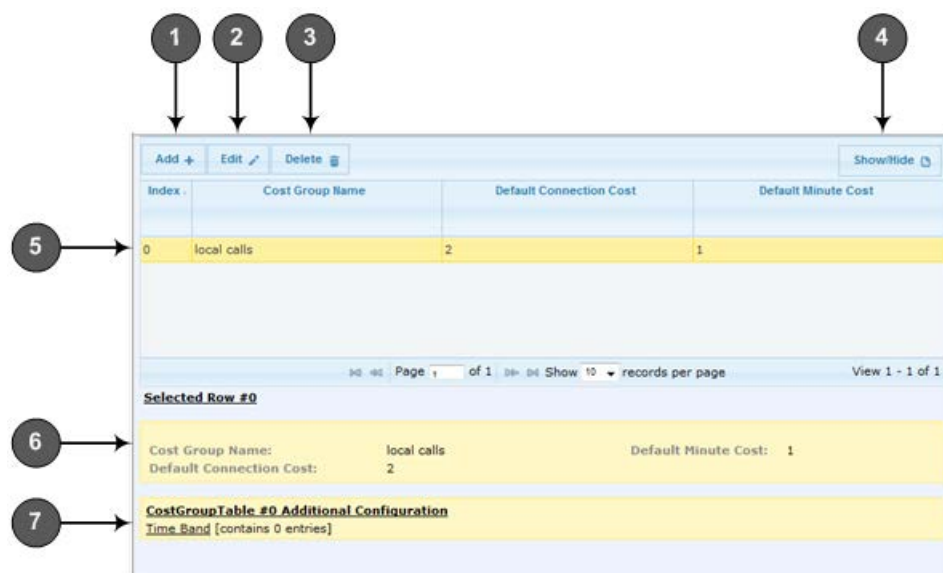


Table 6-3: Enhanced Table Design Description

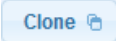
Item #	Button	
5	-	Selected index row entry for editing, deleting and showing configuration.
6	-	Displays the full configuration of the selected row when you click the Show/Hide button.
7	-	Links to access additional configuration tables related to the current configuration.

6.1.7.1 Table Toolbar Description

The configuration tables provide a toolbar with various buttons, as described below.

Table 6-4: Table Toolbar Description

Button	Name	
	Add	Adds a new index entry row to the table. When you click this button, the Add Row dialog box appears with parameters for configuring the new entry. When you have completed configuration, click the Add button in the dialog box to add it to the table.
	Edit	Edits the selected row. When you click this button, the Edit Row dialog box appears to modify the row entry. When you have completed configuration, click the Save button in the dialog box.
	Delete	Removes the selected row from the table. When you click this button, the Delete Row confirmation box appears requesting you to confirm deletion. Click Delete to accept deletion.
	Show / Hide	Toggles between displaying and hiding the full configuration of a selected row. The configuration is displayed below the table and is useful for tables containing many parameters, which cannot all be displayed in the work pane.
	Action	Provides a drop-down list with commands (e.g., Register and Unregister) relevant to the specific table (e.g., Account table). Note: The button only appears in certain tables.
	Insert	Adds a new table row at a selected index. You can add the row at any existing (configured) index number. If you select a row and then click the button, after configuring the parameters, the row is automatically added to the index of the selected row and the row that previously occupied the index row and all rows below it are moved one index down in the table. For example, if you select Index 2 and then click the button, the new row is assigned Index 2 and the row previously occupying Index 2 is moved down to Index 3 and so on. Notes: - The button is available only if the table is sorted according to 'Index' column; otherwise, the button is grayed out. For sorting tables, see "Sorting Tables by Column" on page 55. - The button appears only in certain tables.
	Down	Moves a selected row one index down. The index number of the row changes according to its new position in the table. The row that previously occupied the index row and all rows below it are moved one index down in the table. Notes: - The button is available only if the table contains more than one row and is sorted according to 'Index' column; otherwise, the button is grayed out. For sorting tables, see "Sorting Tables by Column" on page 55. - The button appears only in certain tables.

Button	Name	
	Up	Moves a selected row one index up. The index number of the row changes according to its new position in the table. The row that previously occupied the index row and all rows below it are moved one index down in the table. Notes: - The button is available only if the table contains more than one row and is sorted according to 'Index' column; otherwise, the button is grayed out. For sorting tables, see "Sorting Tables by Column" on page 55. - The button appears only in certain tables.
	Clone	Adds a new row with identical settings as the selected row. Note: The button only appears in certain tables.

6.1.7.2 Toggling Display Mode of Table Dialog Boxes

Add and Edit dialog boxes that appear when you click the **Add** and **Edit** buttons respectively, by default display parameters in Tab view, whereby parameters are grouped under tabs according to functionality (e.g., Rule, Action, and Status). You can change the view mode to Classic view, whereby all parameters appear in one list and whereby parameters are separated according to functionality by a heading instead of a tab.

➤ **To toggle between Tab and Classic view:**

- Click the **Classic View** or **Tabs View** link located at the bottom of the dialog box.

6.1.7.3 Scrolling through Table Pages

You can define the maximum number of rows (indices) to display in the table. To view additional rows, you can scroll through the table pages. The figure below shows the table page navigation area, which is located below the table:

Figure 6-11: Viewing Table Rows per Page



Table 6-5: Row Display and Page Navigation

Item #	Description
1	Defines the page that you want to view. Enter the required page number or use the following page navigation buttons: ➡ - Displays the next page ➡ - Displays the last page ⬅ - Displays the previous page ⬅ - Displays the first page
2	Defines the number of rows to display per page. You can select 5, 10 (default), or 20.
3	Displays the currently displayed number of rows out of the maximum configured.

6.1.7.4 Searching Table Entries

The configuration tables provide you with a search feature that lets you search any value (string or IP address) of a specified parameter (column) in the table. By default, searches are performed on all the table's parameters. You define the search using the search features, located on top of the table (on the table's toolbar), as shown in the example below:

Figure 6-12: Searching Table Entries

Interface Name	Primary DNS	Secondary DNS	Underlying Device
0.0.0.0	0.0.0.0	vlan 1	
0.0.0.0	0.0.0.0	vlan 1	

View 1 - 2 of 2

➤ **To search for a table entry:**

1. From the search drop-down list, select the table column name in which you want to search for the value.
2. In the search text box, enter the value for which you want to search.
3. Click **Search**.

If the device locates searched entries, the table displays only the rows in which the entries were found. If the search was unsuccessful, no rows are displayed and a message is displayed notifying you that no records were found.

To quit the search feature, click the **X** icon, displayed alongside the "Showing results for" message below the **Add** button.



Note: The search feature is supported only for certain tables.

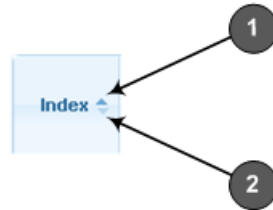
6.1.7.5 Sorting Tables by Column

You can sort table rows by any table column and in ascending (e.g., 1, 2 and 3, or a, b, and c) or descending (3, 2, and 1, or c, b, and a) order. For example, instead of the rows being sorted by the Index column in ascending order (e.g., 1, 2, and 3), you can sort the rows by Index column in descending order (e.g., 3, 2, and 1). By default, most tables are sorted by Index column in ascending order.

➤ **To sort table rows by column:**

1. Click the heading name of the column that you want to sort the table rows by; the up-down arrows appear alongside the heading name and the up button is bolded (see Item 1 in the figure below), indicating that the column is sorted in ascending order:

Figure 6-13: Sorting Table Rows by Column



2. To sort the column in descending order, click the heading name of the column again; only the down arrow appears bolded (see Item 2 in the figure above), indicating that the column is sorted in descending order.

6.1.8 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 6-14: Searched Result Screen

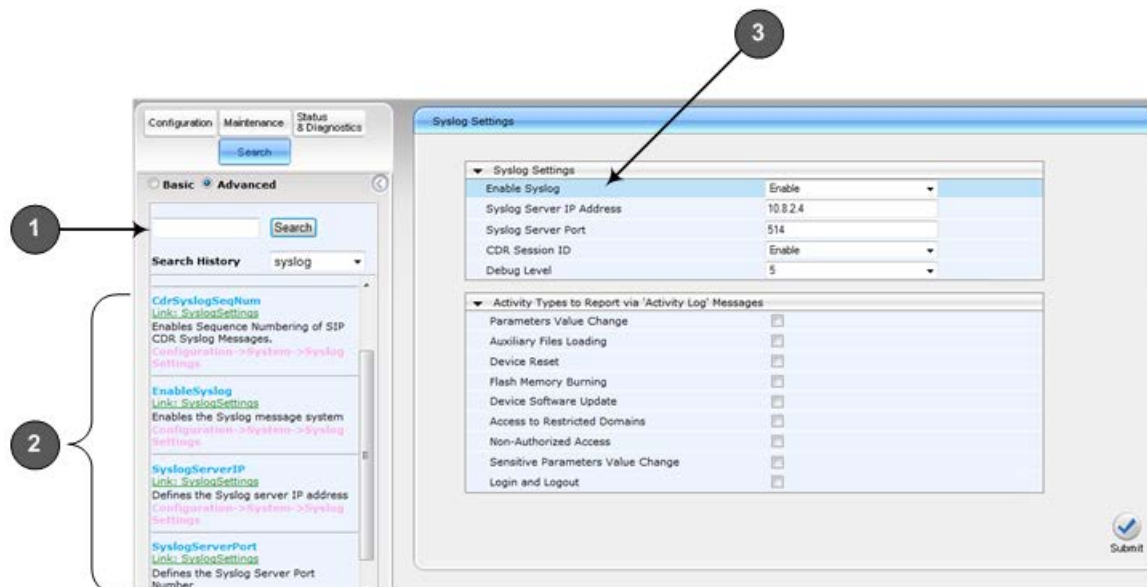


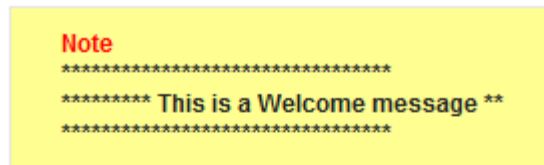
Table 6-6: Search Description

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

6.1.9 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 6-15: User-Defined Web Welcome Message after Login



Web Login

Username

Password

☐ Remember Me

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

Table 6-7: ini File Parameter for Welcome Login Message

Parameter	Description
[WelcomeMessage]	Enables and defines a Welcome message that appears on the Web Login page for logging in to the Web interface. The format of the ini file table parameter is: <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.

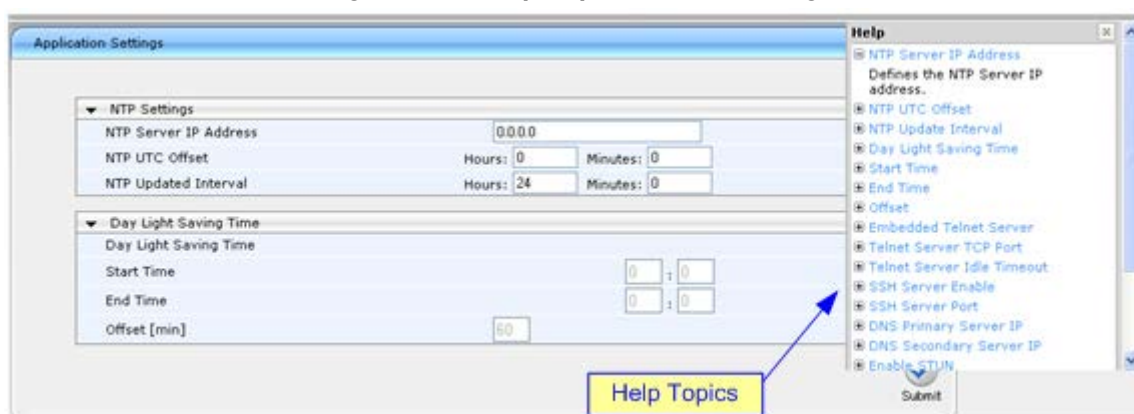
6.1.10 Getting Help

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 6-16: 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.

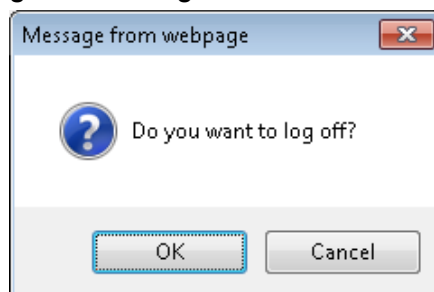
6.1.11 Logging Off the Web Interface

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 6-17: 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.

6.2 Customizing the Web Interface

You can customize the following elements of the device's Web interface (GUI):

- Corporate logo (see Replacing the Corporate Logo on page 60)
- Device's (product) name (see Customizing the Product Name on page 62)
- Favicon (see Customizing the Favicon on page 62)
- Login welcome message (see Creating a Login Welcome Message on page 58)



Note:

- The product name also affects other management interfaces.
- In addition to Web-interface customization, you can customize the following to reference your company instead of AudioCodes:
 - ✓ SNMP Interface: Product system OID (see the SNMPSysOid parameter) and trap Enterprise OID (see the SNMPTrapEnterpriseOid parameter).
 - ✓ SIP Messages: User-Agent header (see the UserAgentDisplayInfo parameter), SDP "o" line (see the SIPSDPSessionOwner parameter), and Subject header (see the SIPSubject parameter).

6.2.1 Replacing the Corporate Logo

You can replace the default corporate logo image (i.e., AudioCodes logo) that is displayed in the Web interface. The logo appears in the following Web areas:

- Web Login screen
- Menu bar

You can replace the logo with one of the following:

- A different image (see Replacing the Corporate Logo with an Image on page 61)

- Text (see Replacing the Corporate Logo with Text on page 61)

6.2.1.1 Replacing the Corporate Logo with an Image

You can replace the logo with a different image.

➤ **To customize the logo:**

1. Save your new logo image file in a folder on the same PC that you are using to access the device's Web interface.
2. In your browser's URL address field, append the case-sensitive suffix `"/AdminPage"` to the device's IP address (e.g., `http://10.1.229.17/AdminPage`).
3. Log in with your credentials; the Admin page appears.
4. On the left pane, click **Image Load to Device**; the right pane displays the following:

Figure 6-18: Customizing Web Logo

5. Use the **Browse** button to select your logo file, and then click **Send File**; the device loads the file.
6. If you want to modify the width of the image, in the 'Logo Width' field, enter the new width (in pixels) and then click the **Set Logo Width** button.
7. On the left pane, click **Back to Main** to exit the Admin page.
8. Reset the device with a save-to-flash for your settings to take effect.



Note:

- The logo image file type can be GIF, PNG, JPG, or JPEG.
- The logo image must have a fixed height of 24 pixels. The width can be up to 199 pixels (default is 145).
- The maximum size of the image file can be 64 Kbytes.
- Ignore the **ini Parameters** option, which is located on the left pane of the Admin page.

6.2.1.2 Replacing the Corporate Logo with Text

You can replace the logo with text.

➤ **To replace the logo with text:**

1. Create an ini file that includes the following parameter settings:


```
UseWebLogo = 1
WebLogoText = < your text >
```
2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 583).

3. Reset the device with a save-to-flash for your settings to take effect.

6.2.2 Customizing the Product Name

You can customize the device's product name. The name is displayed in various places in the management interfaces, as shown below using the customized name, "My Product Name":

- **Web Login screen**

- **Ini file "Board" field:**

```
Board: My Product Name
```

- **CLI prompt:**

```
My Product Name(config-system)#
```

- **To customize the device's product name:**

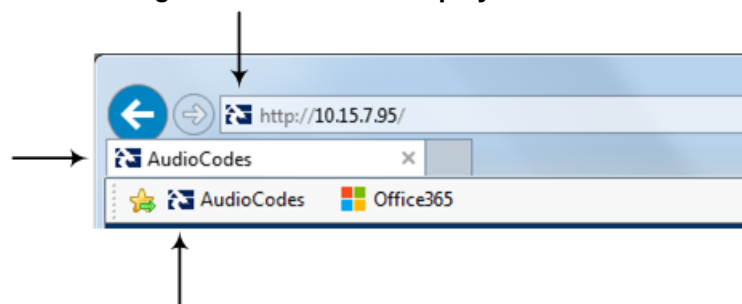
1. Create an ini file that includes the following parameter settings:

```
UseProductName = 1
UseProductName = < name >
```
2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 583).
3. Reset the device with a save-to-flash for your settings to take effect.

6.2.3 Customizing the Favicon

You can replace the default favicon (i.e., AudioCodes) with your own personalized favicon. Depending on the browser, the favicon is displayed in various areas of your browser, for example, in the URL address bar, on the page tab, and when bookmarked:

Figure 6-19: Favicon Display in Browser

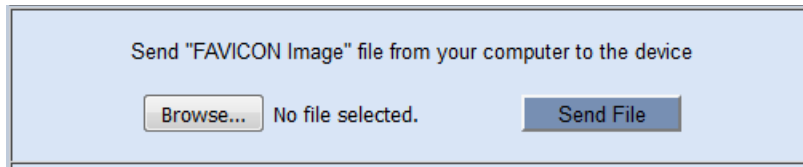


- **To customize the favicon:**

1. Save your new favicon file (.ico) in a folder on the same PC that you are using to access the device's Web interface.
2. In your browser's URL address field, append the case-sensitive suffix "/AdminPage" to the device's IP address (e.g., http://10.1.229.17/AdminPage).
3. Log in with your credentials; the Admin page appears.

4. On the left pane, click **Image Load to Device**; the right pane displays the following:

Figure 6-20: Customizing Favicon



5. Use the **Browse** button to select your favicon file, and then click **Send File**; the device loads the image file.
6. On the left pane, click **Back to Main** to exit the Admin page.
7. Reset the device with a save-to-flash for your settings to take effect.



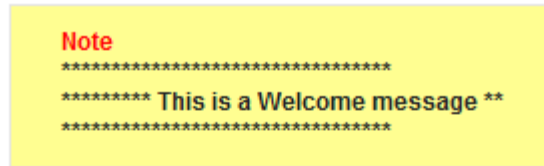
Note:

- The logo image file type can be ICO, GIF, or PNG.
- The maximum size of the image file can be 16 Kbytes.
- Ignore the **ini Parameters** option, which is located on the left pane of the Admin page.

6.2.4 Creating a Login Welcome Message

You can create a personalized welcome message that is displayed on the Web Login screen. The message always begins with the title "Note" and has a color background, as shown in the example below:

Figure 6-21: Creating Login Welcome Message



Web Login

Username

Password

☐ Remember Me

➤ **To create a login welcome message:**

1. Create an ini file that includes the WelcomeMessage table parameter. Use the parameter to configure your message, where each index row is a line in your message, for example:

```
[WelcomeMessage ]
FORMAT WelcomeMessage_Index = WelcomeMessage_Text;
WelcomeMessage 1 = "*****";
WelcomeMessage 2 = "*** This is a Welcome message! ***";
WelcomeMessage 3 = "*****";
[\WelcomeMessage]
```

2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 583).
3. Reset the device with a save-to-flash for your settings to take effect.

➤ **To remove the welcome message:**

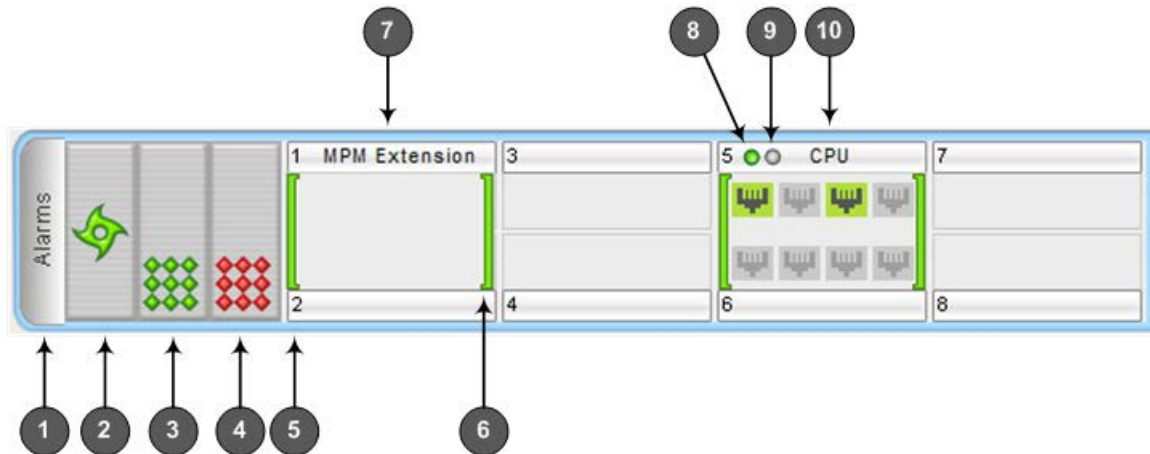
1. Load an empty ini file, using the Auxiliary Files page.
2. Reset the device with a save-to-flash for your settings to take effect.

6.3 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.



Note: For a description of the Home page when the device is in High Availability (HA) mode, see HA Status on the Home Page on page 557.

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

- **IP Address:** IP address of the device
 - **Subnet Mask:** Subnet mask address of the device
 - **Default Gateway Address:** Default gateway used by the device
 - **Firmware Version:** Software version running on the device
 - **Protocol Type:** Signaling protocol currently used by the device (i.e. SIP)
 - **Gateway Operational State:**
 - "LOCKED": device is locked (i.e. no new calls are accepted)
 - "UNLOCKED": device is not locked
 - "SHUTTING DOWN": device is currently shutting down
- To perform these operations, see "Basic Maintenance" on page 575.
- **High Availability:** Status of the device's HA mode

The table below describes the areas of the Home page.

Table 6-8: Home Page Description

Item #	Description
1	Displays the highest severity of an active alarm raised (if any) by the device: - Green: No alarms - Red: Critical alarm - Orange: Major alarm - Yellow: Minor alarm To view active alarms, click the Alarms area to open the Active Alarms page (see Viewing Active Alarms on page 637).
2	Fan tray unit status icon:  (green): Fan tray operating  (red): Fan tray failure To view detailed information on the device's hardware components, click these icons to open the Components Status page (see Viewing Hardware Components Status on page 633).
3	Power Supply module 2 status icon:  (green): Power Supply module is operating  (red): Power Supply module failure or no Power Supply module installed To view detailed information on the device's hardware components, click these icons to open the Components Status page (see Viewing Hardware Components Status on page 633).
4	Power Supply Unit 1 status indicator. See Item #3 for a description.
5	Chassis slot number.
6	Module status icon:  (green): Module has been inserted or is correctly configured  (gray): Module was removed. "Reserved" is displayed alongside the module's name  (red): Module failure. "Failure" is displayed instead of the module's name
7	Media Processing Module (MPM). Note: This is a customer-ordered module.
8	OK LED:  (red): Device undergoing startup  (green): Device startup has completed successfully

Item #	Description
9	Active LED: ■  (gray): No High Availability (HA). This appears if the HA feature is not included in the installed Software License Key, HA is not configured, or upon HA initialization failure. ■  (green): Active device - "stand-alone" state (i.e., Redundant device is missing and HA is unavailable) or in "Active" state and HA is enabled ■  (yellow): ✓ Steady on: Redundant device while HA is unavailable (before HA synchronization with Active device has completed or failed connecting to Active device). ✓ Blinking: Redundant device while HA is enabled or device undergoing failure process (and reporting failure information to flash memory).
10	SBC module, providing the 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 632).

6.4 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 6-9: Web User Access Levels and Privileges

User Level	Numeric Representation in RADIUS	Privileges
Security Administrator	200	Read / write privileges for all pages. It can create all user types and is the only one that can create the first Master user. Note: At least one Security Administrator user must 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).

User Level	Numeric Representation in RADIUS	Privileges
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 6-10: Pre-configured Web User Accounts

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

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



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 username and password.
- 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) using the *ini* file parameter DisableWebConfig (see "Web and Telnet Parameters" on page 719).
- You can define additional Web user accounts using a RADIUS server (see "RADIUS Authentication" on page 236).

6.4.1 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 6-11: Allowed Modifications per Web User Level

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



Notes:

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

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

Figure 6-22: 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.

6.4.2 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. Admin users have read-only privileges in the Web Users table. Monitor users have no access to this table.
- Only Security Administrator and Master users can add, edit, or delete users.
- 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 user types can change their own passwords. This is done in the Web Security Settings page (see "Configuring Web Security Settings" on page 76).
- 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 change only their passwords in the Web Security Settings page (see "Configuring Web Security Settings" on page 76). 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 through the Web interface. You can also configure it through CLI (configure system > create-users-table).

➤ To add Web user accounts with advanced settings:

1. Open the Web Users table:
 - 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 user accounts - Security Administrator ("Admin") and Monitor ("User"):

Figure 6-23: 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

View 1 - 2 of 2

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

Figure 6-24: Web Users Table - Add Record Dialog Box

The dialog box is titled "Add Record" and contains the following fields and values:

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

Buttons: Submit, Cancel

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

Table 6-12: Web User Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Username 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.
Password password	Defines the Web user's password. The valid value is a string of 8 to 40 ASCII characters. To ensure strong passwords, adhere to the following password complexity requirements: - Contain at least eight characters. - Contain at least two letters that are upper case (e.g., A). - Contain at least two letters that are lower case (e.g., a). - Contain at least two numbers (e.g., 4). - Contain at least two symbols (non-alphanumeric characters) (e.g., \$, #, %). - No spaces. - Contain at least four new characters that were not used in the previous password. Note: To enforce the password complexity requirements mentioned above, configure the EnforcePasswordComplexity to 1.

Parameter	Description
Status 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 Login = 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 Session and Access Settings" on page 76). 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. ▪ Inactivity = 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 Session and Access Settings" on page 76). 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 Inactivity 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.
Password Age 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.
Session Limit 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.
Session Timeout 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 Session and Access Settings" on page 76).

Parameter	Description
Block Duration 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 Session and Access Settings" on page 76). 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 Session and Access Settings" on page 76). Note: The 'Deny Authentication Timer' parameter relates to failed Web logins from specific IP addresses.
User Level 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.

6.5 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 6-25: Login Information Window



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

Close

6.6 Configuring Web Security Settings

This section describes how to secure Web-based management.

6.6.1 Configuring Secured (HTTPS) Web

By default, the device allows remote management (client) through HTTP and HTTPS. However, you can enforce secure Web access communication by configuring the device to accept only HTTPS.

➤ To configure secure Web access:

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

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

2. From the 'Secured Web Connection (HTTPS)' drop-down list, select **HTTPS Only**.
3. To enable two-way authentication whereby both management client and server are authenticated using X.509 certificates, from the 'Requires Client Certificates for HTTPS connection' drop-down list, select **Enable**.
4. In the 'HTTPS Cipher String' field, enter the cipher string for HTTPS (in OpenSSL cipher list format).
5. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

For more information on secured Web-based management including TLS certificates, see "TLS for Remote Device Management" on page 121.

6.6.2 Configuring Web Session and Access Settings

You can configure security features related to Web user sessions and access.

➤ To configure Web user sessions and access security:

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

Figure 6-26: Configuring Security Related to Web User Sessions and Access

Session	
Password Change Interval (minutes)	1440
User Inactivity Timeout (days)	90
Session Timeout (minutes)	15

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

2. Web user sessions:
 - a. 'Password Change Interval': Duration of the validity of Web login passwords. When the duration expires, the Web user must change the password in order to log in again.
 - b. 'User Inactivity Timeout': If the user has not logged into the Web interface within this defined duration, the status of the user becomes inactive and the user can no longer access the Web interface. The user can only log in to the Web interface if its status is changed (to "New" or "Valid") by an Administrator or a Master user.
 - c. 'Session Timeout': Duration of Web inactivity (i.e., no actions are performed in the Web interface) of a logged-in user, after which the Web session expires and the user is automatically logged off the Web interface and needs to log in again to continue the session. You can also configure the functionality per user in the Web Users table (see Advanced User Accounts Configuration on page 71), which overrides this global setting.
3. Web user access:
 - a. 'Deny Authentication Timer': Interval (in seconds) that the user needs to wait before the user can attempt to log in from the same IP address after reaching the maximum number of failed login attempts (see next step).
 - b. 'Deny Access On Fail Count': Number of failed login attempts after which the user is prevented access to the device for a user-defined duration (previous step).
4. Click **Submit**.

For a detailed description of the above parameters, see "Web Parameters" on page 720.

6.7 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.

6.8 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 Parameters" on page 720).

➤ **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 6-27: 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 6-28: 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 578.



Notes:

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

7 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.

7.1 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.

7.1.1 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: <Enable mode password>
#
```

The Enable mode prompt is "#".



Note: The default password for accessing the Enable mode is "Admin" (case-sensitive). To change the 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)#

7.1.2 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 7-1: 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: (config-voip)# voip-network ? dns Enter voip-network dns ip-group IP Group table nat-translation NATtranslationtable ... - Displays a list of available commands beginning with certain letter(s), for example: (config)# voip-network d? dns Enter voip-network dns - Displays syntax help for a specific command by entering the command, a space, and then a question mark (?). This includes the range of valid values and a brief description of the next parameter expected for that particular command. For example: (config)# voip-network dns srv2ip ? [0-9] index If a command can be invoked (i.e., all its arguments have been entered), the question mark at its end displays "<cr>" to indicate that a carriage return (Enter) can now be entered to run the command, for example: (config)# logging host 10.1.1.1 ? <cr>
<Ctrl + A>	Moves the cursor to the beginning of the command line.

Shortcut Key	Description
<Ctrl + E>	Moves the cursor to the end of the command line.
<Ctrl + U>	Deletes all the characters on the command line.
auto finish	You need only enter enough letters to identify a command as unique. For example, entering "int G 0/0" at the configuration prompt provides you access to the configuration parameters for the specified Gigabit-Ethernet interface. Entering "interface GigabitEthernet 0/0" would work as well, but is not necessary.
Space Bar at the --More--prompt	Displays the next screen of output. You can configure the size of the displayed output, as described in "Configuring Displayed Output Lines in CLI Terminal Window" on page 91.

7.1.3 Common CLI Commands

The following table contains descriptions of common CLI commands.

Table 7-2: 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.
 <filter>	Applied to a command output. The filter should be typed after the command with a pipe mark ().

Command	Description
	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: # show system version grep Number ;Serial Number: 2239835;Slot Number: 1

7.1.4 Configuring Tables through 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. However, you can specify a different index number.

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

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

When you add a new table row, the device accesses the row's configuration mode. For example, if three rows are configured in the Account table (account-0, account-1, and account-2) and you then add a new row, 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 incremented to the next consecutive index number, as well as all the index rows listed below it 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 you then add a new row 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: The insert table row feature is applicable only to tables that do not have "child" tables (sub-tables).

You can also change the position (index) of a configured row by moving it one row up or one row down in the table, using the following command:

```
# <table> <index to move> move-up|move-down
```

For example, to move the row at Index 1 down to Index 2 in the IP-to-IP Routing table:

```
<config-voip># sbc routing ip2ip-routing 1 move-down
```


In this example, the previous row at Index 2 is moved up to Index 1.



Note: Changing of row position is applicable only to certain tables.

7.1.5 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.

7.2 Enabling CLI

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

7.2.1 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 58).

➤ 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 [minutes]	5
Maximum Telnet Sessions	5

2. From the 'Embedded Telnet Server' drop-down list, select **Enable Unsecured** or **Enable Secured** (i.e, SSL).
3. In the 'Telnet Server TCP Port' field, enter the port number for the embedded Telnet server.
4. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

For a detailed description of the Telnet parameters, see "Telnet Parameters" on page 724.

7.2.2 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 7-1: 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	AAAAB3NzaC1yc2EAAAABJQAAAIBh4d5kzHckyRmI3lawnb2b/Cscd/d2FDolycGWb/qjOYsLl889hCfu/GE5ADW8uf6FPxmM7ehhUcMptE2NzqZnz7S2/mE3xEW/ch451nre/n77J3kUcJH2GdoT8XG2UgP82agwYOpnu4wJukD6K+m1ALMdMQNEVH5k6cww==
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 761.
 - 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.

7.3 Configuring Maximum Telnet/SSH Sessions

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



Note: Before changing the 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**.

7.4 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 on page 37.
- **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 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 67.

- **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
```

7.5 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.

7.6 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 90).



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

7.7 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.

8 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 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 596.
- For more information on using the EMS tool, refer to the *EMS User's Manual* and *EMS Server IOM Manual*.

8.1 Disabling SNMP

By default, SNMP is enabled. You can change the setting, as described in the following procedure.

➤ To disable SNMP:

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

Figure 8-1: Disabling SNMP

2. From the 'Disable SNMP' drop-down list (DisableSNMP parameter), select **Yes**.
3. Click **Submit**, and then reset the device with a save-to-flash for your settings to take effect.

8.2 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. The SNMP community string determines the access privileges (read-only or read-write) of SNMP clients to the device's SNMP.



Note: SNMP community strings are used only for SNMPv1 and SNMPv2c; SNMPv3 uses username-password authentication along with an encryption key (see "Configuring SNMP V3 Users" on page 98).

For detailed descriptions of the SNMP parameters, see "SNMP Parameters" on page 725.

➤ **To configure SNMP community strings:**

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

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

⚡ Disable SNMP	No
Trap Community String	trapuser
Trap Manager Host Name	
Activity Trap	Disable

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 8-1: 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 (_)

Parameter	Description
	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".

8.3 Configuring SNMP Trap Destinations

The SNMP Trap Destinations table lets 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 table (**Configuration** tab > **System** menu > **Management** > **SNMP** > **SNMP Trap Destinations**).

Figure 8-2: SNMP Trap Destinations Table

		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**.



Notes:

- Only row entries whose corresponding check boxes are selected are applied when clicking **Submit**; otherwise, settings revert to their defaults.
- To enable the sending of the trap event `acPerformanceMonitoringThresholdCrossing`, which is sent every time a threshold (high or low) of a performance monitored SNMP object is crossed, configure the ini file parameter `PM_EnableThresholdAlarms` to 1.

Table 8-2: SNMP Trap Destinations Table Parameters Description

Parameter	Description
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
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.
Trap User	Associates a trap user with the trap destination. This determines the trap format, authentication level, and encryption level.

Parameter	Description
[SNMPManagerTrapUser]	- v2cParams (default) = SNMPv2 user community string - SNMPv3 user configured in "Configuring SNMP V3 Users" on page 98
Trap Enable [SNMPManagerTrapSendingEnable_x]	Activates the sending of traps to the SNMP Manager. [0] Disable [1] Enable (Default)

8.4 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 through the Web interface. You can also configure it through ini file (SNMPTrustedMgr_x) or CLI (configure system > snmp > trusted-managers).

➤ **To configure SNMP Trusted Managers:**

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

Figure 8-3: SNMP Trusted Managers Table

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.

8.5 Configuring SNMP V3 Users

The SNMPv3 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 through the Web interface. You can also configure it through ini file (SNMPUsers) or CLI (configure system > snmp v3-users).

➤ **To configure an SNMP v3 user:**

1. Open the SNMPv3 Users table (**Configuration** tab > **System** menu > **Management** > **SNMP** > **SNMP V3 Users**).
2. Click **Add**; the following dialog box appears:

Figure 8-4: SNMPv3 Users Table - Add Row Dialog Box

The dialog box titled "Add Row" contains the following fields and controls:

- Index:** A text input field containing the value "0".
- User Name:** A text input field.
- Authentication Protocol:** A dropdown menu with "None" selected.
- Privacy Protocol:** A dropdown menu with "None" selected.
- Authentication Key:** A text input field.
- Privacy Key:** A text input field.
- Group:** A dropdown menu with "Read-Write" selected.
- Buttons:** "Add" and "Cancel" buttons are located at the bottom right of the dialog.

3. Configure the SNMP V3 parameters according to the table below.
4. Click **Add**, 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 96), the configured trap destination becomes disabled and the trap user reverts to default (i.e., SNMPv2).

Table 8-3: SNMPv3 Users Table Parameters Description

Parameter	Description
Index [SNMPUsers_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
User Name username [SNMPUsers_Username]	Name of the SNMP v3 user. This name must be unique.
Authentication Protocol auth-protocol [SNMPUsers_AuthProtocol]	Authentication protocol of the SNMP v3 user. ▪ [0] None (default) ▪ [1] MD5 ▪ [2] SHA-1

Parameter	Description
Privacy Protocol priv-protocol [SNMPUsers_PrivProtocol]	Privacy protocol of the SNMP v3 user. ▪ [0] None (default) ▪ [1] DES ▪ [2] 3DES ▪ [3] AES-128 ▪ [4] AES-192 ▪ [5] AES-256
Authentication Key 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 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 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.

9 INI File-Based Management

The device can be configured through 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 719.
- To restore the device to default settings through the *ini* file, see "Restoring Factory Defaults" on page 627.

9.1 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 101
- Table parameters - see "Configuring Table ini File Parameters" on page 101

9.1.1 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 103.

9.1.2 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 103.

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_CoderSpecific;
CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0, 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.

9.1.3 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.

9.2 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 578).
 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 104.



Notes:

- If you save an ini file from the device and a table row is configured with invalid values, the ini file displays the row prefixed with an exclamation mark (!), for example:

```
!CpMediaRealm 1 = "ITSP", "Voice", "", 60210, 2, 6030, 0, "",
"";
```

- To restore the device to default settings through the *ini* file, see "Restoring Factory Defaults" on page 627.

9.3 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 583): 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 607): 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*.

9.4 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).

9.5 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>`).

9.6 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 <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

10 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 125.
- Only **Base64 (PEM)** encoded X.509 certificates can be loaded to the device.

10.1 Configuring TLS Certificate Contexts

The TLS Contexts table lets you configure up to 400 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 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 360.
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 342.
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 123.

The following procedure describes how to configure a TLS Context through the Web interface. You can also configure it through ini file (TLSContexts) or CLI (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 10-1: TLS Contexts Table - Add Record Dialog Box

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

Table 10-1: TLS Context Parameter Descriptions

Parameter	Description
Index tls <ID> [TLSContexts_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [TLSContexts_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 31 characters.
TLS Version tls-version [TLSContexts_TLSVersion]	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. ▪ [1] TLSv1.0 = Only TLS 1.0. ▪ [2] TLSv1.1 = Only TLS 1.1. ▪ [3] TLSv1.0 and TLSv1.1 = Only TLS 1.0 and TLS 1.1. ▪ [4] TLSv1.2 = Only TLS 1.2. ▪ [5] TLSv1.0 and TLSv1.2 = Only TLS 1.0 and TLS 1.2. ▪ [6] TLSv1.1 and TLSv1.2 = Only TLS 1.1 and TLS 1.2. ▪ [7] TLSv1.0 TLSv1.1 and TLSv1.2 = Only TLS 1.0, TLS 1.1 and TLS 1.2 (excludes SSL 3.0).

Parameter	Description
DTLS Version [TLSContexts_DTLSVersion]	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.
Cipher Server 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
Cipher Client 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.
OCSP Server ocsp-server [TLSContexts_OcspEnable]	Enables or disables certificate checking using OCSP. [0] Disable (default) [1] Enable
Primary OCSP Server ocsp-server-primary [TLSContexts_OcspServerPrimary]	Defines the IP address (in dotted-decimal notation) of the primary OCSP server. The default IP address is 0.0.0.0.
Secondary OCSP Server ocsp-server-secondary [TLSContexts_OcspServerSecondary]	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
OCSP Port ocsp-port [TLSContexts_OcspServerPort]	Defines the OCSP server's TCP port number. The default port number is 2560.
OCSP Default Response ocsp-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

10.2 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 table, select the required TLS Context index row, and then click the **TLS Context Certificate**  button, located below the table; 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 10-2: 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-----
MIIBtjCCAR8CAQAwZjESMBAGA1UEAxMJYXVkaW8uY29tMRUwEwYDVQQLZwZlZWVkaXVhcnRlcnMxejAQBgNVBAoTCUNvbnBvcnF0ZTEVMBMGA1UEBxMMUG91Z2hrZWVw
c2llMREwDwYDVQZIEWhOZXRcZG9yYzELMAkGA1UEBhMCVVMwZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBAPHpf2t4OLy3FRk5Bw7F12FWCXQ7nvuocHtu7Nns071M
xL7Of8YoL63eeIK2eDo8nm6rJO677z/AHWJmF65pAK1CbOIPgOZNS0g6+5JAmJAA
1LNUnOqjEsK7CF32uvolH//gFkhy5zleNvObI+25Pn38aJzEXc8DkGwZ19rROqRZ
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQDihdqbc1zkHdLFr+5BRuScKyGUXBM6
q7FGjFXAfZk1MmgnBMc/MYfSGTbawrQF7p6dNJ60DivmuCPf6Gzz5m2uqC6LqoIi
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-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFAFAD/MQswCQYDVQQGEw
JGUuZjETMBEGA1UEChMKQ2VydG9wZ3N0ZTEbMBkGA1UEAxMSQ2VydG9wZ3N0ZSB
ZXJ2ZXVyb4XDtk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1
UEBhMCVVMwZ8wDQYJKoZIhvcNAQEBBQADgGEOADCCAQkCggEAPqd4Mz
iR4spWldGRx8bQrhZkonWnNm`+Yhb7+4Q67ecf1janH7GcN/SXsfX7jJpreWUL
f7v7Cvpr4R7qIJcmdHIntmf7JPM5n6cDBv17uSW63er7NkVnMFHwK1QaGFLMyb
FkzaeGrvFm4k3lRefiXDmuOe+FhJgHYezYHf44LvPRPwhSrzi9+Aq3o8pWDguJ
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 10-3: Private key "OK" in Certificate Information Group

▼ Certificate Information	
Certificate subject:	/CN=ACL_5967925
Certificate issuer:	/CN=ACL_5967925
Time to expiration:	7246 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.

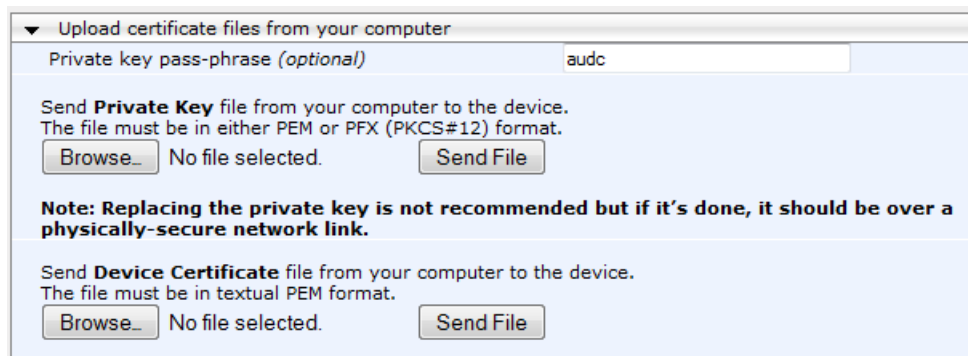
10.3 Assigning Externally Created Private Keys to TLS Contexts

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 table, select the required TLS Context index row, and then click the **TLS Context Certificate**  button, located below the table; the Context Certificates page appears.
4. Scroll down to the **Upload certificate files from your computer** group.

Figure 10-4: 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.

No file selected.

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.

No file selected.

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.

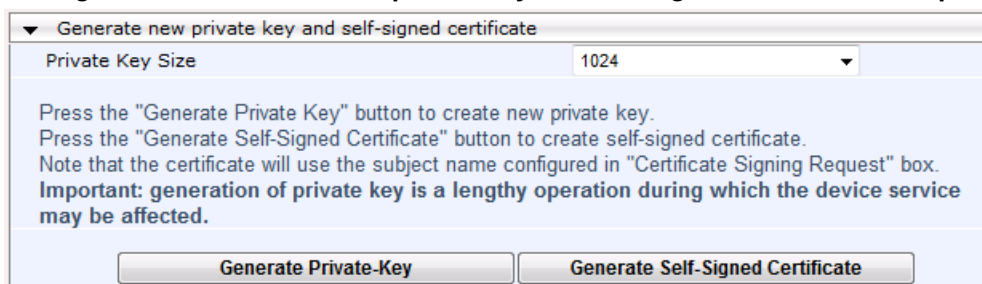
10.4 Generating Private Keys for TLS Contexts

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

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

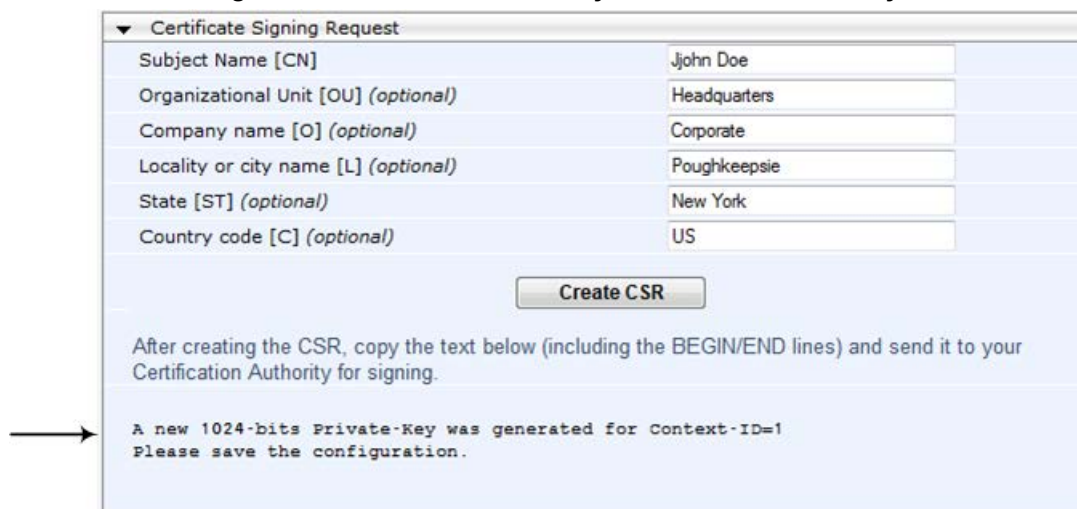
1. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
2. In the table, select the required TLS Context index row, and then click the **Context Certificates**  button, located below the table; the Context Certificates page appears.
3. Scroll down to the **Generate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

Figure 10-5: Generate new private key and self-signed certificate Group



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 10-6: Indication of Newly Generated Private Key



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.

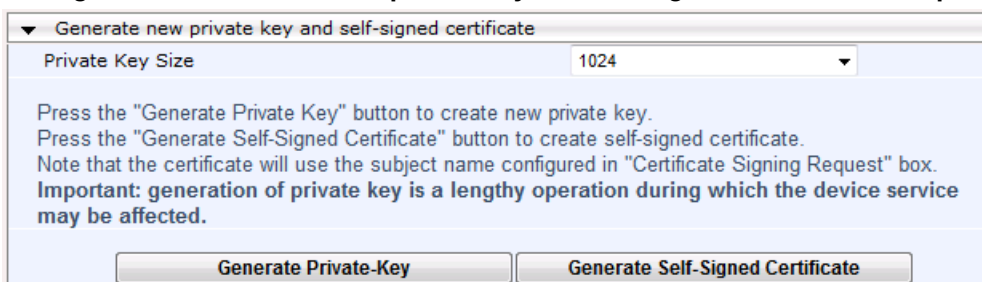
10.5 Creating Self-Signed Certificates for TLS Contexts

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 table, select the required TLS Context index row, and then click the **TLS Context Certificate**  button, located below the table; 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 10-7: Generate new private key and self-signed certificate Group



▼ Generate new private key and self-signed certificate

Private Key Size: 1024

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.

Generate Private-Key Generate Self-Signed Certificate

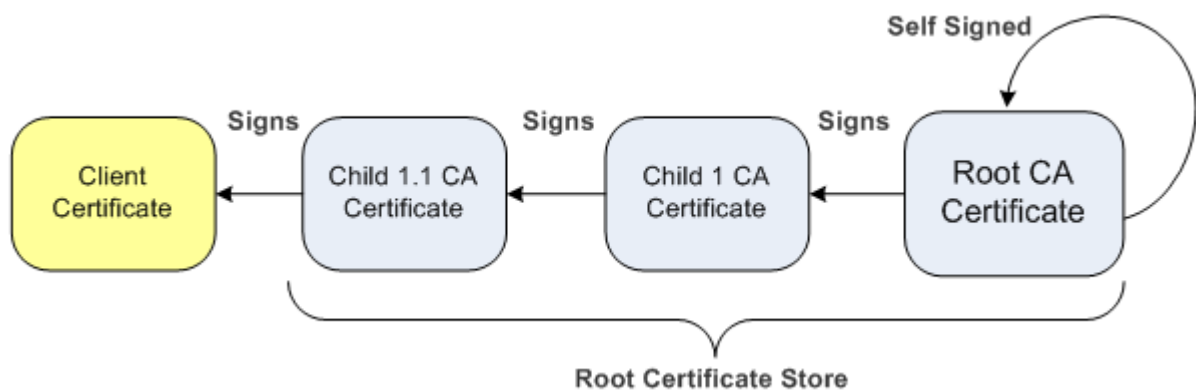
6. Click **Generate Self-Signed Certificate**; a message appears (after a few seconds) displaying the new subject name.
7. Save the configuration with a device reset for the new certificate to take effect.

10.6 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 10-8: 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.



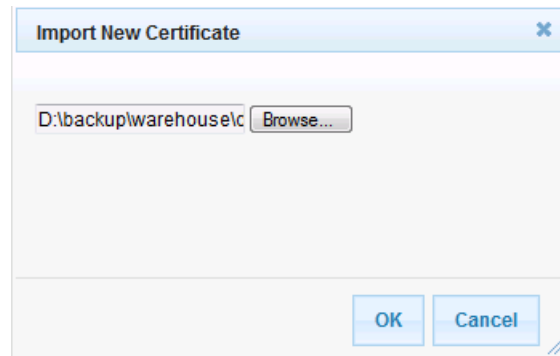
Note: Only Base64 (PEM) encoded X.509 certificates can be loaded to the device.

➤ **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 table, select the required TLS Context index row, and then click the **TLS Context Trusted Root Certificates**  button, located below the table; the Trusted Certificates page appears.

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

Figure 10-9: 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**.

10.7 Configuring Mutual TLS Authentication

This section describes how to configure mutual (two-way) TLS authentication.

10.7.1 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 (see "Configuring SIP Interfaces" on page 342), configure the 'TLS Mutual Authentication' parameter to **Enable** for the specific SIP Interface.
 - b. 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 119).
 - Make sure that the TLS certificate is signed by a CA that the SIP client trusts so that the client can authenticate the device.

10.7.2 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. On the Web Security Settings page (see "Configuring Web Security Settings" on page 76), configure the 'Secured Web Connection (HTTPS)' field to **HTTPS Only**. The setting ensures that you have a method for accessing the device in case the client certificate doesn't work. Restore the previous setting after testing the configuration.
2. In the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 109), select the required TLS Context row, and then click the **TLS Context Trusted Root Certificates**  button, located below the table; the Trusted Certificates page appears.
3. Click the **Import** button, and then select the certificate file.
4. Wait until the import operation finishes successfully.
5. On the Web Security Settings page, configure the 'Requires Client Certificates for HTTPS connection' field to **Enable**.
6. Reset the device with a burn-to-flash for your settings to take effect.

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 109).

10.8 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. The trap 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 10-10: 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 marked blank.

11 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.

11.1 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 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 742.

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

1. Open the Time and Date page (**Configuration** tab > **System** menu > **Time And Date**), and then scroll down to the 'NTP Sever' group:

Figure 11-1: NTP Parameters on Time and Date Page

▼ NTP Server		
Primary NTP Server Address (IP or FQDN)	0.0.0.0	
Secondary NTP Server Address (IP or FQDN)		
NTP Update Interval	Hours: 24	Minutes: 0

2. Configure the NTP server address:
 - In the 'Primary NTP Server Address' (NTPServerIP) field, configure the primary NTP server's address (IP or FQDN).
 - In the 'Secondary NTP Server Address' (NTPSecondaryServerIP) field, configure the secondary NTP server.
3. In the 'NTP Updated Interval' (NTPUpdateInterval) field, configure the period after which the date and time of the device is updated.
4. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**), and then scroll down to the 'NTP Settings' group:

Figure 11-2: NTP Authentication Parameters on Application Settings Page

▼ NTP Settings	
NTP Authentication Key Identifier	0
NTP Authentication Secret Key	

5. 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.
6. Verify that the device has received the correct date and time from the NTP server. The date and time is displayed in the 'UTC Time' read-only field on the Time and Date page.



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.

11.2 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 "Configuring Automatic Date and Time using SNTP" on page 125).

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

1. Open the Time and Date page (**Configuration** tab > **System** menu > **Time And Date**), and then scroll down to the 'Local Time' group:

Figure 11-3: Manually Configured Date and Time on Time and Date Page

Local Time						
	Year	Month	Day	Hour	Minutes	Seconds
Local Time	2015	3	27	4	57	45

2. In the 'Local Time' fields, enter the current date and time of the geographical location in which the device is installed:
 - Date:
 - ◆ 'Year' in yyyy format (e.g., "2015")
 - ◆ 'Month' in mm format (e.g., "3" for March)
 - ◆ 'Day' in dd format (e.g., "27")
 - Time:
 - ◆ 'Hour' in 24-hour format (e.g., "4" for 4 am)
 - ◆ 'Minutes' in mm format (e.g., "57")
 - ◆ 'Seconds' in ss format (e.g., "45")
3. Click **Submit**; the date and time is displayed in the 'UTC Time' read-only field.



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 date and time received from the NTP server.
- After performing a hardware reset, the date and time are returned to default values and thus, you should subsequently update the date and time.

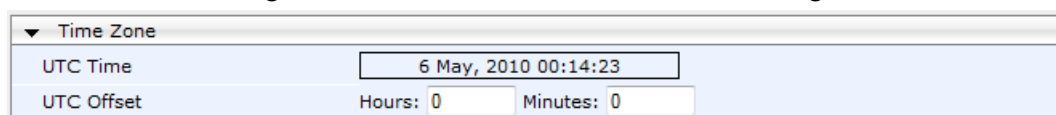
11.3 Configuring the Time Zone

You can configure the time zone in which the device is deployed. This is referred to as the Coordinated Universal Time (UTC) time offset and defines how many hours the device is from Greenwich Mean Time (GMT). For example, Germany Berlin is one hour ahead of GMT (UTC/GMT is +1 hour) and therefore, you would configure the offset to "1". USA New York is five hours behind GMT (UTC/GMT offset is -5 hours) and therefore, the offset is a minus value and configured as "-5".

➤ **To configure the time zone (UTC offset):**

1. Open the Time and Date page (**Configuration** tab > **System** menu > **Time And Date**), and then scroll down to the 'Time Zone' group:

Figure 11-4: UTC Offset on Time and Date Page



The screenshot shows a web interface for configuring the time zone. It features a section titled "Time Zone" with a dropdown arrow. Below this, there are two rows of configuration fields. The first row, labeled "UTC Time", contains a text box displaying "6 May, 2010 00:14:23". The second row, labeled "UTC Offset", contains two sub-fields: "Hours: 0" and "Minutes: 0".

Time Zone	
UTC Time	6 May, 2010 00:14:23
UTC Offset	Hours: 0 Minutes: 0

2. In the 'UTC Offset' fields (NTPServerUTCOffset), configure the time offset in relation to the UTC. For example, if your region is GMT +1 (an hour ahead), enter "1" in the 'Hours' field.
3. Click **Submit**; the updated time is displayed in the 'UTC Time' read-only field and the 'Local Time' fields.

11.4 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 Time and Date page (**Configuration** tab > **System** menu > **Time And Date**), and then scroll down to the 'Time Zone' group:

Figure 11-5: Configuring DST

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

2. From the 'Daylight Saving Time' (DayLightSavingTimeEnable) drop-down list, select **Enable**.
3. 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.
4. In the 'Offset' (DayLightSavingTimeOffset) field, configure the DST offset in minutes.
5. If the current date falls within the DST period, verify that it has been successful applied to the device's current date and time. You can view the device's date and time in the 'UTC Time' read-only field.

Part IV

General VoIP Configuration

12 Network

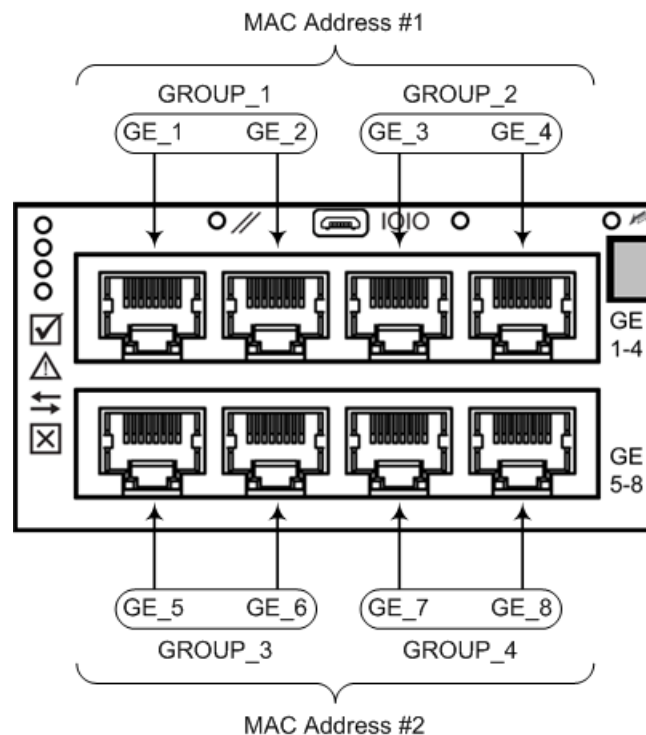
This section describes the network-related configuration.

12.1 Configuring Physical Ethernet Ports

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

The table also displays the status of the port (e.g., active) as well as the port group (*Ethernet Group*) to which the port belongs. You can assign up to two ports to an Ethernet Group. Ethernet Groups with two ports are used for 1+1 Ethernet port redundancy. For more information on Ethernet Groups and for assigning ports to Ethernet Groups, see "Configuring Ethernet Port Groups" on page 133.

The device's management tools (e.g., Web interface) use hard-coded strings to represent the physical ports, as shown below:



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		
2	GE_2	00:1e:67:11:7c:28	100Mbps	
FULL	DOWN			



Note: The device provides two MAC addresses for the LAN ports, which are allocated as follows: a MAC address for ports GE_1 through GE_4, and another MAC address for ports GE_5 through GE_8.

The following procedure describes how to configure the Ethernet ports through the Web interface. You can also configure it through ini file (PhysicalPortsTable) or CLI (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 clicking **Edit**; the following dialog box appears:

Edit Row

Index

0

Port

GE_1

Mode

Enable

Speed&Duplex

Auto Negotiation

Description

User Port #0

Group Member

GROUP_1

Group Status

Active

Save

Cancel

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 12-1: Physical Port Settings Table Parameter Descriptions

Parameter	Description
Port port [PhysicalPortsTable_Port]	(Read-only) Displays the Ethernet port number. The figure in the beginning of this section shows the mapping of this GUI port number to the physical port on the chassis.
Mode mode [PhysicalPortsTable_Mode]	(Read-only) Displays the mode of the port: ▪ [0] Disable ▪ [1] Enable (default)

Parameter	Description
Speed & Duplex 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 port-description [PhysicalPortsTable_PortDescription]	Defines an arbitrary description of the port. By default, the value is "User Port #<row index>".
Group Member group-member [PhysicalPortsTable_GroupMember]	(Read-only) Displays the Ethernet Group to which the port belongs. To assign the port to a different Ethernet Group, see "Configuring Ethernet Port Groups" on page 133.
Group Status 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, both ports show "Active". ▪ "Redundant": Standby (redundant) port.

12.2 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. In such a configuration, one port can be active while the other in standby mode or both ports can be active, depending on the ports' transmit (Tx) and receive (Rx) settings. 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 either 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 eight Ethernet Groups, each containing one port, or up to four Ethernet Groups, each containing two ports. By default, each Ethernet Group is assigned two ports; the other Ethernet Groups are empty.

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 135). Then by assigning the Ethernet Device to the IP network interface in the Interface table (see "Configuring IP Network

Interfaces" on page 138). 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 port names (strings) displayed in the Ethernet Group Settings table represent the physical ports on the device. For the mapping of these strings to the physical ports, see Configuring Physical Ethernet Ports on page 131.

The following procedure describes how to configure Ethernet Groups through the Web interface. You can also configure it through ini file (EtherGroupTable) or CLI (configure voip > ether-group).



Notes:

- 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.
- Two different MAC addresses are assigned to the Ethernet ports: one to ports GE 1-4 (upper ports) and another to ports GE 5-8 (lower ports).
- Ports with the same MAC address (e.g., GE 1-4 ports) must each be connected to a different Layer-2 switch.
- When implementing 1+1 Ethernet port redundancy, each port in the Ethernet Group (port pair) must be connected to a different switch, but in the same subnet.

➤ **To configure Ethernet Groups:**

1. Open the Ethernet Group Settings table (**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:

Edit Row
✕

Index

Group

Mode

Member 1

Member 2

- b. Set the 'Member 1' or 'Member 2' field (depending on where the port appears) to **None** (or assign it 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 12-2: Ethernet Group Settings Parameter Descriptions

Parameter	Description
Group group [EtherGroupTable_Group]	(Read-only) Displays the Ethernet Group number.
Mode mode [EtherGroupTable_Mode]	Defines the mode of operation of the ports in the Ethernet Group. This applies only to Ethernet Groups containing two ports. ▪ [2] 1RX/1TX = (Default) At any given time, only a single port in the Ethernet Group can transmit and receive packets. If a link exists on both ports, then the active one is either the first to have a link up or the lower-numbered port if both have the same link up from start. ▪ [3] 2RX/1TX = Both ports in the 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. This option is applicable only to the Maintenance interface for High Availability (HA) deployments. For more information, see Network Topology Types and Rx/Tx Ethernet Port Group Settings on page 559. ▪ [5] Single = 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. 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. ▪ For Ethernet Group settings for the Maintenance interface when implementing High Availability, see Initial HA Configuration on page 559.
Member 1 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 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.

12.3 Configuring Underlying Ethernet Devices

The Ethernet Device table lets you configure up to 48 *Ethernet Devices*. An Ethernet Device represents a Layer-2 bridging device and is assigned a VLAN ID and an Ethernet Group. Multiple Ethernet Devices can be associated with the same Ethernet Group.

Once configured, you need to assign the Ethernet Device to 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). You can assign the same Ethernet Device to multiple IP network

interfaces, thereby implementing multi-homing (multiple addresses on the same interface/VLAN).

Each Ethernet Device (VLAN) can be configured with a VLAN tagging policy, which determines whether the Ethernet Device accepts tagged or untagged packets received on the Ethernet port associated with the Ethernet Device.

By default, the device provides a pre-configured Ethernet Device at Index 0 with the following settings:

- Name: "vlan 1"
- VLAN ID: 1
- Ethernet Group: GROUP 1
- Tagging Policy: Untagged

The pre-configured Ethernet Device is associated with the default IP network interface (OAMP) in the Interface table. The Untagged policy of the pre-configured Ethernet Device enables you to connect to the device using the default OAMP interface.

You can view configured Ethernet Devices that have been successfully applied to the device (saved to flash), in the Ethernet Device Status table. This page is accessed by clicking the **Ethernet Device Status Table** button, located at the bottom of the Ethernet Device table. The Ethernet Device Status table can also be accessed from the **Status & Diagnostics** tab > **VoIP Status** menu > **Ethernet Device Status Table** (see "Viewing Ethernet Device Status" on page 643).



Note: You cannot delete an Ethernet Device that is associated with an IP network interface (in the Interface table). You can only delete it once you have disassociated it from the IP network interface.

The following procedure describes how to configure Ethernet devices through the Web interface. You can also configure it through ini file (DeviceTable) or CLI (config-voip > interface network-dev).

➤ **To configure an Ethernet Device:**

1. Open the Ethernet Device table (**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 **Add**.

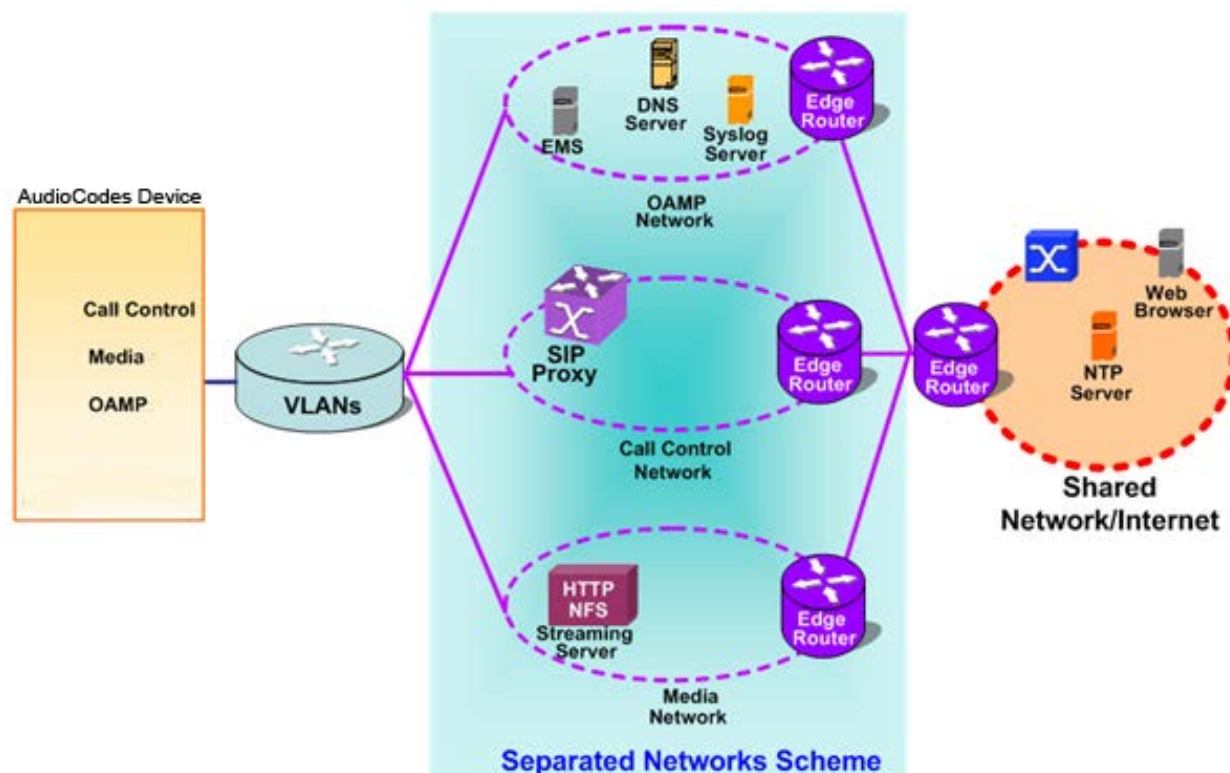
Table 12-3: Ethernet Device Table Parameter Descriptions

Parameter	Description
Index [DeviceTable_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
VLAN ID vlan-id [DeviceTable_VlanID]	Defines a VLAN ID for the Ethernet Device. The valid value is 1 to 3999. The default value is 1. Note: Each Ethernet Group must have a unique VLAN ID.
Underlying Interface underlying-if [DeviceTable_UnderlyingInterface]	Assigns an Ethernet Group to the Ethernet Device. For configuring Ethernet Groups, see Configuring Ethernet Port Groups on page 133. Note: The parameter is mandatory.
Name name [DeviceTable_DeviceName]	Defines a name for the Ethernet Device. This name is used to associate the Ethernet Device with an IP network interface in the Interface table ('Underlying Device' field - see "Configuring IP Network Interfaces" on page 138) and/or with a static route in the Static Route table ('Device Name' field - see "Configuring Static IP Routing" on page 146).
Tagging tagging [DeviceTable_Tagging]	Defines VLAN tagging per Ethernet Device. ▪ [0] Untagged = (Default of pre-configured Ethernet Device) The Ethernet Device accepts untagged packets as well as packets with the same VLAN ID as configured for the Ethernet Device. Incoming untagged packets are assigned the VLAN ID of the Ethernet Device. The Ethernet Device sends these VLAN packets untagged (i.e., removes the VLAN ID). ▪ [1] Tagged = (Default for new Ethernet Devices) The Ethernet Device accepts packets that have the same VLAN ID as configured for the Ethernet Device and sends packets with this VLAN ID. For all Ethernet Devices that are associated with the same Ethernet Group ('Underlying Interface') and set to Tagged, incoming untagged packets received on this Ethernet Group are discarded. Note: Only one Ethernet Device can be configured as Untagged per associated Ethernet Group (port group). In other words, if multiple Ethernet Devices are associated with the same Ethernet Group, only one of these Ethernet Devices can be set to Untagged ; all the others must be set to Tagged .

12.4 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 12-1: Multiple Network Interfaces



The device is shipped with a default OAMP interface. For more information, see "Default OAMP IP Address" on page 33. 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 554.

- IP address (IPv4 and IPv6) and subnet mask (prefix length)
- For configuring Quality of Service (QoS), see "Configuring the QoS Settings" on page 149.
- 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 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 146.



Note: Before configuring IP interfaces, it is recommended that you read the IP interface configuration guidelines in "Interface Table Configuration Guidelines" on page 142.

The following procedure describes how to configure the IP network interfaces through the Web interface. You can also configure it through ini file (InterfaceTable) or CLI (configure voip/interface network-if).

➤ **To configure IP network interfaces:**

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

Index	Interface Name	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Primary DNS	Secondary DNS	Underlying Device
0	LAN	OAMP + Media	IPv4 Manual	10.15.7.96	16	10.15.0.1	0.0.0.0	0.0.0.0	VLAN 1
1	WAN	Media	IPv4 Manual	10.15.7.100	16	0.0.0.0	0.0.0.0	0.0.0.0	VLAN 2

Page 1 of 1 10 View 1 - 2 of 2

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



Notes:

- If you modify the OAMP interface's address, after clicking **Add** in the dialog box you will lose connectivity with the device and need to access the device with the new address.
- If you edit or delete an IP interface, current calls using the interface are immediately terminated.
- If you delete an IP interface, row indices of other tables (e.g., Media Realm table) that are associated with the deleted IP interface, lose their association with the interface ('Interface Name' field displays "None") and the row indices become invalid.
- When editing or deleting the Maintenance interface for HA mode, you must reset the device for your changes to take effect.

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 643.

Table 12-4: Interface Table Parameters Description

Parameter	Description
Table parameters	
Index network-if [InterfaceTable_Index]	Table index row of the interface. The range is 0 to 47.
Application Type 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.
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.
IP Address ip-address [InterfaceTable_IPAddress]	Defines the IPv4/IPv6 address, in dotted-decimal notation.
Prefix Length 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

Parameter	Description
length]	of bits which are set in the dotted-decimal format. For example, 192.168.0.0/16 is synonymous with 192.168.0.0 and subnet 255.255.0.0. This CIDR lists the number of '1' bits in the subnet mask (i.e., replaces the standard dotted-decimal representation of the subnet mask for IPv4 interfaces). For example, a subnet mask of 255.0.0.0 is represented by a prefix length of 8 (i.e., 11111111 00000000 00000000 00000000) and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (i.e., 11111111 11111111 11111111 11111100). The prefix length is a Classless Inter-Domain Routing (CIDR) style presentation of a dotted-decimal subnet notation. The CIDR-style presentation is the latest method for interpretation of IP addresses. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes. The prefix length for IPv4 must be set to a value from 0 to 30. The prefix length for IPv6 must be set to a value from 0 to 64.
Default Gateway 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.
Interface Name 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.
Primary DNS 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.
Secondary DNS 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 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 135.

12.4.1 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.

12.4.2 Multiple Interface Table Configuration Summary and Guidelines

The Interface table configuration must adhere to the following rules:

- Multiple Control and Media interfaces can be configured with overlapping IP addresses and subnets.
- The prefix length replaces the dotted-decimal subnet mask presentation and **must** have a value of 0-30 for IPv4 addresses and a value of 0-64 for IPv6 addresses.
- **One** OAMP interface must be configured and this **must** be an IPv4 address. This OAMP interface can be combined with Media and Control.
- At least one Control interface **must** be configured.
- At least one Media interface **must** be configured.
- Multiple Media and/or Control interfaces can be configured with an IPv6 address.
- The network interface types can be combined:
 - Example 1:
 - ◆ One combined OAMP-Media-Control interface with an IPv4 address
 - Example 2:
 - ◆ One OAMP interface with an IPv4 address
 - ◆ One or more Control interfaces with IPv4 addresses
 - ◆ One or more Media interfaces with IPv4 interfaces
 - Example 3:
 - ◆ One OAMP with an IPv4 address
 - ◆ One combined Media-Control interface with IPv4 address
 - ◆ One combined Media-Control interface with IPv6 address
- Each network interface can be configured with a Default Gateway. The address of the Default Gateway **must** be in the same subnet as the associated interface. Additional static routing rules can be configured in the Static Route table.
- The interface name **must** be configured (mandatory) and must be unique for each interface.
- For IPv4 addresses, the 'Interface Mode' column must be set to IPv4 Manual. For IPv6 addresses, this column must be set to IPv6 Manual or IPv6 Manual Prefix.



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

12.4.3 Networking Configuration Examples

This section provides configuration examples of networking interfaces.

12.4.3.1 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 12-5: 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 12-6: 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.

12.4.3.2 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 12-7: 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 12-8: 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.

12.4.3.3 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 12-9: 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	MediaCntrl1
2	Media & Control	IPv4 Manual	200.200.86.14	24	200.200.86.1	202	MediaCntrl2
3	Media & Control	IPv6 Manual	2000::1:200:200:86:14	64	::	202	V6CntrlMedia2

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

Table 12-10: Example of Static Route Table

Destination	Prefix Length	Gateway
176.85.49.0	24	192.168.0.10

3. The NTP application is configured (through 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

12.4.3.4 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 12-11: Configured Default Gateway Example

Index	Applica tion Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Underl ying 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 12-12: Separate Static Route Table Example

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

12.5 Configuring Static IP Routes

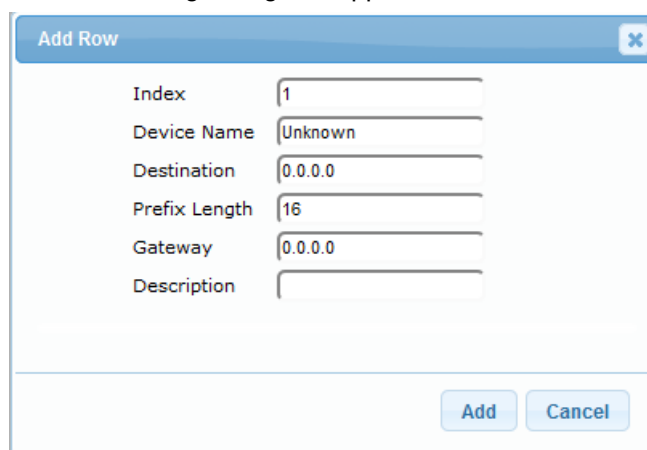
The Static Route table lets you configure up to 30 static IP routing rules. Using static routes lets you communicate with LAN networks that are not located behind the Default Gateway specified for the IP network interface, configured in the Interface table, from which the packets are sent. Before sending 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. 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 643).

The following procedure describes how to configure static routes through the Web interface. You can also configure it through ini file (StaticRouteTable) or CLI (configure voip > routing static).

➤ **To configure a static IP route:**

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



The dialog box titled "Add Row" contains the following fields:

- Index:** 1
- Device Name:** Unknown
- Destination:** 0.0.0.0
- Prefix Length:** 16
- Gateway:** 0.0.0.0
- Description:** (empty text box)

At the bottom right, there are two buttons: **Add** and **Cancel**.

3. Configure a static route according to the parameters described in the table below.
4. Click **Add**, 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.

Table 12-13: Static Route Table Parameter Descriptions

Parameter	Description
Index [StaticRouteTable_Index]	Defines an index number for the new table row. The valid value is 0 to 29. Note: Each row must be configured with a unique index.

Parameter	Description
Device Name 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 138). For configuring Ethernet Devices, see Configuring Underlying Ethernet Devices on page 135.
Destination 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 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 value must be 0 to 31 for IPv4 interfaces and a value of 0 to 64 for IPv6 interfaces.
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 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 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.

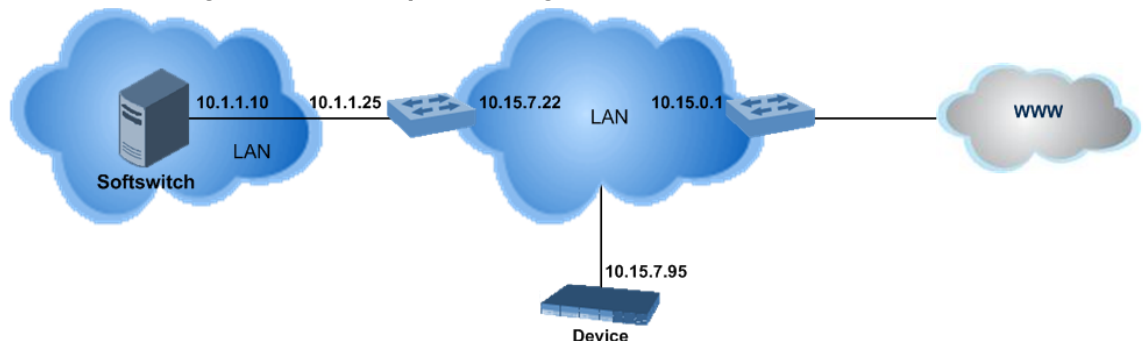
12.5.1 Configuration Example of Static IP Routes

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' fields, 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 12-2: Example of using a 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

12.5.2 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.

12.6 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 12-14: 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 141): - OAMP - Control	Depends on traffic type: - Control: Premium control - Management: Bronze

The following procedure describes how to configure DiffServ-to-VLAN priority mapping through the Web interface. You can also configure it through ini file (DiffServToVlanPriority) or CLI (configure voip > qos vlan-mapping).

➤ **To configure QoS:**

1. Open the Diff Serv table (**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 12-3: DiffServ Table Page - Add Row Dialog Box

The dialog box titled 'Add Row' has a close button (X) in the top right corner. It contains three labeled input fields: 'Index' with the value '10', 'Differentiated Services' with the value '0', and 'VLAN Priority' with the value '0'. At the bottom right, there are two buttons: 'Add' and 'Cancel'.

- b. Configure a DiffServ-to-VLAN priority mapping (Layer-2 QoS) according to the parameters described in the table below.
 - c. Click Add, and then save ("burn") your settings to flash memory.

Table 12-15: DiffServ Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Differentiated Services diff-serv [DiffServToVlanPriority_DiffServ]	Defines a DiffServ value. The valid value is 0 to 63.
VLAN Priority 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 12-4: QoS Settings Page - Differentiated Services

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

12.7 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 through the Web interface. You can also configure it through ini file - 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 12-5: 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 Received ICMP Redirect Messages' parameter.
- To enable or disable the sending of ICMP Destination Unreachable messages, use the 'Send ICMP Unreachable Messages' parameter.
- Click **Submit**.

12.8 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 152
- Internal SRV table - see "Configuring the Internal SRV Table" on page 153

12.8.1 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 138).

The following procedure describes how to configure the DNS table through the Web interface. You can also configure it through ini file (DNS2IP) or CLI (configure voip > voip-network dns dns-to-ip).

➤ **To configure the internal DNS table:**

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

Figure 12-6: Internal DNS Table - Add Row Dialog Box

Add Row	
Index	1
Name	
First IP Address	0.0.0.0
Second IP Address	0.0.0.0
Third IP Address	0.0.0.0
Fourth IP Address	0.0.0.0
Add Cancel	

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

Table 12-16: Internal DNS Table Parameter Description

Parameter	Description
Domain Name 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 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 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 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 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.

12.8.2 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 138).

The following procedure describes how to configure the Internal SRV table through the Web interface. You can also configure it through ini file (SRV2IP) or CLI (configure voip > voip-network dns srv2ip).

➤ To configure an SRV rule:

1. Open the Internal SRV table (**Configuration** tab > **VoIP** menu > **Network** > **DNS** > **Internal SRV Table**).

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

Figure 12-7: Internal SRV Table - Add Row Dialog Box

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

Table 12-17: Internal SRV Table Parameter Descriptions

Parameter	Description
Domain Name domain-name [Srv2lp_InternalDomain]	Defines the host name to be translated. The valid value is a string of up to 31 characters. By default, no value is defined.
Transport Type transport-type [Srv2lp_TransportType]	Defines the transport type. ▪ [0] UDP (default) ▪ [1] TCP ▪ [2] TLS
DNS Name (1-3) 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. By default, no value is defined.
Priority (1-3) priority-1 2 3 [Srv2lp_Priority1/2/3]	Defines the priority of the target host. A lower value means that it is more preferred. By default, no value is defined.
Weight (1-3) weight-1 2 3 [Srv2lp_Weight1/2/3]	Defines a relative weight for records with the same priority. By default, no value is defined.

Parameter	Description
Port (1-3) port - 1 2 3 [Srv2lp_Port1/2/3]	Defines the TCP or UDP port on which the service is to be found. By default, no value is defined.

12.9 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.

12.9.1 Device Located behind NAT

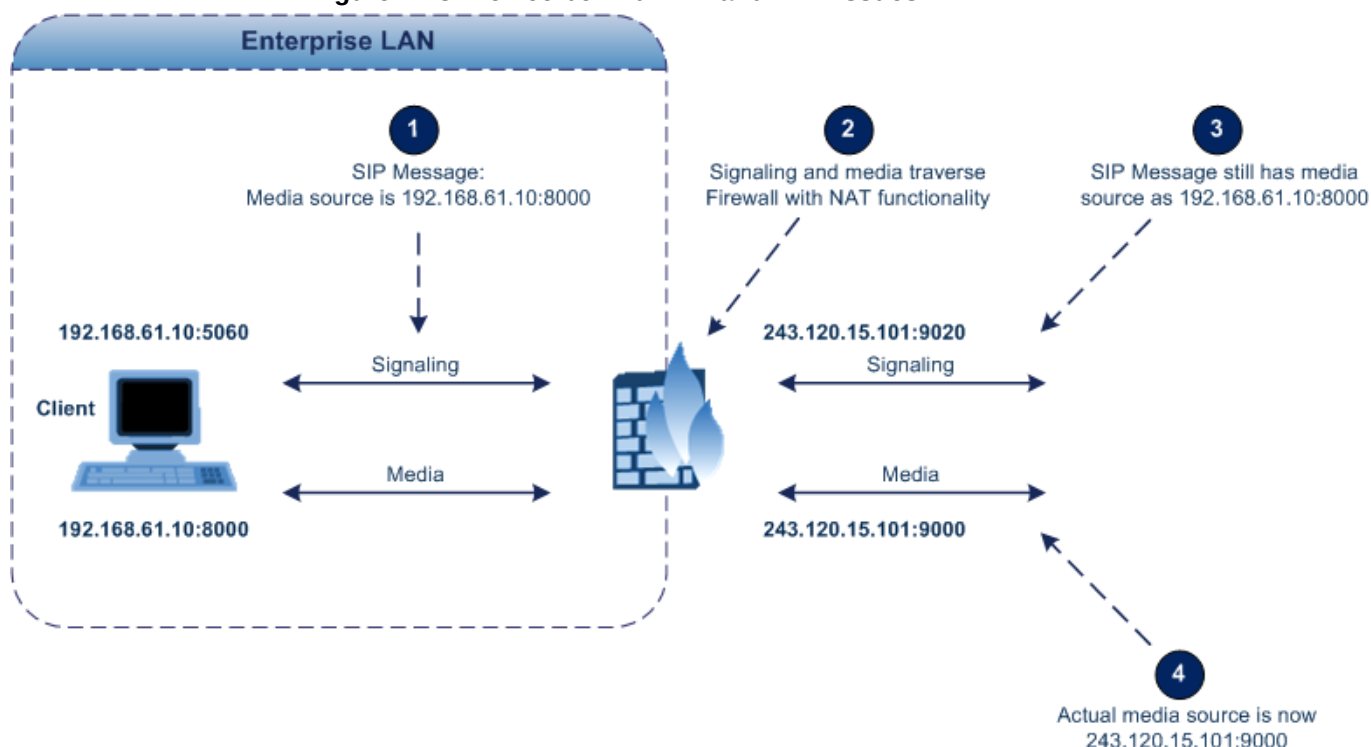
Two different streams traverse through NAT - signaling and media. A device located behind a NAT that initiates a signaling path has problems receiving incoming signaling responses as they are blocked by the NAT server. Therefore, the initiating device must inform the receiving device where to send the 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 157.
- b. If configured, uses the NAT Translation table which configures NAT per interface - see Configuring NAT Translation per IP Interface on page 158.

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 12-8: Device behind NAT and NAT Issues



12.9.1.1 Configuring a Static NAT IP Address for All Interfaces

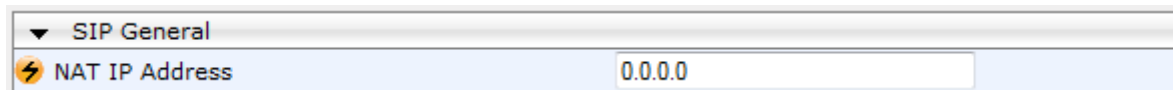
You can configure a global (public) IP address of the router to enable static NAT between the device and the Internet for all network interfaces. Thus, the device replaces the source IP address for media of all outgoing SIP messages sent on any of its network interfaces to this public IP address.

The following procedure describes how to configure a static NAT address through the Web interface. You can also configure it through ini file (StaticNATIP) or CLI (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 12-9: Configuring Static NAT IP Address in SIP General Parameters Page



The screenshot shows a web interface for configuring SIP parameters. A dropdown menu is open, showing 'SIP General' selected. Below it, the 'NAT IP Address' field is visible, containing 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.

12.9.1.2 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 from the "public" network. Each IP network interface, configured in the Interface table, can be associated with a NAT rule, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range).

The following procedure describes how to configure NAT translation rules through the Web interface. You can also configure it through ini file (NATTranslation) or CLI (voip-network nattranslation).

➤ **To configure NAT translation rules:**

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

Figure 12-10: NAT Translation Table - Add Row Dialog Box

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

Table 12-18: NAT Translation Table Parameter Descriptions

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

Parameter	Description
Source Interface src-interface-name [NATTranslation_SrcIPInterfaceName]	Assigns an IP network interface to the rule. Outgoing packets sent from the specified network interface are NAT'ed. By default, no value is defined (None). For configuring IP network interfaces, see "Configuring IP Network Interfaces" on page 138.
Target IP Address target-ip-address [NATTranslation_TargetIPAddress]	Defines the global (public) IP address. The device adds the address to the SIP Via header, Contact header, 'o=' SDP field, and 'c=' SDP field, in the outgoing packet.
Source Start Port src-start-port [NATTranslation_SourceStartPort]	Defines the optional starting port range (0-65535) of the IP interface, used as matching criteria for the 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 src-end-port [NATTranslation_SourceEndPort]	Defines the optional ending port range (0-65535) of the IP interface, used as matching criteria for the 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 target-start-port [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.
Target End Port target-end-port [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.

12.9.2 Remote UA behind NAT

12.9.2.1 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 is a local address, the device considers the endpoint as located behind NAT. In this case, the device sends the SIP messages to the endpoint, using the packet's source IP address. Otherwise (or if you have disabled the NAT Detection mechanism), the device sends the SIP messages according to the SIP standard RFC 3261, where requests within the SIP dialog are sent using the IP address in the Contact header, and responses to INVITES are sent using the IP address in the Via header. To enable or disable the device's NAT Detection mechanism, use the 'SIP NAT Detection' parameter.

If necessary, you can also configure the device to always consider incoming SIP INVITE messages as sent from endpoints that are located behind NAT. When this is enabled, the device sends responses to the INVITE (to the endpoint), using the the source IP address of the packet (INVITE) initially received from the endpoint. This is especially useful in

scenarios where the endpoint is located behind a NAT firewall and the device (for whatever reason) is unable to identify NAT using its regular NAT Detection mechanism. This feature is enabled per specific calls using IP Groups. To configure this feature, use the 'Always Use Source Address' parameter in the IP Group table (see "Configuring IP Groups" on page 348). If this feature is disabled, the device's NAT detection is according to the settings of the global parameter, 'SIP NAT Detection' parameter.

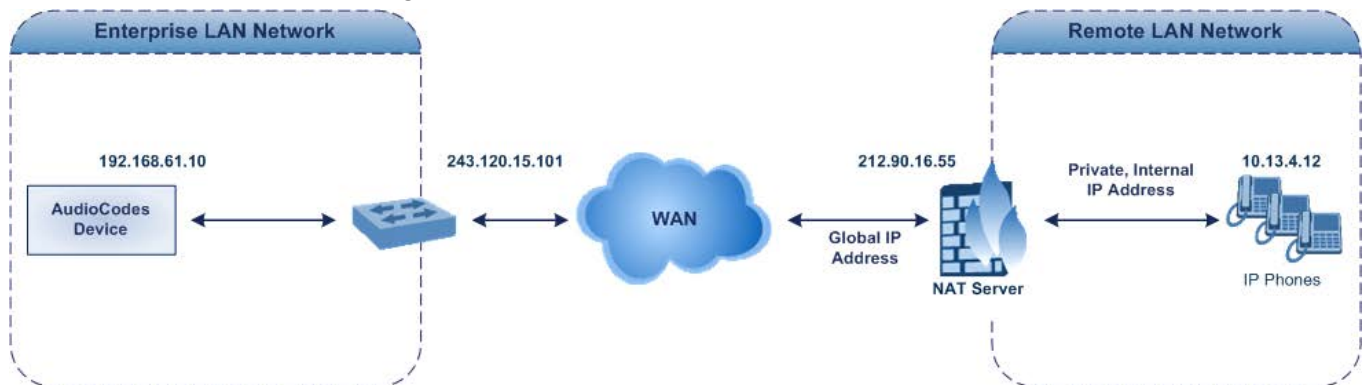
12.9.2.2 Media (RTP/RTCP/T.38)

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 160
- RTP No-Op packets according to the avt-rtp-noop draft - see "No-Op Packets" on page 161

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

Figure 12-11: Remote UA behind NAT



12.9.2.2.1 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:

- [0] Enable NAT Only if Necessary: NAT traversal is performed only if the UA is located behind NAT:
 - UA behind NAT: The device sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA.
 - UA not behind NAT: The device sends the packets to the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message.

Note: 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] Disable NAT: (Default) The device considers the UA as not located behind NAT and sends media packets to the UA using the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message.
- [2] Force NAT: The device always considers the UA as behind NAT and sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA. The device only sends packets to the UA after it receives the first packet from the UA (to obtain the IP address).
- [3] NAT by Signaling = The device identifies whether or not the UA is located behind NAT based on the SIP signaling. The device assumes that if signaling is behind NAT that the media is also behind NAT, and vice versa. If located behind NAT, the device sends media as described in option [2] Force NAT; if not behind NAT, the device sends media as described in option [1] Disable NAT. This option is applicable only to SBC calls. If the parameter is configured to this option, Gateway calls use option [0] Enable NAT Option, by default.

➤ **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 (NATMode).
3. Click **Submit**.

12.9.2.2.2 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.

- **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. 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:

- The No-OP Packet feature requires DSP resources.
- Receipt of No-Op packets is always supported.

12.9.2.2.3 Fax Transmission behind NAT

The device supports transmission from fax machines (connected to the device) located inside (behind) a Network Address Translation (NAT). Generally, the firewall blocks T.38 (and other) packets received from the WAN, unless the device behind the NAT sends at least one IP packet from the LAN to the WAN through the firewall. If the firewall blocks T.38 packets sent from the termination IP fax, the fax fails.

To overcome this, the device sends No-Op ("no-signal") packets to open a pinhole in the NAT for the answering fax machine. The originating fax does not wait for an answer, but immediately starts sending T.38 packets to the terminating fax machine upon receipt of a re-INVITE with T.38 only in the SDP, or T.38 and audio media in the SDP. This feature is configured using the T38FaxSessionImmediateStart parameter. The No-Op packets are enabled using the NoOpEnable and NoOpInterval parameters.

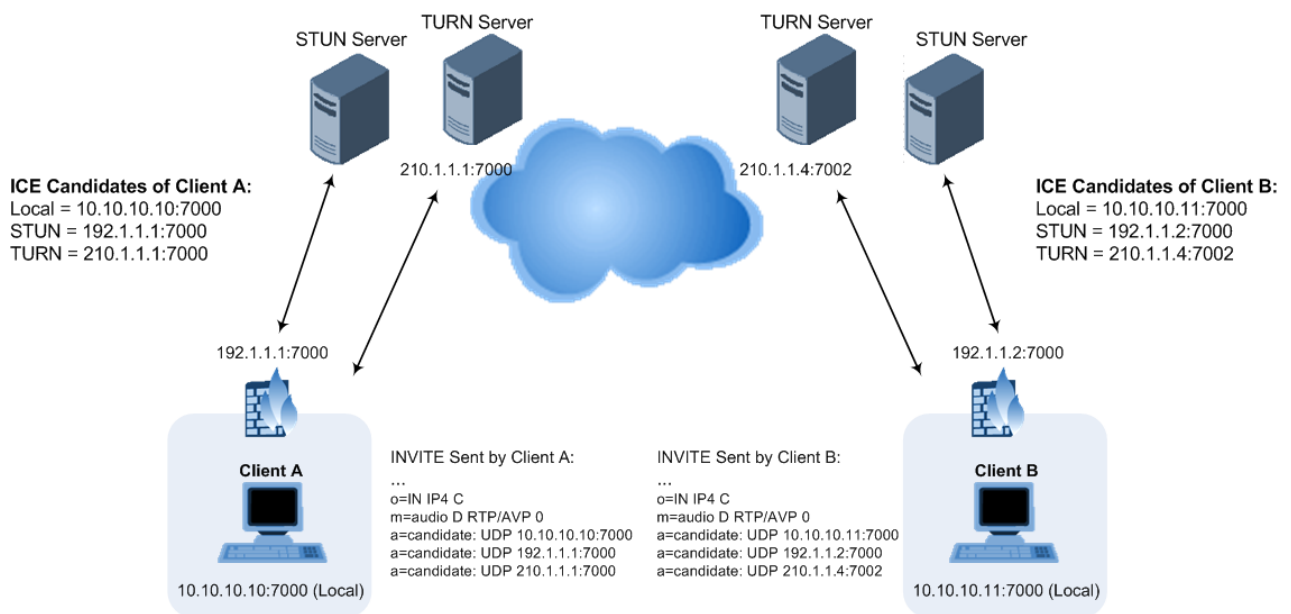
12.9.2.2.4 ICE Lite

The device supports Interactive Connectivity Establishment (ICE) Lite for SBC calls. ICE is a methodology for NAT traversal, enabling VoIP interoperability across networks to work better across NATs and firewalls. It employs Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) protocols to provide a peer with a public IP address and port that can be used to connect to a remote peer.

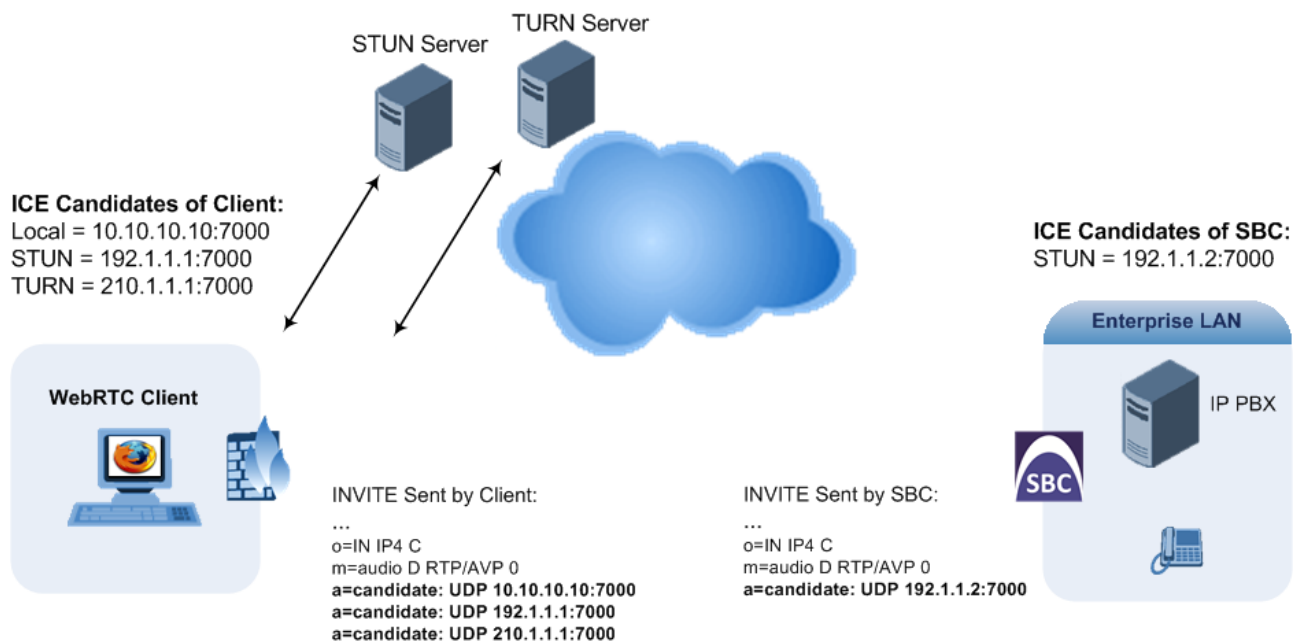
In order for clients behind NATs and/or firewalls to send media (RTP) between one another, they need to discover each others P address and port as seen by the "outside" world. If both peers are located in different private networks behind a NAT, the peers must coordinate to determine the best communication path between them.

ICE first tries to make a connection using the client's private local address. If that fails (which it will for clients behind NAT), ICE obtains an external (public) address using a STUN server. If that fails, traffic is routed through a TURN relay server (which has a public address).

These addresses:ports (local, STUN, TURN and any other network address) of the client are termed "candidates". Each client sends its' candidates to the other in the SDP body of the INVITE message. Peers then perform connectivity checks per candidate of the other peer, using STUN binding requests sent on the RTP and RTCP ports. ICE tries each candidate and selects the one that works (i.e., media can flow between the clients). The following figure shows a simple illustration of ICE:



The device's support for ICE-Lite means that it does not initiate the ICE process. Instead, it supports remote endpoints that initiate ICE to discover their workable public IP address with the device. Therefore, the device supports the receipt of STUN binding requests for connectivity checks of ICE candidates and responds to them with STUN responses. Note that in the response to the INVITE message received from the remote endpoint, the device sends only a single candidate for its' own IP address. This is the IP address of the device that the client uses. To support ICE, the SBC leg interfacing with the ICE-enabled client (SIP entity) must be enabled for ICE. This is done using the IP Profile parameter, `IPProfile_SBCIceMode` (see "Configuring IP Profiles" on page 391).



As the ICE technique has been defined by the WebRTC standard as mandatory for communication with the WebRTC client, ICE support by the device is important for deployments implementing WebRTC. For more information on WebRTC, see "WebRTC"

on page 523. Once a WebRTC session (WebSocket) is established for SIP signaling between the device and the WebRTC client, the client's IP address needs to be discovered by the SBC device using the ICE technique.

12.10 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 12-12: 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 754.

12.11 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.

13 Security

This section describes the VoIP security-related configuration.

13.1 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 79), you must configure a firewall rule that permits traffic from these IP addresses.
- Only users with Security Administrator or Master access levels can configure firewall rules.
- The device supports dynamic firewall pinholes for media (RTP/RTCP) traffic negotiated in the SDP offer-answer of SIP calls. The pinhole allows the device to ignore its firewall and accept the traffic on the negotiated port. The device automatically closes the pinhole once the call terminates. Therefore, it is unnecessary to configure specific firewall rules to allow traffic through specific ports. For example, if you have configured a firewall rule to block all media traffic in the port range 6000 to 7000 and a call is negotiated to use the local port 6010, the device automatically opens port 6010 to allow the call.
- 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 the 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 566.

The following procedure describes how to configure Firewall rules through the Web interface. You can also configure it through ini file (AccessList) or CLI (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 13-1: Firewall Settings Table - Add Row Dialog Box

Parameter	Value
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 **Add**, and then reset the device with a burn-to-flash for your settings to take effect.

Table 13-1: Firewall Settings Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Source IP 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. The default is 0.0.0.0.
Source Port 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. The default is 0. Note: When set to 0, this field is ignored and any source port matches the rule.

Parameter	Description
Prefix Length 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 the parameter to a value other than 0.
Start Port 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 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 (default). Note: When the protocol type isn't TCP or UDP, the entire range must be provided.
Protocol 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. The default is Any. Note: The parameter 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 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 network-interface-name [AccessList_Interface_x]	Defines the network interface to which you want to apply the rule. This is applicable if you enabled the 'Use Specific Interface' field. The list displays interface names as defined in the Interface table in "Configuring IP Network Interfaces" on page 138.

Parameter	Description
Packet Size 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 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 byte-burst [AccessList_Byte_Burst]	Defines the tolerance of traffic rate limit (number of bytes). The default is 0.
Action Upon Match 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 13-2: 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

Parameter	Firewall Rule				
	1	2	3	4	5
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.

13.2 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 109, where you need to configure certificates (TLS Context).



Note: 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.

➤ 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 360 and "Configuring SIP Interfaces" on page 342, 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**).

Figure 13-2: TLS Parameters on General Security Settings Page

▼ TLS Settings	
TLS Version	SSL 2.0-3.0 and TLS 1.0
Strict Certificate Extension Validation	Disable
Client Cipher String	ALL:!ADH
▼ SIP TLS Settings	
TLS Client Re-Handshake Interval	0
⚡ TLS Mutual Authentication	Disable
Peer Host Name Verification Mode	Disable
TLS Client Verify Server Certificate	Disable
TLS Remote Subject Name	

For a description of the TLS parameters, see "TLS Parameters" on page 759.

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**).

13.3 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 181.

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).

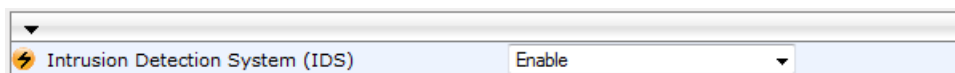
13.3.1 Enabling IDS

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 13-3: Enabling IDS on IDS Global Parameters Page



The screenshot shows a web interface with a dropdown menu labeled 'Intrusion Detection System (IDS)' and a value of 'Enable' selected.

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.

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

Figure 13-4: 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 13-5: IDS Policy Table - Add Row Dialog Box

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

Table 13-3: IDS Policy Table Parameter Descriptions

Parameter	Description
Index policy [IDSPolicy_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name rule [IDSPolicy_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters.
Description [IDSPolicy_Description]	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 13-6: 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 13-7: IDS Rule Table - Add Record

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

- Index: 5
- Reason: Connection abuse
- Threshold Scope: Global
- Threshold Window: 1
- Minor-Alarm Threshold: -1
- Major-Alarm Threshold: -1
- Critical-Alarm Threshold: -1
- Deny Threshold: -1
- Deny Period: -1

Buttons: Add, Cancel

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 **Add**, and then save ("burn") your settings to flash memory.

Table 13-4: IDS Rule Table Parameter Descriptions

Parameter	Description
Index rule-id [IDSRule_RuleID]	Defines an index number for the new table record.
Reason 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

Parameter	Description
	page 471) ✓ Routing failure ✓ Other local rejects (prior to SIP 180 response) ✓ Remote rejects (prior to SIP 180 response) ▪ [5] Abnormal flow = ✓ Requests and responses without a matching transaction user (except ACK requests) ✓ Requests and responses without a matching transaction (except ACK requests)
Threshold Scope 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 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 minor-alm-thr [IDSRule_MinorAlarmThresh old]	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 major-alm-thr [IDSRule_MajorAlarmThresh old]	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 critical-alm-thr [IDSRule_CriticalAlarmThres hold]	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: The 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).

13.3.3 Assigning IDS Policies

The IDS Match table lets you implement your configured IDS Policies. You do this by assigning 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 342.
- **Proxy Sets:** For detection of malicious attacks from specified Proxy Set(s). For configuring Proxy Sets, see "Configuring Proxy Sets" on page 360.
- **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 (**Configuration** tab > **VoIP** menu > **Security** > **Intrusion Detection and Prevention** > **Match Table**).
2. Click **Add**; the following dialog box appears:

Figure 13-8: IDS Match Table - Add Row Dialog Box

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 **Add**, and then save ("burn") your settings to flash memory.

Table 13-5: IDS Match Table Parameter Descriptions

Parameter	Description
Index [IDSMATCH_Index]	Defines an index number for the new table record.
SIP Interface ID 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)

Parameter	Description
Proxy Set ID proxy-set [IDSMATCH_ProxySet]	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 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 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 policy [IDSMATCH_Policy]	Assigns an IDS Policy (configured in "Configuring IDS Policies" on page 176).

13.3.4 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, IDSArmClearPeriod) 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 176). For example, if you set IDSArmClearPeriod 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 637). In this example, a Minor threshold alarm is cleared and replaced by a Major threshold alarm:

Figure 13-9: 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 have 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 occurs only if you have configured Syslog (see "Enabling Syslog" on page 690). An example of a Syslog message with IDS alarms and notifications is shown below:

Figure 13-10: 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_psbrdif(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_psbrdif(97426 ) InsertBoardEvent- event ADD BLACKLIST EV inserted channel -100
```

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

Table 13-6: 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

This page is intentionally left blank.

14 Media

This section describes the media-related configuration.

14.1 Configuring Voice Settings

The Voice Settings page configures various voice parameters such as voice volume and DTMF transport type. For a detailed description of these parameters, see "Configuration Parameters Reference" on page 719.

➤ **To configure the voice parameters:**

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

Figure 14-1: Voice Settings Page

Voice Settings		
Voice Volume (-32 to 31 dB)	0	
Input Gain (-32 to 31 dB)	0	
Silence Suppression	Disable	▼
DTMF Transport Type	Transparent DTMF	▼
DTMF Volume (-31 to 0 dB)	-11	
NTE Max Duration	-1	
⚡ DTMF Generation Twist	0	
Echo Canceller	Enable	▼

Acoustic Echo Suppressor Settings		
⚡ Network Echo Suppressor Enable	Disable	▼
Echo Canceller Type	Line echo canceller	▼
Attenuation Intensity	0	
Max ERL Threshold - DB	0	
Min Reference Delay x10 msec	0	
Max Reference Delay x10 msec	40	

2. Configure the Voice parameters as required.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

14.1.1 Configuring Voice Gain (Volume) Control

The device allows you to configure the level of the received (input gain) IP signal and the level of the transmitted (output gain) IP signal. The gain can be set between -32 and 31 decibels (dB).

The following procedure describes how to configure gain control using the Web interface.

➤ **To configure gain control using the Web interface:**

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

Figure 14-2: Voice Volume Parameters in Voice Settings Page

Voice Volume (-32 to 31 dB)	0
Input Gain (-32 to 31 dB)	0

2. Configure the following parameters:
 - 'Voice Volume' (*VoiceVolume*) - Defines the voice gain control (in decibels) of the transmitted signal.
 - 'Input Gain' (*InputGain*) - Defines the PCM input gain control (in decibels) of the received signal.
3. Click **Submit**.

14.1.2 Configuring Echo Cancellation

The device supports adaptive linear (line) echo cancellation according to G.168-2002. Echo cancellation is a mechanism that removes echo from the voice channel. Echoes are reflections of the transmitted signal.

In this line echo, echoes are generated when two-wire telephone circuits (carrying both transmitted and received signals on the same wire pair) are converted to a four-wire circuit. Echoes are reflections of the transmitted signal, which result from impedance mismatch in the hybrid (bi-directional 2-wire to 4-wire converting device).

An estimated echo signal is built by feeding the decoder output signal to an RLS-like adaptive filter, which adapts itself to the characteristics of the echo path. The 'estimated echo signal' (the output of this filter) is then subtracted from the input signal (which is the sum of the desired input signal and the undesired echo) to provide a clean signal. To suppress the remaining residual echo, a Non Linear Processor (NLP) is used, as well as a double-talk (two people speak at the same time) detector that prevents false adaptation during near-end speech.

The device also supports acoustic echo cancellation for SBC calls. These echoes are composed of undesirable acoustical reflections (non-linear) of the received signal (i.e., from the speaker) which find their way from multiple reflections such as walls and windows into the transmitted signal (i.e., microphone). Therefore, the party at the far end hears his / her echo. The device removes these echoes and sends only the near-end's desired speech signal to the network (i.e., to the far-end party). The echo is composed of a linear part and a nonlinear part. However, in the Acoustic Echo Canceller, a substantial part of the echo is non-linear echo. To support this feature, the Forced Transcoding feature must be enabled so that the device uses DSPs.

The following procedure describes how to configure echo cancellation using the Web interface:

➤ To configure echo cancellation using the Web interface:

1. Configure line echo cancellation:
 - a. Open the Voice Settings page (**Configuration** tab > **VoIP** menu > **Media** > **Voice Settings**).

Echo Canceller	Enable
▼ Acoustic Echo Suppressor Settings	
⚡ Network Echo Suppressor Enable	Disable
Echo Canceller Type	Line echo canceller
Attenuaion Intensity	0
Max ERL Threshold - DB	0
Min Reference Delay x10 msec	0
Max Reference Delay x10 msec	40

- b. Set the 'Echo Canceller' field (*EnableEchoCanceller*) to **Enable**.
2. Enable acoustic echo cancellation for SBC calls:

- a. In the Voice Settings page, configure the following parameters:
 - ◆ 'Network Echo Suppressor Enable' (AcousticEchoSuppressorSupport) - enables the network Acoustic Echo Suppressor
 - ◆ 'Echo Canceller Type' (EchoCancellerType) - defines the echo canceller type
 - ◆ 'Attenuation Intensity' (AcousticEchoSuppAttenuationIntensity) - defines the acoustic echo suppressor signals identified as echo attenuation intensity
 - ◆ 'Max ERL Threshold' (AcousticEchoSuppMaxERLThreshold) - defines the acoustic echo suppressor maximum ratio between signal level and returned echo from the phone
 - ◆ 'Min Reference Delay' (AcousticEchoSuppMinRefDelayx10ms) - defines the acoustic echo suppressor minimum reference delay
 - ◆ 'Max Reference Delay' (AcousticEchoSuppMaxRefDelayx10ms) - defines the acoustic echo suppressor maximum reference delay
- b. Open the IP Profile Settings page (Configuration tab > VoIP menu > Coders and Profiles > IP Profile Settings), and set the 'Echo Canceller' field to Acoustic.
- c. Enable the Forced Transcoding feature (using the TranscodingMode parameter) to allow the device to use DSP channels, which are required for acoustic echo cancellation.



Note: The following additional echo cancellation parameters are configurable only through the *ini* file:

- *ECHybridLoss* - defines the four-wire to two-wire worst-case Hybrid loss
- *ECNLPMode* - defines the echo cancellation Non-Linear Processing (NLP) mode
- *EchoCancellerAggressiveNLP* - enables Aggressive NLP at the first 0.5 second of the call

14.2 Fax and Modem Capabilities

This section describes the device's fax and modem capabilities and corresponding configuration. The fax and modem configuration is done in the Fax/Modem/CID Settings page.



Notes:

- Unless otherwise specified, the configuration parameters mentioned in this section are available on this page.
- Some SIP parameters override these fax and modem parameters. For example, the *IsFaxUsed* parameter and V.152 parameters in Section "V.152 Support" on page 198.
- For a detailed description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 719.

➤ **To access the fax and modem parameters:**

1. Open the Fax/Modem/CID Settings page (**Configuration** tab > **VoIP** menu > **Media** > **Fax/Modem/CID Settings**).

Figure 14-3: Fax/Modem/CID Settings Page

▼ General Settings		
Fax Transport Mode	RelayEnable	▼
Caller ID Transport Type	Mute	▼
Caller ID Type	Standard Bellcore	▼
V.21 Modem Transport Type	Disable	▼
V.22 Modem Transport Type	Enable Bypass	▼
V.23 Modem Transport Type	Enable Bypass	▼
V.32 Modem Transport Type	Enable Bypass	▼
V.34 Modem Transport Type	Enable Bypass	▼
Fax CNG Mode	Disable	▼
CNG Detector Mode	Disable	▼

▼ Fax Relay Settings		
Fax Relay Redundancy Depth	0	
Fax Relay Enhanced Redundancy Depth	4	
Fax Relay ECM Enable	Enable	▼
Fax Relay Max Rate (bps)	33600bps	▼

▼ Bypass Settings		
Fax/Modem Bypass Coder Type	G711Alaw_64	▼
Fax/Modem Bypass Packing Factor	1	
Fax Bypass Output Gain	0	
Modem Bypass Output Gain	0	

2. Configure the parameters, as required.
3. Click **Submit**.

14.2.1 Fax/Modem Operating Modes

The device supports two modes of operation:

- Fax/modem negotiation that is not performed during the establishment of the call.
- Voice-band data (VBD) mode for V.152 implementation (see "V.152 Support" on page 198): fax/modem capabilities are negotiated between the device and the remote endpoint at the establishment of the call. During a call, when a fax/modem signal is detected, transition from voice to VBD (or T.38) is automatically performed and no additional SIP signaling is required. If negotiation fails (i.e., no match is achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

14.2.2 Fax/Modem Transport Modes

The device supports the following transport modes for fax per modem type (V.22/V.23/Bell/V.32/V.34):

- T.38 fax relay (see "T.38 Fax Relay Mode" on page 189)
- G.711 Transport: switching to G.711 when fax/modem is detected (see "G.711 Fax / Modem Transport Mode" on page 192)
- Fax fallback to G.711 if T.38 is not supported (see "Fax Fallback" on page 192)
- Fax and modem bypass: a proprietary method that uses a high bit rate coder (see "Fax/Modem Bypass Mode" on page 193)
- NSE Cisco's Pass-through bypass mode for fax and modem (see "Fax / Modem NSE Mode" on page 194)
- Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (see "Fax / Modem Transparent with Events Mode" on page 195)
- Transparent: passing the fax / modem signal in the current voice coder (see "Fax / Modem Transparent Mode" on page 195)
- RFC 2833 ANS Report upon Fax/Modem Detection (see "RFC 2833 ANS Report upon Fax/Modem Detection" on page 196)

'Adaptations' refer to automatic reconfiguration of certain DSP features for handling fax/modem streams differently than voice.

14.2.2.1 T.38 Fax Relay Mode

In Fax Relay mode, fax signals are transferred using the T.38 protocol. T.38 is the ITU standard for sending fax across IP networks in real-time mode. The device currently supports only the T.38 UDP syntax.

T.38 can be configured in the following ways:

- Switching to T.38 mode using SIP Re-INVITE messages (see "Switching to T.38 Mode using SIP Re-INVITE" on page 189)
- Automatically switching to T.38 mode without using SIP Re-INVITE messages (see "Automatically Switching to T.38 Mode without SIP Re-INVITE" on page 190)

When fax transmission ends, the reverse switching from fax relay to voice is automatically performed at both the local and remote endpoints.

You can change the fax rate declared in the SDP, using the 'Fax Relay Max Rate' parameter (FaxRelayMaxRate). The parameter does not affect the actual transmission rate. You can also enable or disable Error Correction Mode (ECM) fax mode using the 'Fax Relay ECM Enable' parameter (FaxRelayECMEnable).

When using T.38 mode, you can define a redundancy feature to improve fax transmission over congested IP networks. This feature is activated using the 'Fax Relay Redundancy Depth' parameter (FaxRelayRedundancyDepth) and the 'Fax Relay Enhanced Redundancy Depth' parameter (FaxRelayEnhancedRedundancyDepth). Although this is a proprietary redundancy scheme, it should not create problems when working with other T.38 decoders.

14.2.2.1.1 Switching to T.38 Mode using SIP Re-INVITE

In the Switching to T.38 Mode using SIP Re-INVITE mode, upon detection of a fax signal the terminating device negotiates T.38 capabilities using a Re-INVITE message. If the far-end device doesn't support T.38, the fax fails. In this mode, the 'Fax Transport Mode' parameter (FaxTransportMode) is ignored.

➤ To configure T.38 mode using SIP Re-INVITE messages:

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **T.38 Relay** (IsFaxUsed = 1).

2. In the Fax/Modem/CID Settings page, configure the following optional parameters:
 - 'Fax Relay Redundancy Depth' (FaxRelayRedundancyDepth)
 - 'Fax Relay Enhanced Redundancy Depth' (FaxRelayEnhancedRedundancyDepth)
 - 'Fax Relay ECM Enable' (FaxRelayECMEnable)
 - 'Fax Relay Max Rate' (FaxRelayMaxRate)



Note: The terminating gateway sends T.38 packets immediately after the T.38 capabilities are negotiated in SIP. However, the originating device by default, sends T.38 (assuming the T.38 capabilities are negotiated in SIP) only after it receives T.38 packets from the remote device. This default behavior cannot be used when the originating device is located behind a firewall that blocks incoming T.38 packets on ports that have not yet received T.38 packets from the internal network. To resolve this problem, the device should be configured to send CNG packets in T.38 upon CNG signal detection (CNGDetectorMode = 1).

14.2.2.1.2 Automatically Switching to T.38 Mode without SIP Re-INVITE

In the Automatically Switching to T.38 Mode without SIP Re-INVITE mode, when a fax signal is detected, the channel automatically switches from the current voice coder to answer tone mode and then to T.38-compliant fax relay mode.

➤ **To configure automatic T.38 mode:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).
2. In the Fax/Modem/CID Settings page, set the 'Fax Transport Mode' parameter to **T.38 Relay** (FaxTransportMode = 1).
3. Configure the following optional parameters:
 - 'Fax Relay Redundancy Depth' (FaxRelayRedundancyDepth)
 - 'Fax Relay Enhanced Redundancy Depth' (FaxRelayEnhancedRedundancyDepth)

- 'Fax Relay ECM Enable' (FaxRelayECMEnable)
- 'Fax Relay Max Rate' (FaxRelayMaxRate)

14.2.2.1.3 Fax over IP using T.38 Transmission over RTP

The device supports Fax-over-IP (FoIP) transmission using T.38 over RTP, whereby the T.38 payload is encapsulated in the RTP packet, instead of being sent in dedicated T.38 packets (out-of-band). To configure this support, set the coder type to T.38 Over RTP.

To indicate T.38 over RTP, the SDP body uses "udptl" (Facsimile UDP Transport Layer) in the 'a=fmtp' line. The device supports T.38 over RTP according to this standard as well as according to AudioCodes proprietary method:

- **Call Parties belong to AudioCodes Devices:** AudioCodes proprietary T.38-over-RTP method is used, whereby the device encapsulates the entire T.38 packet (payload with all its headers) in the sent RTP. For T.38 over RTP, AudioCodes devices use the proprietary identifier "AcUdpTl" in the 'a=fmtp' line of the SDP. For example:

```
v=0
o=AudiocodesGW 1357424688 1357424660 IN IP4 10.8.6.68
s=Phone-Call
c=IN IP4 10.8.6.68
t=0 0
m=audio 6080 RTP/AVP 18 100 96
a=ptime:20
a=sendrecv
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:100 t38/8000
a=fmtp:100 T38FaxVersion=0
a=fmtp:100 T38MaxBitRate=0
a=fmtp:100 T38FaxMaxBuffer=3000
a=fmtp:100 T38FaxMaxDatagram=122
a=fmtp:100 T38FaxRateManagement=transferredTCF
a=fmtp:100 T38FaxUdpEC=t38UDPRedundancy
a=fmtp:100 AcUdpTl
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
```

- **AudioCodes Call Party with non-AudioCodes Party:** The device uses the standard T.38-over-RTP method, which encapsulates the T.38 payload only, without its headers (i.e., includes only fax data) in the sent RTP packet (RFC 4612).

The T.38-over-RTP method also depends on call initiator:

- **Device initiates a call:** The device always sends the SDP offer with the proprietary token "AcUdpTl" in the 'fmtp' attribute. If the SDP answer includes the same token, the device employs AudioCodes proprietary T.38-over-RTP mode; otherwise, the standard mode is used.
- **Device answers a call:** If the SDP offer from the remote party contains the 'fmtp' attribute with "AcUdpTl", the device answers with the same attribute and employs AudioCodes proprietary T.38-over-RTP mode; otherwise, the standard mode is used.



Note: If both T.38 (regular) and T.38 Over RTP coders are negotiated between the call parties, the device uses T.38 Over RTP.

14.2.2.2 G.711 Fax / Modem Transport Mode

In this mode, when the terminating device detects fax or modem signals (CED or AnsAM), it sends a Re-INVITE message to the originating device, requesting it to re-open the channel in G.711 VBD with the following adaptations:

- Echo Canceller = off
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off
- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

After a few seconds upon detection of fax V.21 preamble or super G3 fax signals, the device sends a second Re-INVITE enabling the echo canceller (the echo canceller is disabled only on modem transmission).

A 'gpmid' attribute is added to the SDP according to the following format:

- **For G.711 A-law:**

```
a=gpmid:0 vbd=yes;ecan=on (or off for modems)
```

- **For G.711 μ -law:**

```
a=gpmid:8 vbd=yes;ecan=on (or off for modems)
```

The following parameters are ignored and automatically set to **Events Only**:

- 'Fax Transport Mode' (FaxTransportMode)
- 'Vxx ModemTransportType' (VxxModemTransportType)

➤ To configure fax / modem transparent mode:

- In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **G.711 Transport** (IsFaxUsed = 2).

14.2.2.3 Fax Fallback

In this mode, when the terminating device detects a fax signal, it sends a Re-INVITE message to the originating device with T.38. If the remote device doesn't support T.38 (replies with SIP response 415 "Media Not Supported"), the device sends a new Re-INVITE with G.711 VBD with the following adaptations:

- Echo Canceller = on
- Silence Compression = off
- Echo Canceller Non-Linear Processor Mode = off
- Dynamic Jitter Buffer Minimum Delay = 40
- Dynamic Jitter Buffer Optimization Factor = 13

When the device initiates a fax session using G.711, a 'gpmid' attribute is added to the SDP according to the following format:

- **For G.711A-law:**

```
a=gpmid:0 vbd=yes;ecan=on
```

- **For G.711 μ -law:**

```
a=gpmid:8 vbd=yes;ecan=on
```

In this mode, the 'Fax Transport Mode' (FaxTransportMode) parameter is ignored and automatically set to **Disable** (transparent mode).

➤ **To configure fax fallback mode:**

- In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **Fax Fallback** (IsFaxUsed = 3).

14.2.2.4 Fax/Modem Bypass Mode

In this proprietary mode, when fax or modem signals are detected, the channel automatically switches from the current voice coder to a high bit-rate coder, according to the 'Fax/Modem Bypass Coder Type' parameter (FaxModemBypassCoderType). The channel is also automatically reconfigured with the following fax / modem adaptations:

- Disables silence suppression
- Enables echo cancellation for fax
- Disables echo cancellation for modem
- Performs certain jitter buffering optimizations

The network packets generated and received during the bypass period are regular voice RTP packets (per the selected bypass coder), but with a different RTP payload type according to the following parameters:

- 'Fax Bypass Payload Type' (FaxBypassPayloadType)
- ModemBypassPayloadType (ini file)

During the bypass period, the coder uses the packing factor, configured by the 'Fax/Modem Bypass Packing Factor' parameter (FaxModemBypassM). The packing factor determines the number of coder payloads (each the size of FaxModemBypassBasicRTPPacketInterval) that are used to generate a single fax/modem bypass packet. When fax/modem transmission ends, the reverse switching, from bypass coder to regular voice coder is performed.

➤ **To configure fax / modem bypass mode:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).
2. In the Fax/Modem/CID Settings page, do the following:
 - a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
 - b. Set the 'V.21 Modem Transport Type' parameter to **Enable Bypass** (V21ModemTransportType = 2).
 - c. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
 - d. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
 - e. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
 - f. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).
3. Set the ini file parameter, BellModemTransportType to 2 (Bypass).

4. Configure the following optional parameters:

- 'Fax/Modem Bypass Coder Type' (FaxModemBypassCoderType).
- 'Fax Bypass Payload Type' (FaxBypassPayloadType) - in the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** > **RTP/RTCP Settings**).
- ModemBypassPayloadType (ini file).
- FaxModemBypassBasicRTPPacketInterval (ini file).
- FaxModemBypassDJBufMinDelay (ini file).



Note: When the device is configured for modem bypass and T.38 fax, V.21 low-speed modems are not supported and fail as a result.



Tip: When the remote (non-AudioCodes) gateway uses the G.711 coder for voice and doesn't change the coder payload type for fax or modem transmission, it is recommended to use the Bypass mode with the following configuration:

- EnableFaxModemInbandNetworkDetection = 1.
- 'Fax/Modem Bypass Coder Type' = same coder used for voice.
- 'Fax/Modem Bypass Packing Factor'(FaxModemBypassM) = same interval as voice.
- ModemBypassPayloadType = 8 if voice coder is A-Law or 0 if voice coder is Mu-Law.

14.2.2.5 Fax / Modem NSE Mode

In this mode, fax and modem signals are transferred using Cisco-compatible Pass-through bypass mode. Upon detection of fax or modem answering tone signal, the terminating device sends three to six special NSE RTP packets (configured by the NSEpayloadType parameter; usually to 100). These packets signal the remote device to switch to G.711 coder, according to the 'Fax/Modem Bypass Packing Factor' parameter. After a few NSE packets are exchanged between the devices, both devices start using G.711 packets with standard payload type (8 for G.711 A-Law and 0 for G.711 Mu-Law). In this mode, no Re-INVITE messages are sent. The voice channel is optimized for fax/modem transmission (same as for usual bypass mode).

The parameters defining payload type for AudioCodes proprietary Bypass mode -- 'Fax Bypass Payload Type' (RTP/RTCP Settings page) and ModemBypassPayloadType (ini file) -- are not used with NSE Bypass.

When configured for NSE mode, the device includes in its SDP the following line:

```
a=rtpmap:100 X-NSE/8000
```

Where 100 is the NSE payload type.

The Cisco gateway must include the following definition:

```
modem passthrough nse payload-type 100 codec g711alaw
```

➤ **To configure NSE mode:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).
2. In the Fax/Modem/CID Settings page, do the following:
 - a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
 - b. Set the 'V.21 Modem Transport Type' parameter to **Enable Bypass** (V21ModemTransportType = 2).

- c. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
 - d. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
 - e. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
 - f. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).
3. Set the ini file parameter, BellModemTransportType to 2 (Bypass).
 4. Set the ini file parameter, NSEMode parameter to 1 (enables NSE).
 5. Set the ini file parameter, NSEPayloadType parameter to 100.

14.2.2.6 Fax / Modem Transparent with Events Mode

In this mode, fax and modem signals are transferred using the current voice coder with the following automatic adaptations:

- Echo Canceller = on (or off for modems)
- Echo Canceller Non-Linear Processor Mode = off
- Jitter buffering optimizations

➤ **To configure fax / modem transparent with events mode:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).
2. In the Fax/Modem/CID Settings page, do the following:
 - a. Set the 'Fax Transport Mode' parameter to **Events Only** (FaxTransportMode = 3).
 - b. Set the 'V.21 Modem Transport Type' parameter to **Events Only** (V21ModemTransportType = 3).
 - c. Set the 'V.22 Modem Transport Type' parameter to **Events Only** (V22ModemTransportType = 3).
 - d. Set the 'V.23 Modem Transport Type' parameter to **Events Only** (V23ModemTransportType = 3).
 - e. Set the 'V.32 Modem Transport Type' parameter to **Events Only** (V32ModemTransportType = 3).
 - f. Set the 'V.34 Modem Transport Type' parameter to **Events Only** (V34ModemTransportType = 3).
3. Set the ini file parameter, BellModemTransportType to 3 (transparent with events).

14.2.2.7 Fax / Modem Transparent Mode

In this mode, fax and modem signals are transferred using the current voice coder without notifications to the user and without automatic adaptations. It's possible to use Profiles (see "Coders and Profiles" on page 385) to apply certain adaptations to the channel used for fax / modem. For example, to use the coder G.711, to set the jitter buffer optimization factor to 13, and to enable echo cancellation for fax and disable it for modem.

➤ **To configure fax / modem transparent mode:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** (IsFaxUsed = 0).
2. In the Fax/Modem/CID Settings page, do the following:
 - a. Set the 'Fax Transport Mode' parameter to **Disable** (FaxTransportMode = 0).
 - b. Set the 'V.21 Modem Transport Type' parameter to **Disable** (V21ModemTransportType = 0).
 - c. Set the 'V.22 Modem Transport Type' parameter to **Disable** (V22ModemTransportType = 0).
 - d. Set the 'V.23 Modem Transport Type' parameter to **Disable** (V23ModemTransportType = 0).
 - e. Set the 'V.32 Modem Transport Type' parameter to **Disable** (V32ModemTransportType = 0).
 - f. Set the 'V.34 Modem Transport Type' parameter to **Disable** (V34ModemTransportType = 0).
3. Set the ini file parameter, BellModemTransportType to 0 (transparent mode).
4. Configure the following optional parameters:
 - a. Coders table - (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**).
 - b. 'Dynamic Jitter Buffer Optimization Factor' (DJBufOptFactor) - RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** > **RTP/RTCP Settings**).
 - c. 'Echo Cancellor' (EnableEchoCancellor) - Voice Settings page.



Note: This mode can be used for fax, but is not recommended for modem transmission. Instead, use the Bypass (see "Fax/Modem Bypass Mode" on page 193) or Transparent with Events modes (see "Fax / Modem Transparent with Events Mode" on page 195) for modem.

14.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection

The device (terminator gateway) sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem answer tones (i.e., CED tone). This causes the originator to switch to fax/modem. The parameter is applicable only when the fax or modem transport type is set to bypass, Transparent-with-Events, V.152 VBD, or G.711 transport. When the device is located on the originator side, it ignores these RFC 2833 events

➤ **To configure RFC 2833 ANS Report upon fax/modem detection:**

1. In the SIP General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), set the 'Fax Signaling Method' parameter to **No Fax** or **Fax Fallback** (IsFaxUsed = 0 or 3).
2. In the Fax/Modem/CID Settings page, do the following:
 - a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
 - b. Set the 'V.xx Modem Transport Type' parameters to **Enable Bypass** (VxxModemTransportType = 2).
3. Set the ini file parameter, FaxModemNTEMode to 1 (enables this feature).

14.2.3 V.34 Fax Support

V.34 fax machines can transmit data over IP to the remote side using various methods. The device supports the following modes for transporting V.34 fax data over IP:

- Bypass mechanism for V.34 fax transmission (see "Bypass Mechanism for V.34 Fax Transmission" on page 197)
- T38 Version 0 relay mode, i.e., fallback to T.38 (see "Relay Mode for T.30 and V.34 Faxes" on page 197)



Note: The CNG detector is disabled in all the subsequent examples. To disable the CNG detector, set the 'CNG Detector Mode' parameter (CNGDetectorMode) to **Disable**.

14.2.3.1 Bypass Mechanism for V.34 Fax Transmission

In this proprietary scenario, the device uses bypass (or NSE) mode to transmit V.34 faxes, enabling the full utilization of its speed.

➤ **To use bypass mode for T.30 and V.34 faxes:**

1. In the Fax/Modem/CID Settings page, do the following:
 - a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
 - b. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
 - c. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
 - d. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
 - e. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).

➤ **To use bypass mode for V.34 faxes, and T.38 for T.30 faxes:**

1. In the Fax/Modem/CID Settings page, do the following:
 - a. Set the 'Fax Transport Mode' parameter to **T.38 Relay** (FaxTransportMode = 1).
 - b. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
 - c. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
 - d. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
 - e. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).

14.2.3.2 Relay Mode for T.30 and V.34 Faxes

In this scenario, V.34 fax machines are forced to use their backward compatibility with T.30 faxes and operate in the slower T.30 mode.

➤ **To use T.38 mode for V.34 and T.30 faxes:**

1. In the Fax/Modem/CID Settings page, do the following:

- a. Set the 'Fax Transport Mode' parameter to **T.38 Relay** (FaxTransportMode = 1).
- b. Set the 'V.22 Modem Transport Type' parameter to **Disable** (V22ModemTransportType = 0).
- c. Set the 'V.23 Modem Transport Type' parameter to **Disable** (V23ModemTransportType = 0).
- d. Set the 'V.32 Modem Transport Type' parameter to **Disable** (V32ModemTransportType = 0).
- e. Set the 'V.34 Modem Transport Type' parameter to **Disable** (V34ModemTransportType = 0).

14.2.4 V.152 Support

The device supports the ITU-T recommendation V.152 (Procedures for Supporting Voice-Band Data over IP Networks). Voice-band data (VBD) is the transport of modem, facsimile, and text telephony signals over a voice channel of a packet network with a codec appropriate for such signals.

For V.152 capability, the device supports T.38 as well as VBD codecs (i.e., G.711 A-law and G.711 μ -law). The selection of capabilities is performed using the coders table (see "Configuring Default Coders" on page 385).

When in VBD mode for V.152 implementation, support is negotiated between the device and the remote endpoint at the establishment of the call. During this time, initial exchange of call capabilities is exchanged in the outgoing SDP. These capabilities include whether VBD is supported and associated RTP payload types ('gpmd' SDP attribute), supported codecs, and packetization periods for all codec payload types ('ptime' SDP attribute). After this initial negotiation, no Re-INVITE messages are necessary as both endpoints are synchronized in terms of the other side's capabilities. If negotiation fails (i.e., no match was achieved for any of the transport capabilities), fallback to existing logic occurs (according to the parameter IsFaxUsed).

Below is an example of media descriptions of an SDP indicating support for V.152. In the example, V.152 implementation is supported (using the dynamic payload type 96 and G.711 u-law as the VBD codec) as well as the voice codecs G.711 μ -law and G.729.

```
v=0
o=- 0 0 IN IPV4 <IPAddress>
s=-
t=0 0
p=+1
c=IN IP4 <IPAddress>
m=audio <udpPort A> RTP/AVP 18 0
a=ptime:10
a=rtpmap:96 PCMU/8000
a=gpmd: 96 vbd=yes
```

Instead of using VBD transport mode, the V.152 implementation can use alternative relay fax transport methods (e.g., fax relay over IP using T.38). The preferred V.152 transport method is indicated by the SDP 'pmft' attribute. Omission of this attribute in the SDP content means that VBD mode is the preferred transport mechanism for voice-band data. To configure T.38 mode, use the CodersGroup parameter.



Note: You can also configure the device to handle G.711 coders received in INVITE SDP offers as VBD coders, using the HandleG711asVBD parameter. 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 subsequent bypass (passthrough) sessions if fax / modem signals are detected during the call.

14.3 Configuring RTP/RTCP Settings

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

14.3.1 Configuring the Dynamic Jitter Buffer

Voice frames are transmitted at a fixed rate. If the frames arrive at the other end at the same rate, voice quality is perceived as good. However, some frames may arrive slightly faster or slower than the other frames. This is called jitter (delay variation) and degrades the perceived voice quality. To minimize this problem, the device uses a jitter buffer. The jitter buffer collects voice packets, stores them and sends them to the voice processor in evenly spaced intervals.

The device uses a dynamic jitter buffer that can be configured with the following:

- **Minimum delay:** Defines the starting jitter capacity of the buffer. For example, at 0 msec, there is no buffering at the start. At the default level of 10 msec, the device always buffers incoming packets by at least 10 msec worth of voice frames.
- **Optimization Factor:** Defines how the jitter buffer tracks to changing network conditions. When set at its maximum value of 12, the dynamic buffer aggressively tracks changes in delay (based on packet loss statistics) to increase the size of the buffer and doesn't decay back down. This results in the best packet error performance, but at the cost of extra delay. At the minimum value of 0, the buffer tracks delays only to compensate for clock drift and quickly decays back to the minimum level. This optimizes the delay performance but at the expense of a higher error rate.

The default settings of 10 msec Minimum delay and 10 Optimization Factor should provide a good compromise between delay and error rate. The jitter buffer 'holds' incoming packets for 10 msec before making them available for decoding into voice. The coder polls frames from the buffer at regular intervals in order to produce continuous speech. As long as delays in the network do not change (jitter) by more than 10 msec from one packet to the next, there is always a sample in the buffer for the coder to use. If there is more than 10 msec of delay at any time during the call, the packet arrives too late. The coder tries to access a frame and is not able to find one. The coder must produce a voice sample even if a frame is not available. It therefore compensates for the missing packet by adding a Bad-Frame-Interpolation (BFI) packet. This loss is then flagged as the buffer being too small. The dynamic algorithm then causes the size of the buffer to increase for the next voice session. The size of the buffer may decrease again if the device notices that the buffer is not filling up as much as expected. At no time does the buffer decrease to less than the minimum size configured by the Minimum delay parameter.

In certain scenarios, the **Optimization Factor is set to 13**: One of the purposes of the Jitter Buffer mechanism is to compensate for clock drift. If the two sides of the VoIP call are not synchronized to the same clock source, one RTP source generates packets at a lower rate, causing under-runs at the remote Jitter Buffer. In normal operation (optimization factor 0 to 12), the Jitter Buffer mechanism detects and compensates for the clock drift by occasionally dropping a voice packet or by adding a BFI packet.

Fax and modem devices are sensitive to small packet losses or to added BFI packets. Therefore, to achieve better performance during modem and fax calls, the Optimization Factor should be set to 13. In this special mode the clock drift correction is performed less frequently - only when the Jitter Buffer is completely empty or completely full. When such condition occurs, the correction is performed by dropping several voice packets simultaneously or by adding several BFI packets simultaneously, so that the Jitter Buffer returns to its normal condition.

The following procedure describes how to configure the jitter buffer using the Web interface.

➤ **To configure jitter buffer using the Web interface:**

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

Figure 14-4: Jitter Buffer Parameters in the RTP/RTCP Settings Page

General Settings		
Dynamic Jitter Buffer Minimum Delay	10	
Dynamic Jitter Buffer Optimization Factor	10	

2. Set the 'Dynamic Jitter Buffer Minimum Delay' parameter (DJBufMinDelay) to the minimum delay (in msec) for the Dynamic Jitter Buffer.
3. Set the 'Dynamic Jitter Buffer Optimization Factor' parameter (DJBufOptFactor) to the Dynamic Jitter Buffer frame error/delay optimization factor.
4. Click **Submit**.

14.3.2 Configuring RFC 2833 Payload

The following procedure describes how to configure the RFC 2833 payload using the Web interface:

➤ **To configure RFC 2833 payload using the Web interface:**

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

Figure 14-5: RFC 2833 Payload Parameters on RTP/RTCP Settings Page

RTP Redundancy Depth	0	
Packing Factor	1	
RFC 2833 TX Payload Type	96	
RFC 2833 RX Payload Type	96	
RFC 2198 Payload Type	104	

2. Configure the following parameters:
 - 'RTP Redundancy Depth' (RTPRedundancyDepth) - enables the device to generate RFC 2198 redundant packets.
 - 'RFC 2833 TX Payload Type' (RFC2833TxPayloadType) - defines the Tx RFC 2833 DTMF relay dynamic payload type.
 - 'RFC 2833 RX Payload Type' (RFC2833RxPayloadType) - defines the Rx RFC 2833 DTMF relay dynamic payload type.
 - 'RFC 2198 Payload Type' (RFC2198PayloadType) - defines the RTP redundancy packet payload type according to RFC 2198.
3. Click **Submit**.

14.3.3 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 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.

The port range is calculated using the following equation:

`BaseUDPPort` to 65,535

Where, *BaseUDPPort* is a parameter for configuring the lower boundary of the port range (default is 6000) and *number of channels* is the maximum number of channels purchased from AudioCodes (included in the installed Software License Key).

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 323). 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 through 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 14-6: RTP Based UDP Port in RTP/RTCP Settings Page



The screenshot shows a web interface for configuring RTP/RTCP settings. It features a lightning bolt icon and the label 'RTP Base UDP Port' next to a text input field containing the value '6000'.

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



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 342). 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.

14.4 Event Detection and Notification using X-Detect Header

The device can detect certain events in the media stream and notify of their detection to a remote application server, using the SIP X-Detect header. The request for event notification is done by the application server when establishing a SIP dialog (i.e., INVITE message) or during an already established call using a re-INVITE message.

The device can detect the following event types:

- Answering Machine Detection (AMD): Detects events that are related to the AMD feature. AMD detects whether an answering machine or live voice has answered the call. It can also be used to detect silence, or the beep sound played by an answering machine to indicate the end of the greeting message after which a voice message can be left. For more information on AMD, see "Answering Machine Detection (AMD)" on page 206.
- Call Progress Tone (CPT): Detects whether a specific tone, defined in the installed CPT file is received from the call. It can be used to detect the beep sound played by an answering machine (as mentioned above) and the busy, reorder and ring tones.



Note: Event detection is supported only for calls using the G.711 coder.

The X-Detect header is used for event detection as follows:

- X-Detect header in the INVITE message received from the application server requesting a specific event detection:
X-Detect: Request=[event type to detect]
- X-Detect header in the SIP response message -- SIP 183 (for early dialogs) or 200 OK (for confirmed dialogs) -- sent by the device to the application server specifying which of the requested events it can detect (absence of the X-Detect header indicates that the device cannot detect any of the events):
X-Detect: Response=[supported event types]
- Each time the device detects the supported event, it sends an INFO message to the remote party with the following message body:
Content-Type: Application/X-Detect
Type = [event type]
Subtype = [subtype of each event type]

The table below lists the event types and subtypes that the device can detect. The text shown in the table are the actual strings that are used in the X-Detect header. The table also provides a summary of the required configuration. For SBC calls, event detection is enabled using the IPProfile_SBCHandleXDetect parameter in the IP Profile table (see Configuring IP Profiles on page 391).

Table 14-1: Supported X-Detect Event Types

Event Type	Subtype	Description and Required Configuration
AMD	Voice (live voice) Automata (answering machine) Silence (no voice) Unknown Beep (greeting message of	Event detection using the AMD feature. For more information, see Answering Machine Detection (AMD) on page 206.

Event Type	Subtype	Description and Required Configuration
	answering machine)	
CPT	Busy Reorder Ringtone Beep (greeting message of answering message)	Event detection of tones using the CPT file. 1 Create a CPT file with the required tone types of the events that you want to detect. 2 Install the CPT file on the device. Note: For configuring beep detection, see Detecting Answering Machine Beep on page 203.

14.4.1 Detecting Answering Machine Beeps

The device supports the detection of the beep sound played by an answering machine to indicate the end of the answering machine's greeting message. This is useful in that the device can then notify, for example, a third-party, application server that it can now leave a voice message on the answering machine. The device supports the following methods for detecting and reporting beeps:

- **AMD-based Detection:** The device uses its beep detector that is integrated in the AMD feature. You can configure the beep detection timeout and beep detection sensitivity level (for more information, see "Configuring AMD" on page 208). To enable the AMD beep detection, the received INVITE message must contain an X-Detect header with the value "Request=AMD",

X-Detect: Request=AMD

and the AMDBeepDetectionMode parameter must be set to 1 or 2. If set to 1, the beep is detected only after the answering machine is detected. If set to 2, the beep is detected even if the answering machine was not detected.

- **Tone-based Detection (Call Progress Tone):** The device detects the beep according to a call progress tone (CPT). This is enabled if the device receives a specific beep tone (Tone Type #46) that is also defined in the installed CPT file, and the received INVITE message contains an X-Detect header with the value "Request=CPT":

X-Detect: Request=CPT

For more information on the CPT file, see "Call Progress Tones File" on page 586.

The device reports beep detections to application servers, by sending a SIP INFO message that contains a body with one of the following values, depending on the method used for detecting the beep:

- AMD-detected Beep:

Type= AMD
SubType= Beep

- CPT-detected Beep:

Type= CPT
SubType=Beep

14.4.2 SIP Call Flow Examples of Event Detection and Notification

Two SIP call flow examples are provided below of event detection and notification:

- The following example shows a SIP call flow of the device's AMD and event detection feature, whereby the device detects an answering machine and the subsequent start and end of the greeting message, enabling the third-party application server to know

when to play a recorded voice message to an answering machine:

1. Upon detection of the answering machine, the device sends the following SIP INFO message to the application server:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac1566945480
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c1505895240
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29758@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.80A.227.005
Content-Type: application/x-detect
Content-Length: 30
Type= AMD
SubType= AUTOMATA
```

2. Upon detection of the start of voice (i.e., the greeting message of the answering machine), the device sends the following INFO message to the application server:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c419779142
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.80A.227.005
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-START
```

3. Upon detection of the end of voice (i.e., end of the greeting message of the answering machine), the device sends the following INFO message to the application server:

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249:5060>;tag=1c419779142
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.6.80A.227.005
Content-Type: application/x-detect
Content-Length: 34
```


Type= **PTT**
SubType= **SPEECH-END**

4. The application server sends its message to leave on the answering message.

- The following example shows a SIP call flow for event detection and notification of the beep of an answering machine:

1. The device receives a SIP message containing the X-Detect header from the remote application requesting beep detection:

```
INVITE sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous"
<sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X-Detect: Request=AMD,CPT
```

2. The device sends a SIP response message to the remote party, listing the events in the X-Detect header that it can detect:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
From: "anonymous"
<sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>;tag=1c19282
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:101@10.33.2.53>
X-Detect: Response=AMD,CPT
```

3. The device detects the beep of an answering machine and sends an INFO message to the remote party:

```
INFO sip:101@10.33.2.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
Max-Forwards: 70
From: "anonymous"
<sip:anonymous@anonymous.invalid>;tag=1c25298
To: <sip:101@10.33.2.53;user=phone>
Call-ID: 11923@10.33.2.53
CSeq: 1 INVITE
Contact: <sip:100@10.33.2.53>
X-Detect: Response=AMD,CPT
Content-Type: Application/X-Detect
Content-Length: xxx
Type = CPT
Subtype = Beep
```

14.5 Answering Machine Detection (AMD)

The device's Answering Machine Detection (AMD) feature can detect whether an outbound call has been answered by a human (including fax) or an answering machine. The device analyzes the sound (speech) patterns received in the first few seconds of the call to determine whether a human (live person) or machine has answered the call. Typically, when a human answers the call, there is a short "hello ..." followed by silence to wait for the other party to respond. In contrast, when an answering machine answers the call, there is constant speech (answering message) followed by a beep to leave a voice-mail message.

When the device detects what answered the call (human or machine), it can notify this detection type to, for example, a third-party application server used for automatic dialing applications. The X-Detect SIP header is used for requesting event detection and notification. For more information, see "Event Detection and Notification using X-Detect Header" on page 202. The device can also detect beeps played by an answering machine at the end of its greeting message. For more information, see "Detecting Answering Machine Beeps" on page 203.

The device's default AMD feature is based on voice detection for North American English (see note below). It uses AudioCodes' sophisticated speech detection algorithms which are based on hundreds of real-life recordings of answered calls by live voice and answering machines in English. The algorithms are used to detect whether it's human or machine based on voice and silence duration as well as speech patterns. The algorithms of the language-based recordings are compiled into a file called AMD Sensitivity. This file is provided by default, pre-installed on the device.



Note: As the main factor (algorithm) for detecting human and machine is the voice pattern and silence duration, the language on which the detection algorithm is based, is in most cases not important as these factors are similar across most languages. Therefore, the default, pre-installed AMD Sensitivity file, which is based on North American English, may suffice your deployment even if the device is located in a region where a language other than English is used.

However, if (despite the information stated in the note above) you wish to implement AMD in a different language or region or if you wish to fine-tune the default AMD algorithms to suit your specific deployment, please contact your AudioCodes sales representative for more information on this service. You will be typically required to provide AudioCodes with a database of recorded voices (calls) in the language on which the device's AMD feature can base its voice detector algorithms. The data needed for an accurate calibration should be recorded under the following guidelines:

- Statistical accuracy: The number of recorded calls should be as high as possible (at least 100) and varied. The calls must be made to different people. The calls must be made in the specific location in which the device's AMD feature is to operate.
- Real-life recording: The recordings should simulate real-life answering of a called person picking up the phone, and without the caller speaking.
- Normal environment interferences: The environment in which the recordings are done should simulate real-life scenarios, in other words, not sterile but not too noisy either. Interferences, for example, could include background noises of other people talking, spikes, and car noises.

Once you have provided AudioCodes with your database of recordings, AudioCodes compiles it into a loadable file. For a brief description of the file format and for installing the file on the device, see "AMD Sensitivity File" on page 595.

The device supports up to eight AMD algorithm suites called *Parameter Suites*, where each suite defines a range of detection sensitivity levels. Sensitivity levels refer to how accurately, based on AudioCodes' voice detection algorithms, the device can detect

whether a human or machine has answered the call. Each level supports a different detection sensitivity to human and machine. For example, a specific sensitivity level may be more sensitive to detecting human than machine. In deployments where the likelihood of a call answered by an answering machine is low, it would be advisable to configure the device to use a sensitivity level that is more sensitive to human than machine. In addition, this allows you to tweak your sensitivity to meet local regulatory rules designed to protect consumers from automatic dialers (where, for example, the consumer picks up the phone and hears silence). Each suite can support up to 16 sensitivity levels (0 to 15), except for Parameter Suite 0, which supports up to 8 levels (0 to 7). The default, pre-installed AMD Sensitivity file, based on North American English, provides the following Parameter Suites:

- Parameter Suite 0 (normal sensitivity) - contains 8 sensitivity detection levels
- Parameter Suite 1 (high sensitivity) - contains 16 sensitivity detection levels

As Parameter Suite 1 provides a greater range of detection sensitivity levels (i.e., higher detection resolution), this may be the preferable suite to use in your deployment. The detected AMD type (human or machine) and success of detecting it correctly are sent in CDR and Syslog messages. For more information, see "Syslog Fields for Answering Machine Detection (AMD)" on page 686.

The Parameter Suite and sensitivity level can be applied globally for all calls, or for specific calls using IP Profiles. For enabling AMD and selecting the Parameter Suite and sensitivity level, see "Configuring AMD" on page 208.

The tables below show the success rates of the default, pre-installed AMD Sensitivity file (based on North American English) for correctly detecting "live" human voice and answering machine:

Table 14-2: Approximate AMD Normal Detection Sensitivity - Parameter Suite 0 (Based on North American English)

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
0 (Best for Answering Machine)	-	-
1	82.56%	97.10%
2	85.87%	96.43%
3	88.57%	94.76%
4	88.94%	94.31%
5	90.42%	91.64%
6	90.66%	91.30%
7 (Best for Live Calls)	94.72%	76.14%

Table 14-3: Approximate AMD High Detection Sensitivity - Parameter Suite 1 (Based on North American English)

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
0 (Best for Answering)	72%	97%

AMD Detection Sensitivity	Performance	
	Success Rate for Live Calls	Success Rate for Answering Machine
Machine)		
1	77%	96%
2	79%	95%
3	80%	95%
4	84%	94%
5	86%	93%
6	87%	92%
7	88%	91%
8	90%	89%
9	90%	88%
10	91%	87%
11	94%	78%
12	94%	73%
13	95%	65%
14	96%	62%
15 (Best for Live Calls)	97%	46%

14.5.1 Configuring AMD

You can configure AMD for all calls using the global AMD parameters, or for specific calls using IP Profiles. The procedure below describes how to configure AMD for all calls. For configuring AMD for specific calls, use the AMD parameters in the IP Profile table (see "Configuring IP Profiles" on page 391).

➤ To enable and configure AMD for all calls:

1. Open the IPMedia Settings page (Configuration tab > VoIP > Media > IPMedia Settings):

Figure 14-7: Configuring AMD Parameters in the IPMedia Settings Page

The screenshot shows the 'IPMedia Settings' page with the 'IPMedia Detectors' section expanded. The 'Enable Answer Detector' is set to 'Disable'. The 'Answer Machine Detector Sensitivity Parameter Suit' is set to '1'. The 'Answer Machine Detector Sensitivity Level' is set to '8'. The 'Answer Machine Detector Beep Detection Timeout' is set to '200'. The 'Answer Machine Detector Beep Detection Sensitivity' is set to '0'.

IPMedia Settings	
IPMedia Detectors	Enable
Enable Answer Detector	Disable
Answer Detector Activity Delay	0
Answer Detector Silence Time	10
Answer Detector Redirection	0
Answer Detector Sensitivity	0
Answer Machine Detector Sensitivity Parameter Suit	1
Answer Machine Detector Sensitivity Level	8
Answer Machine Detector Beep Detection Timeout	200
Answer Machine Detector Beep Detection Sensitivity	0

2. From the 'IPMedia Detectors' drop-down list (EnableDSPIPMDetectors), select **Enable** to enable AMD.
3. Select the AMD algorithm suite:
 - a. In the 'Answer Machine Detector Sensitivity Parameter Suit' field, select the required Parameter Suite included in the installed AMD Sensitivity file.
 - b. In the 'Answer Machine Detector Sensitivity' field, enter the required detection sensitivity level of the selected Parameter Suite.
4. Configure the answering machine beep detection:
 - a. In the 'Answer Machine Detector Beep Detection Timeout' field (AMDBeepDetectionTimeout), enter the duration that the beep detector operates from when detection is initiated.
 - b. In the 'Answer Machine Detector Beep Detection Sensitivity' field (AMDBeepDetectionSensitivity), enter the AMD beep detection sensitivity level.
5. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

For a complete list of AMD-related parameters, see "IP Media Parameters" on page 820.

14.6 Automatic Gain Control (AGC)

Automatic Gain Control (AGC) adjusts the energy of the output signal to a required level (volume). This feature compensates for near-far gain differences. AGC estimates the energy of the incoming signal from the IP, determined by the 'AGC Redirection' parameter, calculates the essential gain, and then performs amplification. Feedback ensures that the output signal is not clipped. You can configure the required Gain Slope in decibels per second using the 'AGC Slope' parameter and the required signal energy threshold using the 'AGC Target Energy' parameter.

When the AGC first detects an incoming signal, it begins operating in Fast Mode, which allows the AGC to adapt quickly when a conversation starts. This means that the Gain Slope is 8 dB/sec for the first 1.5 seconds. After this period, the Gain Slope is changed to the user-defined value. You can disable or enable the AGC's Fast Mode feature, using the *ini* file parameter AGCDisableFastAdaptation. After Fast Mode is used, the signal should be off for two minutes in order to have the feature turned on again.

The following procedure describes how to configure AGC using the Web interface:

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

1. Open the IPMedia Settings page (**Configuration** tab > **VoIP** menu > **Media** > **IPMedia Settings**). The AGC parameters are shown in the figure below:

Figure 14-8: AGC Parameters in IPMedia Settings Page

Enable AGC	Enable
AGC Slope	3
AGC Redirection	0
AGC Target Energy	19
⚡ AGC Minimum Gain	20
⚡ AGC Maximum Gain	15
⚡ AGC Disable Fast Adaptation	Disable

2. Configure the following parameters:
 - 'Enable AGC' (*EnableAGC*) - Enables the AGC mechanism.
 - 'AGC Slope' (*AGCGainSlope*) - Determines the AGC convergence rate.
 - 'AGC Redirection' (*AGCRedirection*) - Determines the AGC direction.
 - 'AGC Target Energy' - Defines the signal energy value (dBm) that the AGC attempts to attain.
 - 'AGC Minimum Gain' (*AGCMinGain*) - Defines the minimum gain (in dB) by the AGC when activated.
 - 'AGC Maximum Gain' (*AGCMaxGain*) - Defines the maximum gain (in dB) by the AGC when activated.
 - 'AGC Disable Fast Adaptation' (*AGCDisableFastAdaptation*) - Enables the AGC Fast Adaptation mode.
3. When using AGC with the SBC application, the 'Transcoding Mode' (*TranscodingMode*) parameter must be set to Force. The parameter can either be the global parameter or per IP Profile.
4. Click **Submit**.

14.7 Configuring Various Codec Attributes

The following codec attribute settings can be configured in the General Media Settings page:

- AMR coder:
 - 'Payload Format': Defines the AMR payload format type.
- SILK coder (Skype's default audio codec):
 - 'Silk Tx Inband FEC': Enables forward error correction (FEC) for the SILK coder.
 - 'Silk Max Average Bit Rate': Defines the maximum average bit rate for the SILK coder.

For a detailed description of these parameters and for additional codec parameters, see "Coder Parameters" on page 798.

➤ **To configure codec attributes:**

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

Figure 14-9: Codec Settings in General Media Settings Page

The screenshot shows the 'General Settings' page with the 'SILK Coders Settings' section expanded. It contains two rows: 'Silk Tx Inband FEC' with a dropdown menu set to 'Disable', and 'Silk Max Average Bit Rate' with a text input field containing '16000'. Below this, the 'AMR Bandwidth Efficient Configuration' section is also expanded, showing 'AMR Payload Format' with a dropdown menu set to 'Octet Aligned'.

2. Configure the parameters as required, and then click **Submit**.
3. To save the changes to flash memory, see "Saving Configuration" on page 578.

14.8 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 cryptographic key exchange mechanism to negotiate the keys. To negotiate the keys, the device supports the Session Description Protocol Security Descriptions (SDS) protocol (according to RFC 4568) or Datagram Transport Layer Security (DTLS) protocol for SBC calls. For more information on DTLS, see SRTP using DTLS Protocol on page 213. The key exchange is done by adding the '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 (SDP offer), it can generate a Master Key Identifier (MKI). You can configure the MKI size globally (using the `SRTPTxPacketMKISize` parameter) or per SIP entity (using the IP Profile parameter, `IpProfile_MKISize`). The length of the MKI is limited to four bytes. If the remote side sends a longer MKI, the key is ignored.



Note: The device can forward MKI size transparently for SRTP-to-SRTP media flows or override the MKI size during negotiation (inbound or outbound leg).

The key lifetime field is not supported. However, if it is included in the key it is ignored and the call does not fail. For SBC calls belonging to a specific SIP entity, you can configure the device to remove the lifetime field in the 'a=crypto' attribute (using the IP Profile parameter, `IpProfile_SBCRemoveCryptoLifetimeInSDP`).

For SDP, the keys are sent in the SDP body ('a=crypto') of the SIP message and are typically secured using SIP over TLS (SIPS). The encryption of the keys is in plain text in the SDP. The device supports the following session parameters:

- 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 forward the MKI size received in the SDP offer 'a=crypto' line in the SDP answer. You can enable symmetric MKI globally (using the `EnableSymmetricMKI` parameter) or per SIP entity (using the IP Profile parameter, `IpProfile_EnableSymmetricMKI` and `IpProfile_SBCEnforceMKISize`). For more information on symmetric MKI, see "Configuring IP Profiles" on page 391.

You can configure the enforcement policy of SRTP, using the `IpProfile_SBCMediaSecurityBehaviour` parameter. For example, if negotiation of the cipher suite fails or if incoming calls exclude encryption information, the device can be configured to reject the calls.

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 "Configuring IP Profiles" on page 391 and "SRTP Parameters" on page 757.
- When SRTP is used, the channel capacity may be reduced.

The procedure below describes how to configure SRTP through the Web interface.

➤ **To enable and configure SRTP:**

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

Figure 14-10: Media Security Page

General Media Security Settings	
Media Security	Disable
Aria Protocol Support	Disable
Media Security Behavior	Preferable
Authentication On Transmitted RTP Packets	Active
Encryption On Transmitted RTP Packets	Active
Encryption On Transmitted RTCP Packets	Active
SRTP Tunneling Authentication for RTP	Disable
SRTP Tunneling Authentication for RTCP	Disable
SRTP Setting	
Master Key Identifier (MKI) Size	0
Symmetric MKI Negotiation	Disable
SRTP Offered Suites	
Offered SRTP Cipher Suites	All

2. From the 'Media Security' drop-down list (EnableMediaSecurity), select **Enable** to enable SRTP.
3. Configure the other SRTP parameters as required.
4. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

14.8.1 SRTP using DTLS Protocol

For SBC calls, you can configure the device to use the Datagram Transport Layer Security (DTLS) protocol to secure UDP-based media streams (according to RFC 5763 and 5764) for specific SIP entities, using IP Profiles. DTLS allows datagram-based applications to communicate in a way that is designed to prevent eavesdropping, tampering or message forgery. The DTLS protocol is based on the stream-oriented TLS protocol, providing similar security. The device can therefore, interwork in mixed environments where one network may require DTLS and the other may require Session Description Protocol Security Descriptions (SDES) or even non-secure RTP. The device supports DTLS negotiation for RTP-to-SRTP and SRTP-to-SRTP calls.

DTLS support is important for deployments with WebRTC. WebRTC requires that media channels be encrypted through DTLS for SRTP key exchange. Negotiation of SRTP keys through DTLS is done during the DTLS handshake between WebRTC client and peer. For more information on WebRTC, see "WebRTC" on page 523.

In contrast to SDDES, DTLS key encryption is done over the media channel (UDP), not signaling. Thus, DTLS-SRTP is generally known as "secured key exchange over media". DTLS is similar to TLS, but runs over UDP, whereas TLS is over TCP. Before the DTLS handshake, the peers exchange DTLS parameters (fingerprint and setup) and algorithm types in the SDP body of the SIP messages exchanged for establishing the call (INVITE request and response). The peers participate in a DTLS handshake during which they exchange certificates. These certificates are used to derive a symmetric key, which is used to encrypt data (SRTP) flow between the peers. A hash value calculated over the certificate is transported in the SDP using the 'a=fingerprint' attribute. At the end of the handshake, each side verifies that the certificate it received from the other side fits the fingerprint from the SDP. To indicate DTLS support, the SDP offer/answer of the SIP message uses the 'a=setup' attribute. The 'a=setup:actpass' attribute value is used in the SDP offer by the

device. This indicates that the device is willing to be either a client ('act') or a server ('pass') in the handshake. The 'a=setup:active' attribute value is used in the SDP answer by the device. This means that the device wishes to be the client ('active') in the handshake.

```
a=setup:actpass
a=fingerprint: SHA-1
\4A:AD:B9:B1:3F:82:18:3B:54:02:12:DF:3E:5D:49:6B:19:E5:7C:AB
```

DTLS cipher suite reuses the TLS cipher suite. The DTLS handshake is done for every new call configured for DTLS. In other words, unlike TLS where the connection remains "open" for future calls, a new DTLS connection is required for every new call. Note that the entire authentication and key exchange for securing the media traffic is handled in the media path through DTLS. The signaling path is used only to verify the peers' certificate fingerprints. DTLS messages are multiplexed onto the same ports that are used for the media.

➤ **To configure DTLS:**

1. In the TLS Context table (see "Configuring TLS Certificate Contexts" on page 109), configure a TLS Context for DTLS.
2. Open the IP Group table (see "Configuring IP Groups" on page 348) and for the IP Group associated with the SIP entity, assign it the TLS Context for DTLS, using the 'DTLS Context' parameter (IPGroup_DTLSContext).
3. Open the IP Profile table (see "Configuring IP Profiles" on page 391) and for the IP Profile associated with the SIP entity, configure the following:
 - Configure the 'SBC Media Security Mode' parameter (IPProfile_SBCMediaSecurityBehavior) to **SRTP** or **Both**.
 - Configure the 'Media Security Method' parameter (IPProfile_SBCMediaSecurityMethod) to **DTLS**.
 - Configure the 'RTCP Mux' parameter (IPProfile_SBCRTCPMux) to **Supported**. Multiplexing is required as the DTLS handshake is done for the port used for RTP and thus, RTCP and RTP must be multiplexed onto the same port.
 - Configure the ini file parameter, SbcDtlsMtu (or CLI command configure voip > sbc general-setting > sbc-dtls-mtu) to define the maximum transmission unit (MTU) size for the DTLS handshake.



Notes:

- The 'Cipher Server' parameter must be configured to "ALL".
- The device does not support forwarding of DTLS transparently between endpoints.

15 Services

This section describes configuration for various supported services.

15.1 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 10,000 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 215) 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 220.

15.1.1 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 15-1: 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

DHCP Option Code	DHCP Option Name
Option 2	Time Offset
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 220
- Additional DHCP Options - see "Configuring Additional DHCP Options" on page 221
- Static IP addresses for DHCP clients - see "Configuring Static IP Addresses for DHCP Clients" on page 223



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 224.

The following procedure describes how to configure the DHCP server through the Web interface. You can also configure it through ini file (DhcpServer) or CLI (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**).

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

Figure 15-1: DHCP Servers Table - Add Row Dialog Box

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

- Index: 10
- Interface Name: None
- Start IP Address: 192.168.0.100
- End IP Address: 192.168.0.149
- Subnet Mask: 255.255.255.0
- 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

- Configure a DHCP server according to the parameters described in the table below.
- Click **Add**.

Table 15-2: DHCP Servers Table Parameter Descriptions

Parameter	Description
Index dhcp server <index>	Defines an index number for the new table row. Notes: - Each row must be configured with a unique index. - Currently, only one index row can be configured.
Interface Name 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" on page 138). By default, no value is defined.
Start IP Address 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.

Parameter	Description
End IP Address 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'.
Subnet Mask 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.
Lease Time 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 214,748,364. The default is 1440. When set to 0, the lease time is infinite.
DNS Server 1 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.
DNS Server 2 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).
NetBIOS Name Server 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.
NetBIOS Node Type 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
NTP Server 1 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.
NTP Server 2 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.

Parameter	Description
Time Offset 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.
TFTP Server 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.
Boot file name 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.
Expand Boot-file Name 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)
Override Router 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.
SIP Server 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 the 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.

Parameter	Description
SIP server type 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 configured with an IP address of the SIP server.

15.1.2 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 through the Web interface. You can also configure it through ini file (DhcpVendorClass) or CLI (configure voip > dhcp vendor-class).

➤ To configure DHCP Vendor Class Identifiers:

1. Open the DHCP Servers table (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the 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 below the table; the DHCP Vendor Class table opens.
3. Click **Add**; the following dialog box appears:

Figure 15-2: DHCP Vendor Class Table - Add Row Dialog Box

The screenshot shows a dialog box titled "Add Row" with a close button (X) in the top right corner. Inside the dialog, there are three labeled input fields: "Index" (a text box with the value "1"), "DHCP Server Index" (a dropdown menu currently showing "None"), and "Vendor Class Identifier" (a text box). At the bottom right of the dialog, there are two buttons: "Add" and "Cancel".

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

Table 15-3: DHCP Vendor Class Table Parameter Descriptions

Parameter	Description
Index dhcp_vendor-class <index> [DhcpVendorClass_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
DHCP Server Index dhcp-server-number [DhcpVendorClass_DhcpServerIndex]	Associates the VCI table entry with a DHCP server that you configured in "Configuring the DHCP Server" on page 215. Note: Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0.
Vendor Class Identifier 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.

15.1.3 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 through the Web interface. You can also configure it through ini file (DhcpOption) or CLI (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 table (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the 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 below the table; the DHCP Option table opens.

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

Figure 15-3: DHCP Option Table - Add Row Dialog Box

The dialog box titled "Add Row" contains the following fields and values:

- Index: 0
- DHCP Server Index: None
- Option: 159
- Type: ASCII
- Value: (empty text box)
- Expand Value: Yes

Buttons: Add, Cancel

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

Table 15-4: DHCP Option Table Parameter Descriptions

Parameter	Description
Index dhcp_option [DhcpOption_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
DHCP Server Index 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 215. Note: Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0.
Option 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.
Type 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
Value value [DhcpOption_Value]	Defines the value of the DHCP Option. For example, if you are using Option 66, the 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>
Expand Value 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: The parameter is applicable only to values of type ASCII (see the 'Type' parameter above).

15.1.4 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 through the Web interface. You can also configure it through ini file (DhcpStaticIP) or CLI (configure voip > dhcp static-ip <index>).

➤ **To configure static IP addresses for DHCP clients:**

1. Open the DHCP Servers table (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the 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 below the table; the DHCP Static IP table opens.

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

Figure 15-4: DHCP Static IP Table - Add Row Dialog Box

The dialog box titled 'Add Row' has a close button (X) in the top right corner. It contains four labeled input fields: 'Index' with the value '0', 'DHCP Server Index' with a dropdown menu showing 'None', 'IP Address' with the value '0.0.0.0', and 'MAC Address' with the value '00:90:8f:00:00:00'. At the bottom right, there are two buttons: 'Add' and 'Cancel'.

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

Table 15-5: DHCP Static IP Table Parameter Descriptions

Parameter	Description
Index dhcp static-ip <index> [DhcpStaticIP_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
DHCP Server Index 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 215. Note: Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0.
IP Address ip-address [DhcpStaticIP_IPAddress]	Defines the "reserved", static IP address (IPv4) to assign the DHCP client. The default is 0.0.0.0.
MAC Address mac-address [DhcpStaticIP_MACAddresses]	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.

15.1.5 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 through the Web interface. You can also view this through CLI:

- 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 table (**Configuration** tab > **VoIP** menu > **Services** > **DHCP Servers**).
2. In the 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 below the table; the DHCP Clients table opens:

Figure 15-5: 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 215) 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.

15.2 SIP-based Media Recording

The device can record SIP-based media call sessions traversing it. The media recording 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.



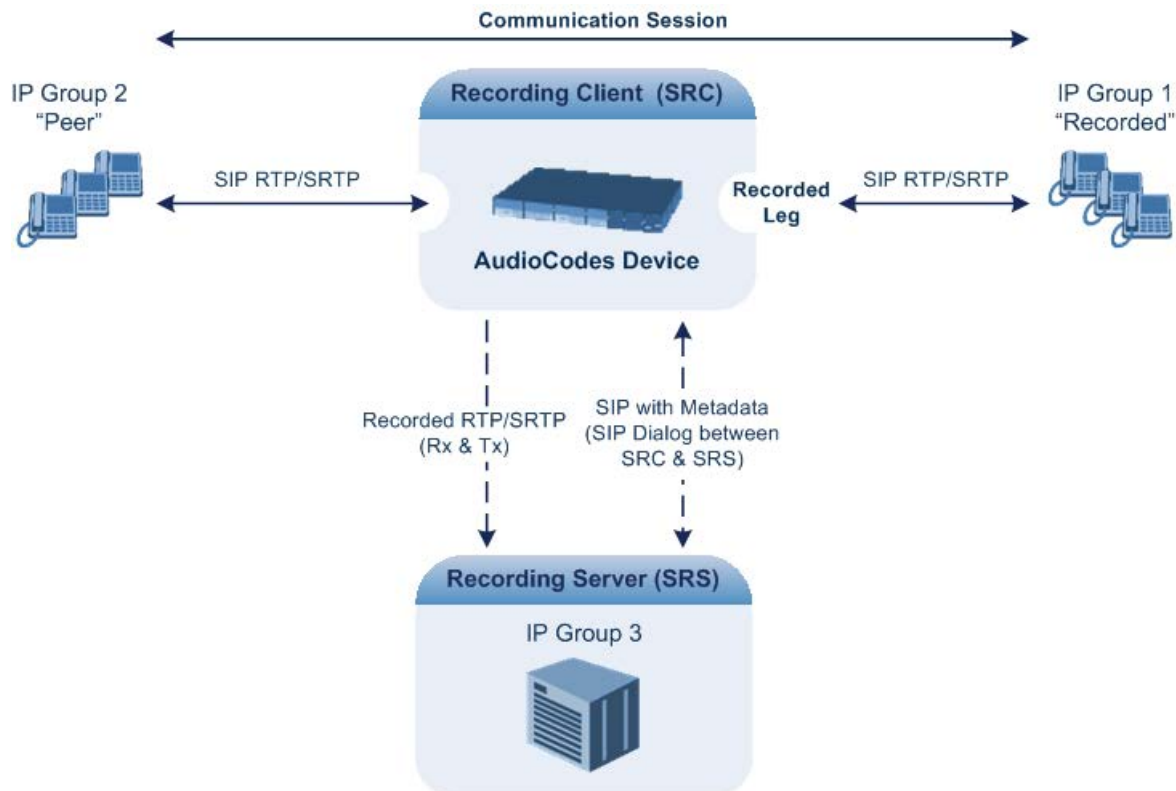
Notes:

- 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 596. 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.

Figure 15-6: SIP-based Recording where Device Serving as SRC



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) such as 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 SBC calls. For an SBC RTP-SRTP session, the recorded IP Group in the SIP Recording 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 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.227.005
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:Avaya
aUCID>
    </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-

```

15.2.1 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 Rules" on page 230) 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.

15.2.2 Configuring SIP Recording Rules

The SIP Recording table lets you configure up to 30 SIP-based media recording rules. A SIP Recording rule defines calls that you want to record. For an overview of this feature, see "SIP-based Media Recording" on page 226.



Note: To configure the device's timestamp format (local or UTC) in SIP messages sent to the SRS, see the SIPRecTimeStamp parameter.

The following procedure describes how to configure SIP Recording Routing rules through the Web interface. You can also configure it through ini file (SIPRecRouting) or CLI (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 15-7: SIP Recording Table - Add Row Dialog Box

The figure above shows a configuration example where the device records calls made by IP Group "ITSP" to IP Group "IP PBX" that have the destination number prefix, "1800". The device records the calls from the leg interfacing with IP Group "IP PBX", sending the recorded media to IP Group "SRS".

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

Table 15-6: SIP Recording Table Parameter Descriptions

Parameter	Description
Index [SIPRecRouting_Index]	Defines an index number for the new table record.
Recorded IP Group recorded-ip-group-name	Defines the IP Group participating in the call and the recording is done on the leg interfacing with this IP Group. For configuring IP Groups, see "Configuring IP Groups" on page 348.

Parameter	Description
[SIPRecRouting_RecordedIPGroupName]	By default, all IP Groups are defined (Any). 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 recorded-src-prefix [SIPRecRouting_RecordedSourcePrefix]	Defines calls to record based on source number or URI. By default, all source numbers or URIs are defined (*).
Recorded Destination Prefix recorded-dst-prefix [SIPRecRouting_RecordedDestinationPrefix]	Defines calls to record based on destination number or URI. By default, all destination numbers or URIs are defined (*).
Peer IP Group peer-ip-group-name [SIPRecRouting_PeerIPGroupName]	Defines the peer IP Group that is participating in the call. By default, all IP Groups are defined (Any).
Caller 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 srs-ip-group-name [SIPRecRouting_SRSIPGroupName]	Defines the IP Group of the recording server (SRS). By default, no value is defined. (None). 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).

15.2.3 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.

15.2.4 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.

15.2.4.1 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">4B0KLLA3VH66JF112M1CC9VHKS1
4F0KP</ac:GenesysUUID>
```

No configuration is required for this support.

15.2.4.2 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.

15.3 RADIUS-based Services

The device supports Remote Authentication Dial In User Service (RADIUS), by acting as a RADIUS client. You can use RADIUS for the following:

- Authentication and authorization of management users (login username and password) to gain access to the device's management interface.
- Accounting where the device sends accounting data of SIP calls as call detail records (CDR) to a RADIUS Accounting server (for third-party billing purposes).

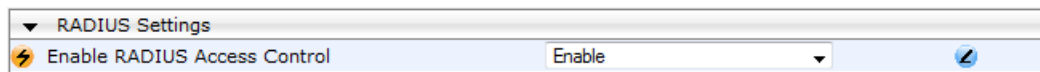
15.3.1 Enabling RADIUS Services

Before you can implement any RADIUS services, you must enable the RADIUS feature, as described in the procedure below.

➤ **To enable RADIUS:**

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

Figure 15-8: Enabling RADIUS



2. From the 'Enable RADIUS Access Control' drop-down list, select **Enable**.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

15.3.2 Configuring RADIUS Servers

The RADIUS Servers table lets you configure up to three RADIUS servers. The RADIUS servers can be used for RADIUS-based management-user login authentication and/or RADIUS-based accounting (sending of SIP CDRs to the RADIUS server).

When multiple RADIUS servers are configured, RADIUS server redundancy can be implemented. When the primary RADIUS server is down, the device sends a RADIUS request twice (one retransmission) and if both fail (i.e., no response), the device considers the server as down and attempts to send requests to the next server. The device continues sending RADIUS requests to the redundant RADIUS server even if the primary server returns to service later on. However, if a device reset occurs or a switchover occurs in a High-Availability (HA) system, the device sends RADIUS requests to the primary RADIUS server. By default, the device waits for up to two seconds (i.e., timeout) for a response from the RADIUS server for RADIUS requests and retransmission before it considers the server as down.

For each RADIUS server, the IP address, port, and shared secret can be configured. Each RADIUS server can be defined for RADIUS-based login authentication and/or RADIUS-based accounting. By setting the relevant port (authentication or accounting) to "0" disables the corresponding functionality. If both ports are configured, the RADIUS server is used for authentication and accounting. All servers configured with non-zero Authorization ports form an Authorization redundancy group and the device sends authorization requests to one of them, depending on their availability. All servers configured with non-zero Accounting ports form an Accounting redundancy group and the device sends accounting CDRs to one of them, depending on their availability. Below are example configurations:

- Only one RADIUS server is configured and used for authorization and accounting purposes (no redundancy). Therefore, both the Authorization and Accounting ports are defined.
- Three RADIUS servers are configured:
 - Two servers are used for authorization purposes only, providing redundancy. Therefore, only the Authorization ports are defined, while the Accounting ports are set to 0.
 - One server is used for accounting purposes only (i.e., no redundancy). Therefore, only the Accounting port is defined, while the Authorization port is set to 0.
- Two RADIUS servers are configured and used for authorization and accounting purposes, providing redundancy. Therefore, both the Authorization and Accounting ports are defined.

The status of the RADIUS servers can be viewed using the following CLI command:

```
# show system radius servers status
```

The example below shows the status of two RADIUS servers in redundancy mode for authorization and accounting:

```
servers 0
ip-address 10.4.4.203
auth-port 1812
auth-ha-state "ACTIVE"
acc-port 1813
acc-ha-state "ACTIVE"
servers 1
ip-address 10.4.4.202
auth-port 1812
auth-ha-state "STANDBY"
acc-port 1813
acc-ha-state "STANDBY"
```

Where *auth-ha-state* and *acc-ha-state* display the authentication and accounting redundancy status respectively. "ACTIVE" means that the server was used for the last sent authentication or accounting request; "STANDBY" means that the server was not used in the last sent request.

The following procedure describes how to configure a RADIUS server through the Web interface. You can also configure it through ini file (RadiusServers) or CLI configure system > radius > servers).



Note: To enable and configure RADIUS-based accounting, see "Configuring RADIUS Accounting" on page 667.

➤ **To configure a RADIUS server:**

1. Open the RADIUS Servers table (**Configuration** tab > **System** menu > **Management** > **RADIUS Servers**).

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

Figure 15-9: RADIUS Servers Table - Add Row Dialog Box

The dialog box titled "Add Row" contains the following fields and values:

- Index:
- IP Address:
- Authentication Port:
- Accounting Port:
- Shared Secret:

Buttons: Add, Cancel

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

Table 15-7: RADIUS Servers Table Parameter Descriptions

Parameter	Description
Index [RadiusServers_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
IP Address ip-address [RadiusServers_IPAddress]	Defines the IP address of the RADIUS server (in dotted-decimal notation).
Authentication Port auth-port [RadiusServers_AuthenticationPort]	Defines the port of the RADIUS Authentication server for authenticating the device with the RADIUS server. When set to any value other than 0, the RADIUS server is used by the device for RADIUS-based management-user login authentication. When set to 0, RADIUS-based login authentication is not implemented. The valid value is 0 to any integer. The default is 1645.
Accounting Port acc-port [RadiusServers_AccountingPort]	Defines the port of the RADIUS Accounting server to where the device sends accounting data of SIP calls as call detail records (CDR). When set to any value other than 0, the RADIUS server is used by the device for RADIUS-based accounting (CDR). When set to 0, RADIUS-based accounting is not implemented. The valid value is 0 to any integer. The default is 1646.
Shared Secret shared-secret [RadiusServers_SharedSecret]	Defines the shared secret (password) for authenticating the device with the RADIUS server. This should be a cryptically strong password. The shared secret is also used by the RADIUS server to verify the authentication of the RADIUS messages sent by the device (i.e., message integrity). The valid value is up to 48 characters. By default, no value is defined.

15.3.3 Configuring Interface for RADIUS Communication

The device can communicate with the RADIUS server through its' OAMP (default) or SIP Control network interface. To change the interface used for RADIUS traffic, use the RadiusTrafficType parameter.



Note: If set to Control, only one Control interface must be configured in the Interface table (see "Configuring IP Network Interfaces" on page 138); otherwise, RADIUS communication will fail.

15.3.4 Configuring General RADIUS Parameters

The procedure below describes the configuration of RADIUS parameters that are common between RADIUS-based user authentication and RADIUS-based accounting.

➤ **To configure general RADIUS parameters:**

1. Open the Authentication Settings page (**Configuration** tab > **System** menu > **Management** > **Authentication Settings**).
2. Scroll down the page to the RADIUS Settings group.
3. In the 'RADIUS VSA Vendor ID' field, enter the **same** vendor ID number as set on the third-party RADIUS server. The vendor-specific attribute (VSA) identifies the device to the RADIUS server using the Vendor ID. For an example of using the Vendor ID, see "Setting Up a Third-Party RADIUS Server" on page 237.
4. Configure RADIUS packet retransmission when no response is received from the RADIUS server:
 - a. In the 'RADIUS Packets Retransmission' field (RADIUSRetransmission), enter the maximum number of RADIUS retransmissions that the device performs if no response is received from the RADIUS server.
 - b. In the 'RADIUS Response Time Out' field (RadiusTO), enter the interval (in seconds) that the device waits for a response before sending a RADIUS retransmission.
5. Click **Submit**.

15.3.5 RADIUS-based Management User Authentication

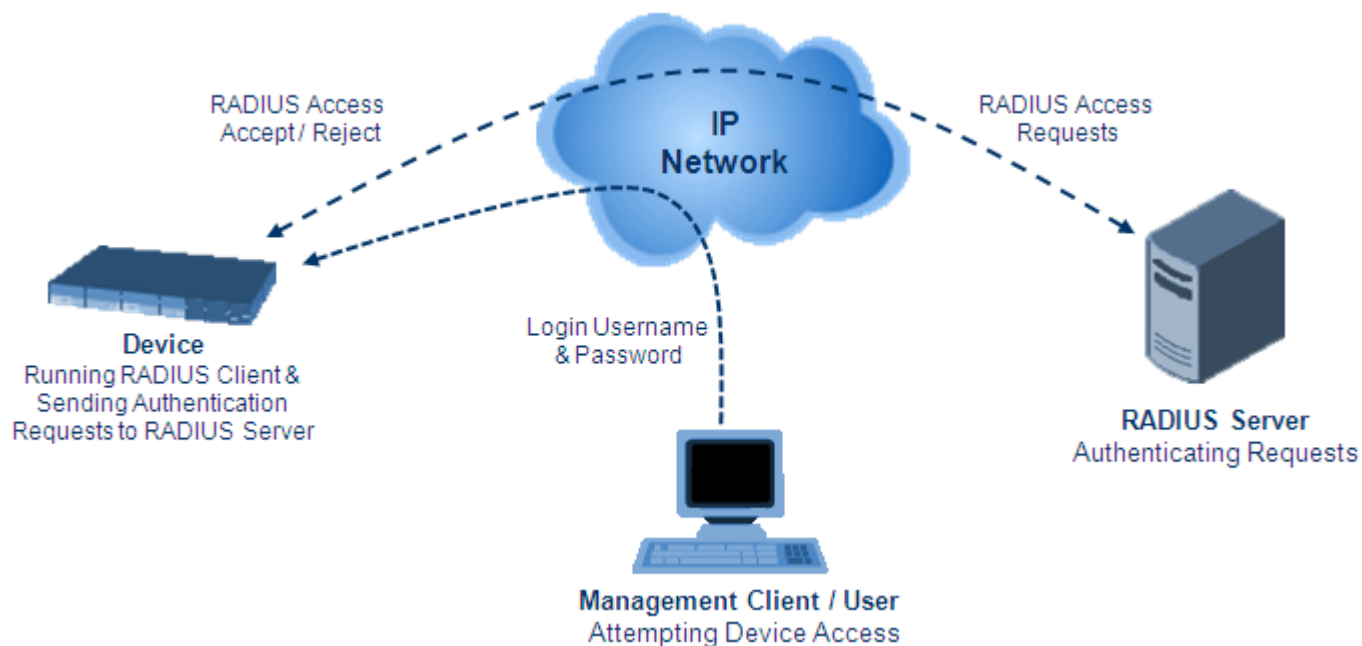
You can enhance security for your device by implementing Remote Authentication Dial-In User Service (RADIUS - RFC 2865) for authenticating multiple management user accounts of the device's embedded Web and Telnet (CLI) servers. Thus, RADIUS also prevents unauthorized access to your device.

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" on page 67.

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 15-10: 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 237
- Configure the device as a RADIUS client for communication with the RADIUS server - see "Configuring RADIUS Authentication" on page 238

15.3.5.1 Setting Up a Third-Party RADIUS Server

The following procedure provides an example for setting up a third-party RADIUS server, *FreeRADIUS*, which can be downloaded from <http://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 device as an authorized client of the RADIUS server, with the following:
 - Predefined *shared secret* (password used to secure communication between the device and the RADIUS server)
 - Vendor ID

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

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

2. If access levels are required, set up a Vendor-Specific Attributes (VSA) dictionary for

the RADIUS server and select an attribute ID that represents each user's access level. The example below shows a dictionary file for FreeRADIUS that defines the attribute "ACL-Auth-Level" with "ID=35". For the device's user access levels and their corresponding numeric representation in RADIUS servers, see "Configuring Web User Accounts" on page 67.

```
#
# 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.

15.3.5.2 Configuring RADIUS-based User Authentication

The following procedure describes how to configure the RADIUS parameters specific to login authentication. For a detailed description of the RADIUS parameters, see "RADIUS Parameters" on page 826.

➤ **To configure RADIUS parameters for login authentication:**

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

Figure 15-11: 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)	900
Default Access Level	200

LDAP settings	
⚡ Use LDAP for Web/Telnet Login	Disable ▼

RADIUS Settings	
⚡ Enable RADIUS Access Control	Disable ▼
Use RADIUS for Web/Telnet Login	Disable ▼
RADIUS VSA Vendor ID	5003
RADIUS VSA Access Level Attribute	35
RADIUS Response Time Out (sec)	2
RADIUS Packets Retransmission	1

2. From the 'Use RADIUS for Web/Telnet Login' drop-down list, select **Enable** to enable RADIUS authentication for Web and Telnet login.
3. 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 237.
 - **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.
4. 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.
5. 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 if a server is configured but connectivity with the server is down (if the server is up, the device authenticates the user with the server).
 - **Always:** First attempts to authenticate the user using the Web Users table, but if not found, it authenticates the user with the RADIUS server.
6. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

15.3.5.3 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**).

15.3.5.4 RADIUS-based User Authentication in URL

RADIUS authentication of the management user 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's 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.

15.3.6 RADIUS-based CDR Accounting

Once you have configured a RADIUS server(s) for accounting in "Configuring RADIUS Servers" on page 233, you need to enable and configure RADIUS-based CDR accounting (see "Configuring RADIUS Accounting" on page 667).

15.4 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 250). 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 264.

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 253.

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=abc,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 251). 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 DNs (Base Paths) per LDAP Server" on page 250.
- 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 251.
- 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 245).

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).
- 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 257.

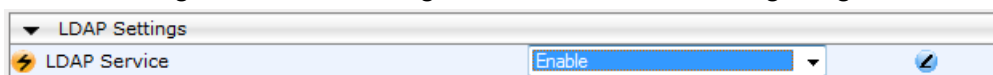
15.4.1 Enabling the LDAP Service

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 15-12: 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.

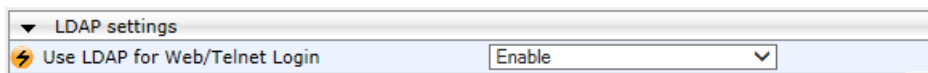
15.4.2 Enabling LDAP-based Web/CLI User Login Authentication and Authorization

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 15-13: Authentication Settings Page - Enabling LDAP-based Login



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.

15.4.3 Configuring LDAP Server Groups

The LDAP Server Groups table lets you configure up to 250 LDAP Server Groups. An LDAP Server Group is a logical configuration entity that contains up to two LDAP servers. LDAP servers are assigned to LDAP Server Groups in the LDAP Configuration table (see "Configuring LDAP Servers" on page 245). To use a configured LDAP server, you must assign it to an LDAP Server Group.

To use an LDAP server for call routing, you must configure its' LDAP Server Group as "Control" type, and then assign the LDAP Server Group to a Routing Policy. The Routing Policy in turn, needs to be assigned to the relevant routing rule(s). A Routing Policy can be assigned only one LDAP Server Group. Therefore, for multi-tenant deployments where multiple Routing Policies are employed, each tenant can be assigned a specific LDAP Server Group through its unique Routing Policy.

To use an LDAP server for management user login authentication and authorization, you must configure its' LDAP Server Group as "Management" type. Additional LDAP-based management parameters need to be configured, as described in "Enabling LDAP-based Web/CLI User Login Authentication and Authorization" on page 243 and "Configuring LDAP Servers" on page 245.

The following procedure describes how to configure an LDAP Server Group through the Web interface. You can also configure it through ini file (LDAPServersGroup) or CLI (configure voip/ldap/ldap-servers-group).

➤ **To configure an LDAP Server Group:**

1. Open the LDAP Server Groups table (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Server Groups**).

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

Figure 15-14: LDAP Server Groups Table - Add Row Dialog Box

The dialog box titled "Add Row" contains the following fields and values:

- Index: 12
- Name: (empty)
- Type: Control
- Server Search Method: Parallel
- Cache Entry Timeout [min]: 1200
- Cache Entry Removal Timeout [hrs]: 0
- DN Search Method: Sequential

Buttons: Add, Cancel

- Configure an LDAP Server Group according to the parameters described in the table below.
- Click **Add**.

Table 15-8: LDAP Server Groups Table Parameter Descriptions

Parameter	Description
Index [LdapServersGroup_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [LdapServersGroup_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 20 characters. Note: Each row must be configured with a unique name.
Type server-type [LdapServersGroup_ServerType]	Defines whether the servers in the group are used for SIP-related LDAP queries (Control) or management login authentication-related LDAP queries (Management). [0] Control (Default) [1] Management Note: Only one LDAP Server Group can be defined for management.
Server Search Method server-search-method [LdapServersGroup_SearchMethod]	Defines the method for querying between the two LDAP servers in the group. [0] Parallel = (Default) The device queries the LDAP servers at the same time. [1] Sequential = The device first queries one of the LDAP servers and if the DN object is not found or the search fails, it queries the second LDAP server.
Cache Entry Timeout cache-entry-timeout [LdapServersGroup_CacheEntryTimeout]	Defines the duration (in minutes) that an entry in the device's LDAP cache is valid. If the timeout expires, the cached entry is used only if there is no connectivity with the LDAP server. The valid range is 0 to 35791. The default is 1200. If set to 0, the LDAP entry is always valid.

Parameter	Description
Cache Entry Removal Timeout cache-entry-removal-timeout [LdapServersGroup_CacheEntryRemovalTimeout]	Defines the duration (in hours) after which the LDAP entry is deleted from the device's LDAP cache. The valid range is 0 to 596. The default is 0 (i.e., the entry is never deleted).
DN Search Method search-dn-method [LdapServersGroup_SearchDnsMethod]	Defines the method for querying the Distinguished Name (DN) objects within each LDAP server. ▪ [0] Sequential = (Default) The query is done in each DN object, one by one, until a result is returned. 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. ▪ [1] Parallel = The query is done in all DN objects at the same time. For example, a search for the DN object record "JohnD" is done at the same time in the "Marketing", "Sales" and "Administration" DN objects.

15.4.4 Configuring LDAP Servers

The LDAP Configuration table lets you configure up to 500 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 through the Web interface. You can also configure it through ini file (LdapConfiguration) or CLI (configure voip > ldap > ldap-configuration).



Note: When you configure an LDAP server, you need to assign it to an LDAP Server Group. Therefore, before you can configure an LDAP server in the table, you must first configure at least one LDAP Server Group in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 243).

➤ **To configure an LDAP server:**

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

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

Figure 15-15: LDAP Configuration Table - Add Row Dialog Box

The figure shows two instances of the 'Add Row' dialog box for LDAP configuration. Both dialogs have a title bar with 'Add Row' and a close button. Below the title bar is a yellow instruction bar that says 'Please select an LDAP Servers Group'. The main area contains a list of configuration fields:

- Index: 0 (top) / 1 (bottom)
- LDAP Servers Group: (empty dropdown)
- LDAP Server IP: 0.0.0.0
- LDAP Server Port: 389
- LDAP Server Max Respond Time [msec]: 3000
- LDAP Server Domain Name: (empty text field)
- LDAP Password: (empty text field)
- LDAP Bind DN: (empty text field)
- LDAP Network Interface: None (dropdown)
- Management Attribute: (empty text field)
- Use TLS: No (dropdown)
- TLS Context: None (dropdown)
- Verify Certificate: No (dropdown)
- Connection Status: (empty text field)

At the bottom right of each dialog are 'Add' and 'Cancel' buttons.

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

Table 15-9: LDAP Configuration Table Parameter Descriptions

Parameter	Description
Index [LdapConfiguration_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
LDAP Servers Group server-group [LdapConfiguration_Group]	Assigns the LDAP server to an LDAP Server Group, configured in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 243). Notes: - The parameter is mandatory and must be set before configuring the other parameters in the table. - Up to two LDAP servers can be assigned to the same LDAP Server Group.
LDAP Server IP 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. Notes: - The parameter is mandatory. - If you want to use an FQDN for the LDAP server, leave the parameter undefined and configure the FQDN in the 'LDAP Server Domain Name' parameter (see below).
LDAP Server Port 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 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 257.
LDAP Server Domain Name 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: If the 'LDAP Server IP' parameter is configured, the 'LDAP Server Domain Name' parameter is ignored. Thus, if you want to use an FQDN, leave the 'LDAP Server IP' parameter undefined.

Parameter	Description
LDAP Password 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, \$. Notes: - The parameter is mandatory. - 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).
LDAP Bind DN 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: The 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 the 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 interface-type [LdapConfiguration_Interface]	Assigns one of the device's IP network interfaces through which communication with the LDAP server is done. By default, no value is defined (None) and the device uses the OAMP network interface, configured in the Interface table. For configuring IP network interfaces, see "Configuring IP Network Interfaces" on page 138. Note: The parameter is mandatory.

Parameter	Description
Management Attribute 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 251. Notes: ▪ The 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 251.
Use TLS use-tls [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
TLS Context [LdapConfiguration_ContextName]	Assigns a TLS Context for the connection with the LDAP server. By default, no value is defined (None) and the device uses the default TLS Context (ID 0). For configuring TLS Contexts, see "Configuring TLS Certificate Contexts" on page 109. Note: The parameter is applicable only if the 'Use TLS' parameter is configured to Yes.
Verify Certificate verify-certificate [LdapConfiguration_VerifyCertificate]	Enables certificate verification when the connection with the LDAP server uses TLS. ▪ [0] No = (Default) No certificate verification is done. ▪ [1] Yes = The device verifies the authentication of the certificate received from the LDAP server. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the LDAP server. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context. Note: The parameter is applicable only if the 'Use TLS' parameter is configured to Yes.
Connection Status 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.

15.4.5 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 245) 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 through the Web interface. You can also configure it through ini file (LdapServersSearchDNs) or CLI (configure voip/ldap/ldap-servers-search-dns).

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

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

Figure 15-16: LDAP Search Base DN Table - Add Row Dialog Box

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

Table 15-10: LDAP Server Search Base DN Table Parameter Descriptions

Parameter	Description
Index set internal-index [LdapServersSearchDNs_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Base Path 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).

15.4.6 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 250.
- **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 245).

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 15-17: LDAP Settings Page - LDAP Search Filter

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

2. Make sure that the 'LDAP Service' parameter is configured to **Enable**.
3. In the 'LDAP Authentication Filter' parameter, enter the LDAP search filter attribute for searching the login username for user authentication.
4. Click **Submit**.

15.4.7 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 245) 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 245).

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" on page 67.

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 250.
- 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 through the Web interface. You can also configure it through ini file (MgmtLDAPGroups) or CLI (configure voip > ldap > mgmt-ldap-groups).

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

1. Open the LDAP Configuration table (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Configuration Table**).
2. In the 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** link, located below the table; the Management LDAP Groups table opens.
3. Click **Add**; the following dialog box appears:

Figure 15-18: Management LDAP Groups Table - Add Row Dialog Box

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

Table 15-11: Management LDAP Groups Table Parameter Descriptions

Parameter	Description
Index [MgmtLDAPGroups_GroupIndex]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Level Level [MgmtLDAPGroups_Level]	Defines the access level of the group(s). ▪ [0] Monitor (Default) ▪ [1] Admin ▪ [2] Security Admin
Groups 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 (;).

15.4.8 Configuring the Device's LDAP Cache

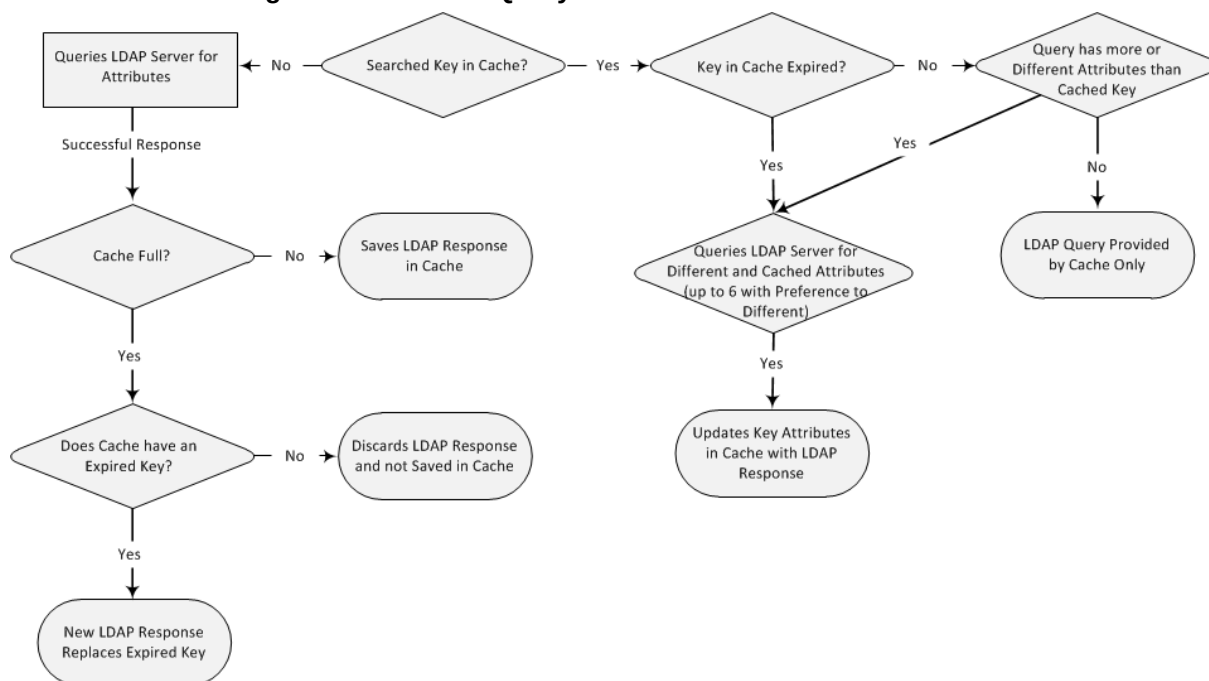
The device can optionally store LDAP queries of LDAP Attributes for a searched key with an LDAP server and the responses (results) in its local cache. The cache is used for subsequent queries and/or in case of LDAP server failure. The benefits of this feature include the following:

- 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 using the device's LDAP cache is shown in the flowchart below:

Figure 15-19: LDAP Query Process with Local LDAP Cache



If an LDAP query is required for an Attribute of a key that is already cached with that same Attribute, instead of sending a query to the LDAP server, the device uses the cache. However, if an LDAP query is required for an Attribute that does not appear for the cached key, the device queries the LDAP server and then saves the new Attribute (and response) in the cache for that key. When the device queries new Attributes for a cached key, the device also includes already cached Attributes of the key, while adhering to the maximum number of allowed saved Attributes (see note below), with preference to the new Attributes. In other words, if the cached key already contains the maximum Attributes and an LDAP query is required for a new Attribute, the device sends an LDAP query to the server for the new Attribute and for the five most recent Attributes already cached with the key. Upon the LDAP response, the new Attribute replaces the oldest cached Attribute while the values of the other Attributes are refreshed with the new response. The following table shows an example of different scenarios of LDAP queries of a cached key whose cached Attributes include a, b, c, and d, where a is the oldest and d the most recent Attribute:

Table 15-12: Example of LDAP Query for Cached Attributes

Attributes Requested in New LDAP Query for Cached Key	Attributes Sent in LDAP Query to LDAP Server	Attributes Saved in Cache after LDAP Response
e	e, a, b, c, d	e, a, b, c, d
e, f	e, f, a, b, c, d	e, f, a, b, c, d
e, f, g, h, i	e, f, g, h, i, a	e, f, g, h, i, a
e, f, g, h, i, j	e, f, g, h, i, j	e, f, g, h, i, j

**Note:**

- The LDAP Cache feature is applicable only to LDAP-based SIP queries (Control).
- The maximum LDAP cache size is 10,000 entries.
- The device can save up to six LDAP Attributes in the cache per user (search LDAP key).
- The device also saves in the cache queried Attributes that do not have any values in the LDAP server.

The following procedure describes how to configure the device's LDAP cache through the Web interface. For a full description of the cache parameters, see 'LDAP Parameters' on page 828.

➤ **To enable and configure the LDAP cache:**

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

Figure 15-20: LDAP Settings Page - Cache Parameters

LDAP Cache	
LDAP Cache Service	Enable
LDAP Cache Entry Timeout	1200
LDAP Cache Entry Removal Timeout	0

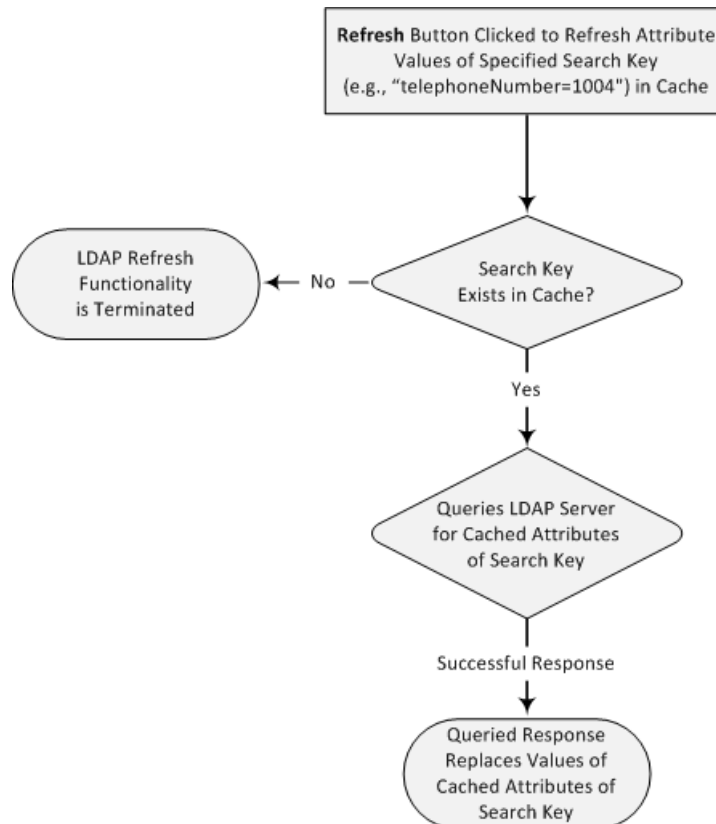
2. Under the Cache group, do the following:
 - a. From the 'LDAP Cache Service' drop-down list, select **Enable** to enable LDAP cache.
 - b. In the 'LDAP Cache Entry Timeout' field, enter the duration (in minutes) for which an entry in the LDAP cache is valid.
 - c. In the 'LDAP Cache Entry Removal Timeout' field, enter the duration (in hours) after which the device removes the LDAP entry from the cache.
3. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

15.4.8.1 Refreshing the LDAP Cache

You can refresh values of LDAP Attributes associated with a specified LDAP search key that are stored in the device's LDAP cache. The device sends an LDAP query to the LDAP server for the cached Attributes of the specified search key and replaces the old values in the cache with the new values received in the LDAP response.

For example, assume the cache contains a previously queried LDAP Attribute "telephoneNumber=1004" whose associated Attributes include "displayName", "mobile" and "ipPhone". If you perform a cache refresh based on the search key "telephoneNumber=1004", the device sends an LDAP query to the server requesting values for the "displayName", "mobile" and "ipPhone" Attributes of this search key. When the device receives the LDAP response, it replaces the old values in the cache with the new values received in the LDAP response.

Figure 15-21: LDAP Cache Refresh Flowchart



➤ **To refresh the LDAP cache per LDAP Server Group:**

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

Figure 15-22: Refreshing LDAP Cache

LDAP Cache Actions	
LDAP Group Index	0
LDAP Refresh Cache by Key	Refresh
LDAP Clear Cache	Clear Group

2. Under the Cache Actions group, do the following:
 - a. From the 'LDAP Group Index' drop-down list, select the required LDAP Server Group (see 'Configuring LDAP Server Groups' on page 243).
 - b. In the 'LDAP Refresh Cache by Key' field, enter the LDAP search key that you want to refresh (e.g., telephoneNumber=1004).
 - c. Click **Refresh**; if a request with the specified key exists in the cache, a request is sent to the LDAP server for the Attributes associated in the cache with the search key.

15.4.8.2 Clearing the LDAP Cache

You can remove (clear) all LDAP entries in the device's LDAP cache for a specific LDAP Server Group, as described in the following procedure.

➤ **To clear the LDAP cache:**

1. Open the LDAP Settings page (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **LDAP Settings**).
2. Under the Cache Actions group, do the following:
 - a. From the 'LDAP Group Index' drop-down list, select the required LDAP Server Group (see 'Configuring LDAP Server Groups' on page 243).
 - b. Click **Clear Group**.

15.4.9 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" on page 67.



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 15-23: 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 if a server is configured but connectivity with the server is down (if the server is up, the device authenticates the user with the server).
 - ♦ **Always:** First attempts to authenticate the user using the Web Users table, but if not found, it authenticates the user with the LDAP/RADIUS server.
 - 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**.

15.4.10 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 15-24: Base Path (DN) in LDAP Server

Path: CN=John Doe,OU=testMgmt,OU=QA,DC=testqa,DC=local, 10.3.9.93 [testqa.testqa.local]

Active Directory Explorer

10.3.9.93 [testqa.testqa.local]

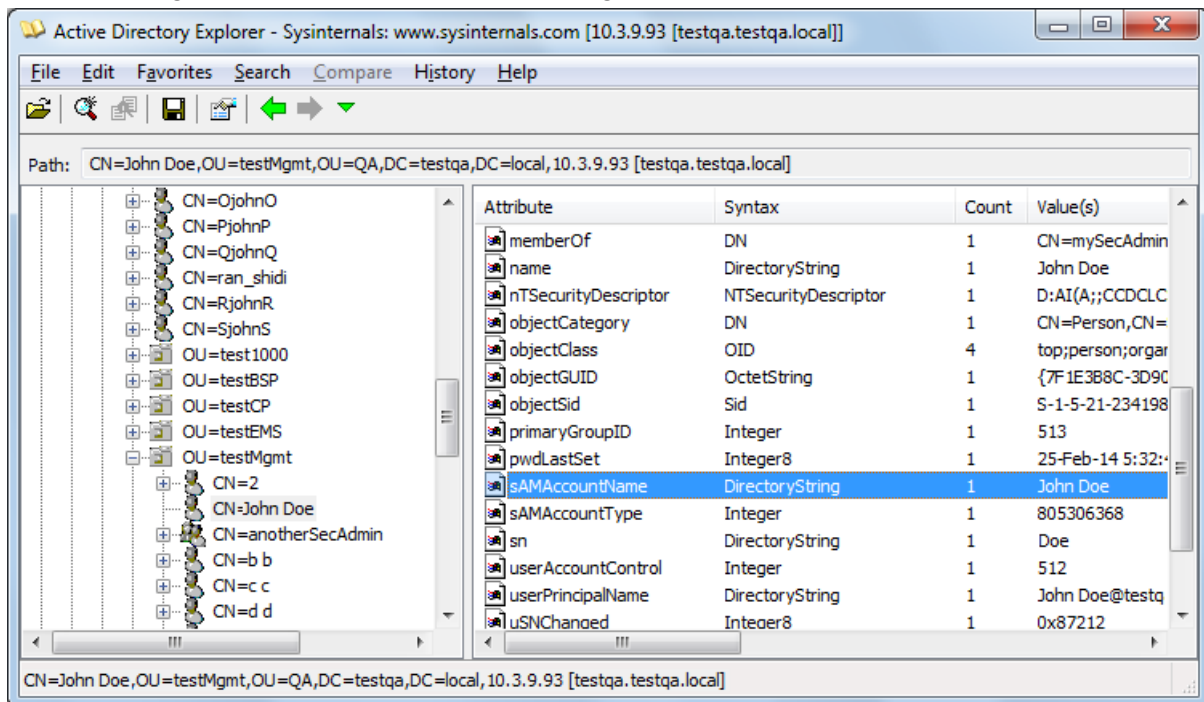
- DC=testqa,DC=local
 - CN=Builtin
 - CN=Computers
 - CN=Deleted Objects
 - OU=Domain Controllers
 - CN=ForeignSecurityPrincipals
 - CN=Infrastructure
 - CN=LostAndFound
 - CN=NTDS Quotas
 - CN=Program Data
 - OU=QA
 - CN=Aaapaul50digitsL
 - CN=Aaapaul51digitsL
 - CN=AjohnA
 - CN=BjohnB
 - CN=CjohnC
 - CN=DjohnD
 - CN=EjohnE
 - CN=Firstaaaa Lastbbbb
 - CN=FjohnF
 - CN=George Harrison
 - CN=GjohnG
 - CN=HjohnH
 - CN=IjohnI
 - CN=JjohnJ
 - CN=John Doe
 - CN=John Doe Bind
 - CN=KjohnK
 - CN=LjohnL
 - OU=Misc
 - CN=MjohnM
 - CN=NjohnN
 - CN=OjohnO
 - CN=PjohnP
 - CN=QjohnQ
 - CN=ran_shidi
 - CN=RjohnR
 - CN=SjohnS
 - OU=test1000
 - OU=testBSP
 - OU=testCP
 - OU=testEMS
 - OU=testMgmt
 - CN=2
 - CN=John Doe**
 - CN=anotherSecAdmin
 - CN=b b
 - CN=c c
 - CN=d d

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;;CCDCLCSWRPWPDP
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
uSNCChanged	Integer8	1	0x87212
uSNCreated	Integer8	1	0x8311F
whenChanged	GeneralizedTime	1	25-Feb-14 5:32:45 PM
whenCreated	GeneralizedTime	1	06-Oct-02 5:27:51 AM

CN=John Doe,OU=testMgmt,OU=QA,DC=testqa,DC=local, 10.3.9.93 [testqa.testqa.local]

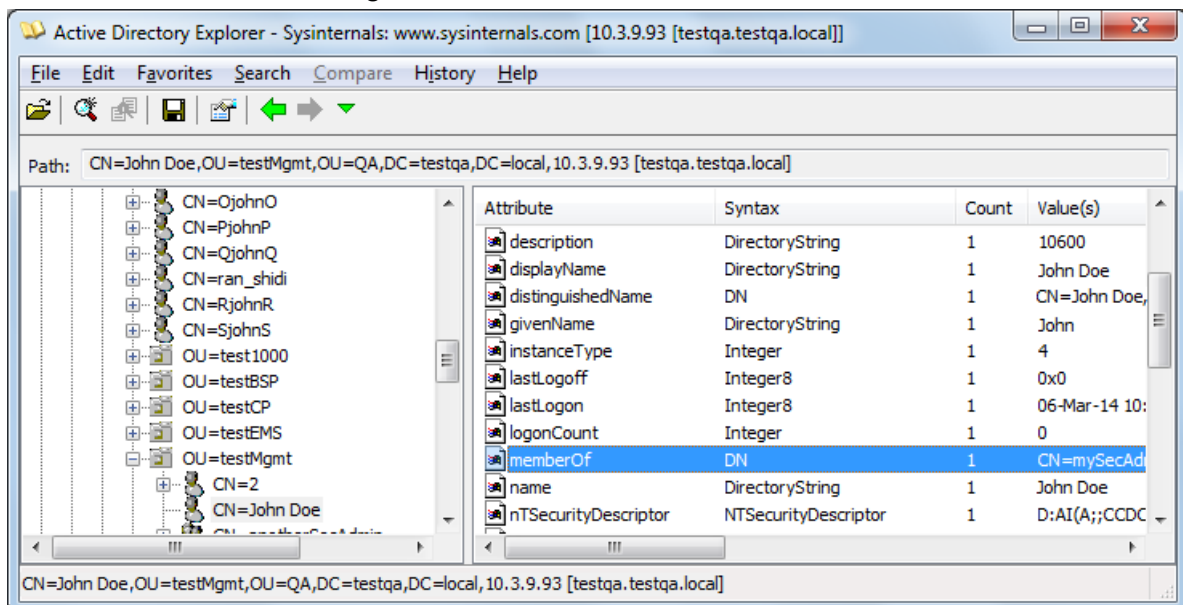
- **Search Attribute Filter:** (sAMAccountName=\$). The login username is found based on this attribute (where the attribute's value equals the username):

Figure 15-25: Username Found using sAMAccount Attribute Search Filter



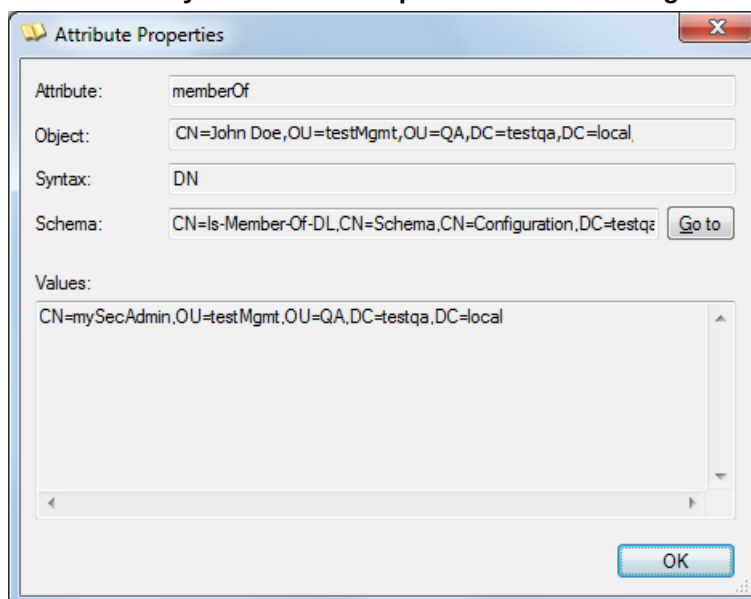
- **Management Attribute:** memberOf. The attribute contains the member groups of the user:

Figure 15-26: User's memberOf Attribute



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

Figure 15-27: User's mySecAdmin Group in memberOf Management Attribute



The 'Attribute Properties' dialog box displays the following information:

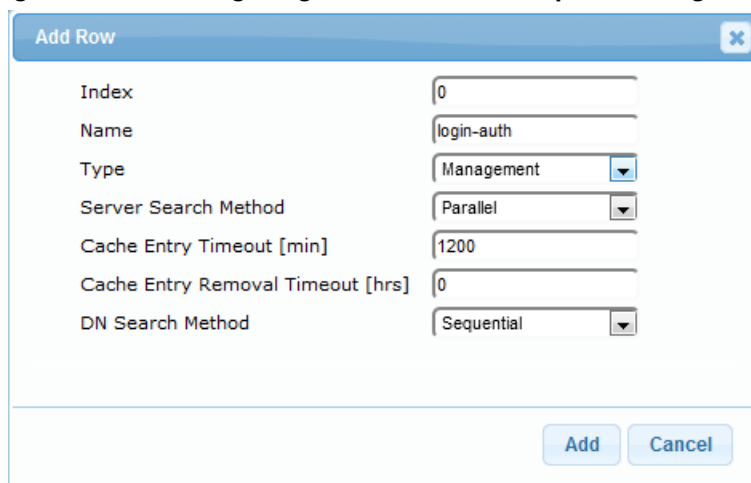
- Attribute:** memberOf
- Object:** CN=John Doe,OU=testMgmt,OU=QA,DC=testqa,DC=local
- Syntax:** DN
- Schema:** CN=ls-Member-Of-DL,CN=Schema,CN=Configuration,DC=testqa (with a 'Go to' button)
- Values:** A list box containing the value: CN=mySecAdmin,OU=testMgmt,OU=QA,DC=testqa,DC=local

An 'OK' button is located at the bottom right of the dialog.

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

- LDAP-based login authentication (management) is enabled in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 243):

Figure 15-28: Configuring LDAP Server Group for Management



The 'Add Row' dialog box contains the following configuration fields:

Field	Value
Index	0
Name	login-auth
Type	Management
Server Search Method	Parallel
Cache Entry Timeout [min]	1200
Cache Entry Removal Timeout [hrs]	0
DN Search Method	Sequential

'Add' and 'Cancel' buttons are located at the bottom right of the dialog.

- The DN is configured in the LDAP Server Search Base DN table (see "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 250):

Figure 15-29: 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 251):

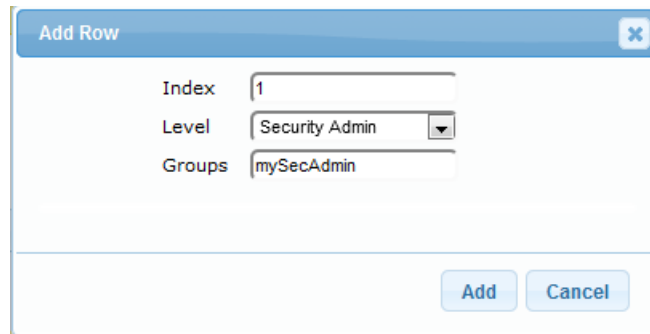
Figure 15-30: Configuring Search Attribute Filter

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

Figure 15-31: Configuring Management Attribute

- 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 251):

Figure 15-32: Configuring Management Group Attributes for Determining Access Level



Add Row	
Index	1
Level	Security Admin
Groups	mySecAdmin

Add Cancel

15.4.11 Enabling LDAP Searches for Numbers with Characters

Typically, the device performs LDAP searches in the AD for complete numbers where the digits are adjacent to one another (e.g., 5038234567). However, if the number is defined in the AD with characters (such as spaces, hyphens and periods) separating the digits (e.g., 503-823 4567), the LDAP query returns a failed result.

To enable the device to search the AD for numbers that may contain characters between its digits, you need to specify the Attribute (up to five) for which you want to apply this functionality, using the `LDAPNumericAttributes` parameter. For example, the `telephoneNumber` Attribute could be defined in AD with the telephone number "503-823-4567" (i.e., hyphens), "503.823.4567" (i.e., periods) or "503 823 4567" (i.e., spaces). If the device performs an LDAP search on this Attribute for the number 5038234567, the LDAP query will return results only if you configure the `LDAPNumericAttributes` parameter with the `telephoneNumber` Attribute (e.g., `LDAPNumericAttributes=telephoneNumber`). To search for the number with characters, the device inserts the asterisk (*) wildcard between all digits in the LDAP query (e.g., `telephoneNumber = 5*0*3*8*2*3*4*5*6*7`). As the AD server recognizes the * wildcard as representing any character, it returns all possible results to the device. Note that the wildcard represents only a character; a query result containing a digit in place of a wildcard is discarded and the device performs another query for the same Attribute. For example, it may return the numbers 533-823-4567 (second digit "3" and hyphens) and 503-823-4567. As the device discards query results where the wildcard results in a digit, it selects 503-823-4567 as the result. The correct query result is cached by the device for subsequent queries and/or in case of LDAP server failure.

15.4.12 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)

15.4.12.1 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:

Table 15-13: Parameters for Configuring Query Attribute Key

Parameter	Queried User Domain (Attribute) in AD	Query or Query Result Example
MSLDAPPBXNumAttribute Name	PBX or IP PBX number (e.g., "telephoneNumber" - default)	telephoneNumber=+3233554447
MSLDAPOCSNumAttribute Name	Mediation Server / Lync client number (e.g., "msRTCSIP-line")	msRTCSIP-line=john.smith@company.com
MSLDAPMobileNumAttribute Name	Mobile number (e.g., "mobile")	mobile=+3247647156
MSLDAPPrivateNumAttribute Name	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 Tel-to-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



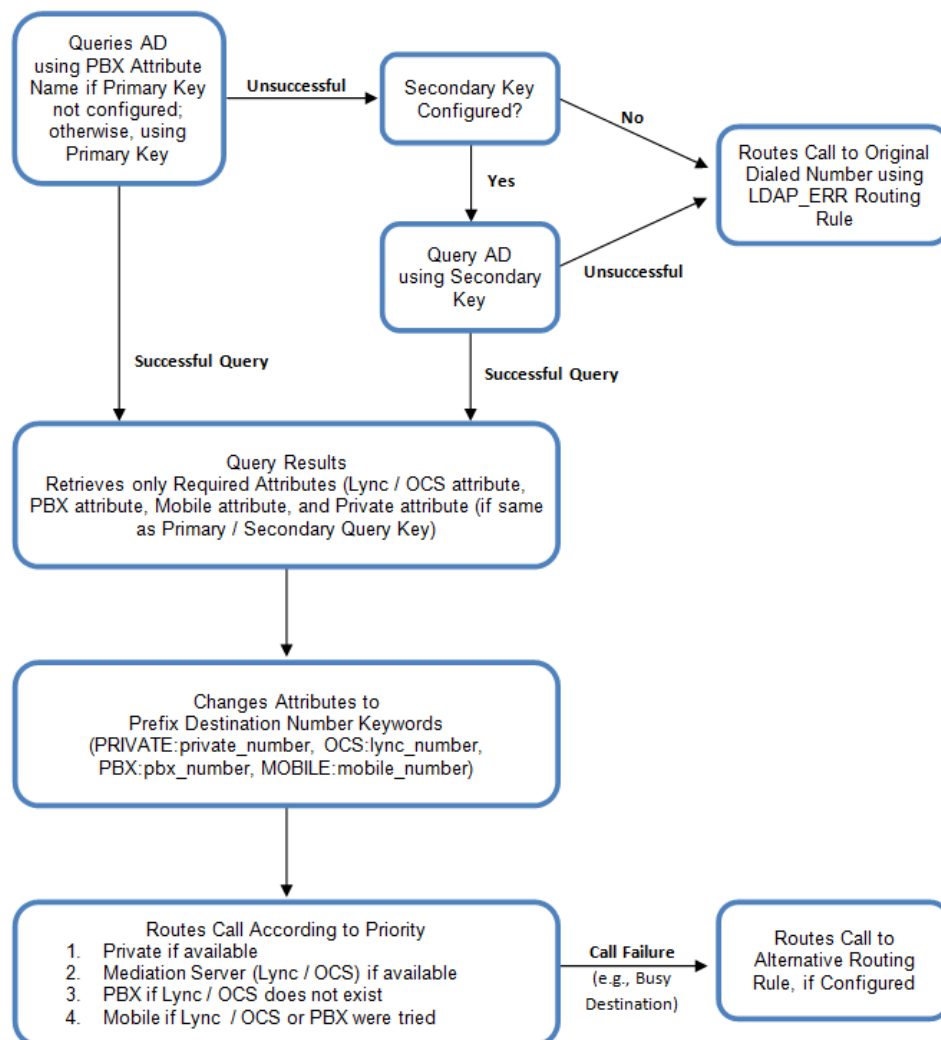
Note: These prefixes are involved only in the routing and manipulation processes; they are not used as the final destination number.

7. The device uses the Tel-to-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 15-33: LDAP Query Flowchart



Note: If you are using the device's local LDAP cache, see "Configuring the Device's LDAP Cache" on page 253 for the LDAP query process.

15.4.12.2 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 245.
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 15-34: 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 query-result routing rules for each IP domain (private, PBX / IP PBX, Lync clients, and mobile), using the LDAP keywords (case-sensitive) for the prefix destination number:
4. Configure AD-based IP-to-IP routing rules:
 - a. Open the IP-to-IP Routing table (Configuration tab > VoIP menu > SBC > Routing SBC > IP-to-IP Routing Table). For more information, see Configuring SBC IP-to-IP Routing Rules on page 480.
 - 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:

Table 15-14: 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
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 on page 491) 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.

15.5 Least Cost Routing

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

15.5.1 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 the routing table for matching routing rules and then selects the rule with the lowest call cost. If two routing rules have identical costs, the rule appearing higher up in the table is used (i.e., first-matched rule). If the selected route is unavailable, the device selects the next least-cost routing rule.

Even if a matched routing rule is not assigned a Cost Group, the device can select it as the preferred route over other matched rules that are assigned Cost Groups. This is determined according to the settings of the 'Default Call Cost' parameter configured for the Routing Policy (associated with the routing rule). For configuring the Routing Policy, see Configuring SBC Routing Policy Rules on page 493.

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:

$$\text{Total Call Cost} = \text{Connection Cost} + (\text{Minute Cost} * \text{Average Call Duration})$$

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 15-15: 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 rule.

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 Tel-to-IP Routing table:

The 'Default Call Cost' parameter in the Routing Policy rule is configured to **Lowest Cost**, meaning that if the device locates other matching 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 Call Cost' parameter is configured to **Lowest Cost**, 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 15-35: 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

15.5.2 Configuring LCR

To configure LCR, perform the following main steps:

1. Enable LCR - see Configuring SBC Routing Policy Rules on page 493.
2. Configure Cost Groups - see "Configuring Cost Groups" on page 271.
3. Configure Time Bands for a Cost Group - see "Configuring Time Bands for Cost Groups" on page 272.
4. Assign Cost Groups to outbound IP routing rules - see "Assigning Cost Groups to Routing Rules" on page 274.

15.5.2.1 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 through the Web interface. You can also configure it through ini file (CostGroupTable) or CLI (configure voip > services least-cost-routing cost-group).

➤ **To configure a Cost Group:**

1. Open the Cost Group table (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Cost Group Table**).
2. Click **Add**; the following dialog box appears:

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

Table 15-16: Cost Group Table Parameter Descriptions

Parameter	Description
Index [CostGroupTable_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name cost-group-name [CostGroupTable_CostGroupName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. Note: Each Cost Group must have a unique name.
Default Connection Cost 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 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.

15.5.2.2 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 through the Web interface. You can also configure it through ini file (CostGroupTimebands) or CLI (configure voip >services least-cost-routing cost-group-time-bands).

➤ **To configure a Time Band per Cost Group:**

1. Open the Cost Group table (**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:

The dialog box titled "Add Row" contains the following fields:

- Index:** A dropdown menu with a small 'x' icon on the right.
- Start Time (ddd:hh:mm):** A text input field.
- End Time (ddd:hh:mm):** A text input field.
- Connection Cost:** A text input field with the value '0'.
- Minute Cost:** A text input field with the value '0'.

At the bottom right of the dialog are two buttons: **Add** and **Cancel**.

4. Configure a Time Band according to the parameters described in the table below.
5. Click **Add**, and then save ("burn") your settings to flash memory.

Table 15-17: Time Band Table Description

Parameter	Description
Index timeband-index [CostGroupTimebands_TimebandIndex]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Start Time 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: ▪ <i>DDD</i> 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). ▪ <i>hh</i> and <i>mm</i> denote the time of day, where <i>hh</i> is the hour (00-23) and <i>mm</i> the minutes (00-59) For example, SAT:22:00 denotes Saturday at 10 pm.
End Time 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 connection-cost [CostGroupTimebands_ConnectionCost]	Defines the call connection cost during this time band. This is added as a fixed charge to the call. The valid value range is 0-65533. The default is 0. Note: The entered value must be a whole number (i.e., not a decimal).

Parameter	Description
Minute Cost 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).

15.5.2.3 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 480

15.6 HTTP-based Remote Services

15.6.1 Configuring HTTP Services

The HTTP Remote Services table lets you configure up to seven HTTP-based services provided by third-party remote hosts (e.g., routing server). The following types of services can be offered by the remote host:

- **Routing:** Call routing service, whereby the remote host (e.g., routing server) determines the next hop of an incoming call on the path to the final destination. For more information on employing a third-party, remote routing server, see "Centralized Third-Party Routing Server" on page 280.
- **Call Status:** Call status of calls processed by the device. The call status is provided to the remote host through CDRs sent by the device.
- **Topology Status:** Status of device configuration (add, edit and delete). The device sends topology status to the HTTP host, using the REST TopologyStatus API command. To enable the functionality, configure the 'Topology Status' (RoutingServerGroupStatus) parameter to **Enable**. The parameter is located below the table.

Topology status includes the following:

- IP Groups: status is reported when the keep-alive mechanism (enabled for the associated Proxy Set) detects that the IP Group is unavailable, or when CAC thresholds (configured in the Admission Control table) are crossed.
- Status is reported when IP Groups or SIP Interfaces that are configured to be used by HTTP-based services (i.e., the UsedByRoutingServer parameter is set to 1 - Used) are created or deleted. If you subsequently change the settings of the UsedByRoutingServer parameter or the 'Name' parameter, the device reports the change as a creation or deletion of the corresponding configuration entity.
- **Capture:** Recording of signaling and RTP packets, and Syslog. The remote host can be, for example, a Syslog server or AudioCodes SEM.



Notes:

- You can configure only **one** HTTP Remote Service entry for Routing, for Call Status, and for Topology. However, you can configure up to four HTTP Remote Services for Capture.
- The Routing service also includes the Call Status and Topology Status services.
- Currently, the Capture service is not supported.
- The device supports HTTP redirect responses (3xx) only during connection establishment with the host. Upon receipt of a redirect response, the device attempts to open a new socket with the host and if this is successful, closes the current connection.

The following procedure describes how to configure HTTP Remote Services through the Web interface. You can also configure it through ini file (HTTPRemoteServices).

➤ **To configure an HTTP-based service**

1. Open the HTTP Remote Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Services** > **HTTP Remote Services**).
2. Click **Add**; the following dialog box appears:

Figure 15-36: HTTP Remote Services Table - Add Row Dialog Box

The 'Add Row' dialog box is shown with the following parameters:

Parameter	Value
Index	0
Name	
Path	api
Type	Routing
Policy	Round Robin
Login Needed	Enable
Persistent Connection	Enable
Number of Sockets	1
Username	user
Password	
TLS Context	None
Verify Certificate	Disable
Response Timeout [sec]	5
Keep-Alive Timeout [sec]	0
Status	

3. Configure an HTTP remote service according to the parameters described in the table below.
4. Click **Add**, and then save ("burn") your settings to flash memory.

Table 15-18: HTTP Remote Services Table Parameter Descriptions

Parameter	Description
Index [HTTPRemoteServices_Index]	Defines an index number for the new table row. Notes: - Each row must be configured with a unique index. - The parameter is mandatory.
Name [HTTPRemoteServices_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. Notes: - Each row must be configured with a unique name. - The parameter is mandatory.
Path [HTTPRemoteServices_Path]	Defines the path (prefix) to the REST APIs. The valid value is a string of up to 80 characters. The default is "api".

Parameter	Description
Type [HTTPRemoteServices_HT TPTYPE]	Defines the type of service provided by the HTTP remote host: [0] Routing (default) = Routing service (also includes Call Status and Topology Status). [1] Call Status = Call status service. [2] Topology Status = Topology status service (e.g., change in configuration). [3] Capture = Recording of signaling and RTP packets, which can be sent to a remote host, for example, to a Syslog server or AudioCodes SEM. Notes: - You can only configure one HTTP service for each of the following service types: Routing, Call Status and Topology Status. - For the Topology Status option to be functional, you must enable the RoutingServerGroupStatus parameter. - Currently, the Capture option is not supported.
Policy [HTTPRemoteServices_Pol icy]	Defines the mode of operation when you have configured multiple remote hosts (in the HTTP Remote Hosts table) for a specific HTTP service. [0] Round Robin = (Default) Load balancing of traffic across all configured hosts. Every consecutive message is sent to the next available host. [1] Sticky Primary = Device always attempts to send traffic to the first (primary) host. If the host does not respond, the device sends the traffic to the next available host. If the primary host becomes available again, the device sends the traffic to the primary host. [2] Sticky Next = Similar to Sticky Primary, but if the primary host does not respond, the device sends the traffic to the next available host and continues sending traffic to this host even if the primary host becomes available again.
Login Needed [HTTPRemoteServices_Lo ginNeeded]	Enables the use of proprietary REST API Login and Logout commands for connecting to the remote host. The commands verify specific information (e.g., software version) before allowing connectivity with the device. [0] Disable = Commands are not used. [1] Enable (default)
Persistent Connection [HTTPRemoteServices_Pe rsistentConnection]	Defines whether the HTTP connection with the host remains open or is only opened per request. [0] Disable = Connection is not persistent and closes when the device detects inactivity. The device uses HTTP keep-alive messages to detect inactivity. [1] Enable = (Default) Connection remains open (persistent) even during inactivity. The device uses HTTP keep-alive / HTTP persistent connection messages to keep the connection open.
Number of Sockets [HTTPRemoteServices_Nu mOfSockets]	Defines how many sockets (connection) are established per remote host. The valid value is 1 to 10. The default is 1.
Username [HTTPRemoteServices_Aut hUserName]	Defines the username for HTTP authentication. The valid value is a string of up to 80 characters. The default is "user".

Parameter	Description
Password [HTTPRemoteServices_AuthPassword]	Defines the password for HTTP authentication. The valid value is a string of up to 80 characters. The default is "password".
TLS Context [HTTPRemoteServices_TLSContext]	Assigns a TLS Context for the connection with the HTTP service. By default, no value is defined (None). For configuring TLS Contexts, see "Configuring TLS Certificate Contexts" on page 109. Note: The parameter is applicable only if the connection is HTTPS.
Verify Certificate [HTTPRemoteServices_VerifyCertificate]	Enables certificate verification when the connection with the host is based on HTTPS. [0] Disable (default) = No certificate verification is done. [1] Enable = The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context. Note: The parameter is applicable only if the connection is HTTPS.
Response Timeout [HTTPRemoteServices_TimeOut]	Defines the TCP response timeout (in seconds) from the remote host. If one of the remote hosts does not respond to a request within the specified timeout, the device closes the corresponding socket and attempts to connect to the next remote host. The valid value is 1 to 65535. The default is 5.
Keep-Alive Timeout [HTTPRemoteServices_KeepAliveTimeOut]	Defines the duration/timeout (in seconds) in which HTTP-REST keep-alive messages are sent by the device if no other messages are sent. Keep-alive messages may be required for HTTP services that expire upon inactive sessions. The valid value is 0 to 65535. The default is 0 (i.e., no keep-alive messages are sent). Note: The parameter is applicable only if the 'Persistent Connection' parameter (in the table) is configured to Enable .
Topology Status [HTTPRemoteServices_ServiceStatus]	Indicates the status of the host. "Connected": at least one of the hosts is connected. "Disconnected": all hosts are disconnected. "Not In Service": Configuration of the service is invalid.

15.6.2 Configuring Remote HTTP Hosts

The HTTP Remote Hosts table lets you configure up to 10 remote HTTP hosts per HTTP Remote Service. The HTTP Remote Hosts table is a "child" of the HTTP Remote Services table (configured in "Configuring HTTP Services" on page 275).

The following procedure describes how to configure HTTP Remote hosts through the Web interface. You can also configure it through ini file (HTTPRemoteServices).

➤ **To configure an HTTP-based service**

1. Open the HTTP Remote Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Services** > **HTTP Remote Services**).
2. In the table, select the required HTTP Remote Service index row, and then click the **HTTP Remote Hosts** button, located below the table; the HTTP Remote Hosts page appears.
3. Click **Add**; the following dialog box appears:

Figure 15-37: HTTP Remote Hosts Table - Add Row Dialog Box

4. Configure an HTTP remote host according to the parameters described in the table below.
5. Click **Add**, and then save ("burn") your settings to flash memory.

Table 15-19: HTTP Remote Hosts Table Parameter Descriptions

Parameter	Description
Index [HTTPRemoteHosts_RemoteHostindex]	Defines an index number for the new table row. Notes: - Each row must be configured with a unique index. - The parameter is mandatory.
Name [HTTPRemoteHosts_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no value is defined. Notes: - Each row must be configured with a unique name. - The parameter is mandatory.
Address [HTTPRemoteHosts_Address]	Defines the address (IP address or FQDN) of the host. The valid value is a string of up to 80 characters. Notes: - An IPv6 address can only be configured if the interface is a CONTROL type. - If the address is an FQDN and the DNS resolution results in multiple IP addresses, the device device attempts to establish multiple connections (sessions) for each IP address. Only the first 10 resolved IP addresses are used regardless of the number of hosts.

Parameter	Description
	- FQDN resolution is also performed (immediately) when connection is subsequently "closed" (by timeout or by the remote host) and connections are updated accordingly. In addition, the device periodically (every 15 minutes) performs DNS name resolution to ensure that the list of resolved IP addresses has not changed. If a change is detected, the device updates its' list of IP addresses and re-establishes connections accordingly. - In addition to multiple HTTP sessions, the device establishes multiple (TCP) connections per session, thereby enhancing data exchange capabilities with the host.
Port [HTTPRemoteHosts_Port]	Defines the port of the host. The valid value is 0 to 65535. The default is 80.
Interface [HTTPRemoteHosts_Interface]	Assigns one of the device's IP network interfaces through which communication with the remote host is done. By default, no value is defined and the OAMP interface is used.
Transport Type [HTTPRemoteHosts_HTTPTransportType]	Defines the protocol used for communicating with the host: [0] HTTP (default) [1] HTTPS
Status	Read-only field displaying the status of the connection. "Connected": The hosts is connected. "Disconnected": The host is disconnected. "Not In Service": Configuration of the host is invalid.

15.6.3 Centralized Third-Party Routing Server or ARM

You can employ a remote, third-party Routing server or AudioCodes Routing Manager (ARM) routing server to handle call routing decisions in deployments consisting of multiple AudioCodes devices. Employing a routing server replaces the need for the device's routing tables (IP-to-IP Routing table) to determine call destination.

When the device receives an incoming call (SIP INVITE, NOTIFY or MESSAGE), it searches the IP-to-IP Routing table for a matching routing rule that is also configured to use a routing server. If found, the device requests the routing server for an appropriate destination. The request is sent to the routing server using an HTTP Get Route message. The request contains information about the call (SIP message).

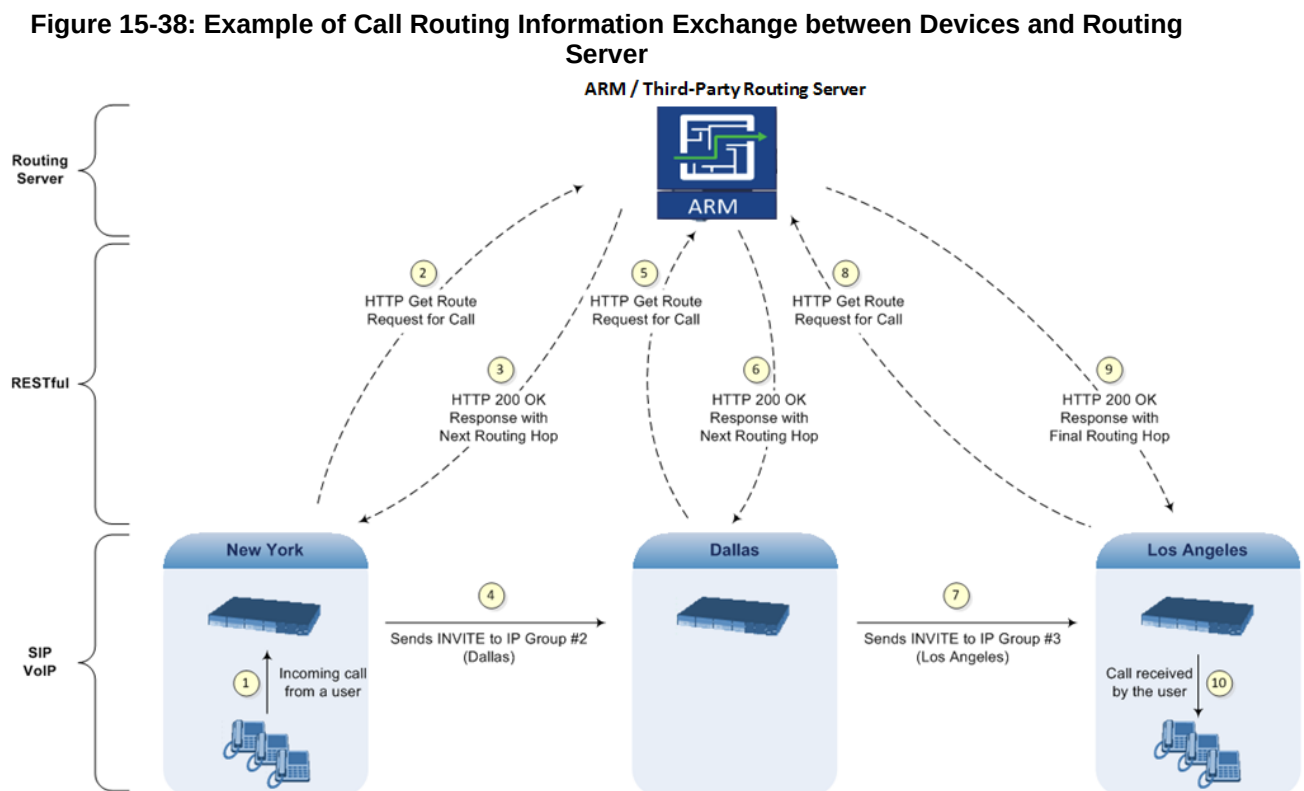
The routing server uses its own algorithms and logic in determining the best route path. The routing server manages the call route between devices in "hops", which may be spread over different geographical locations. The destination to each hop (device) can be by IP address (with port) or IP Group. If the destination is an IP address, even though the destination type (in the IP-to-IP Routing table) is an IP Group, the device only uses the IP Group for profiling (i.e., associated IP Profile etc.). If multiple devices exist in the call routing path, the routing server sends the IP address only to the last device ("node") in the path.

Once the device receives the resultant destination hop from the routing server, it sends the call to that destination. The routing server can provide the device with an appropriate route or reject the call. However, if for the initial request (first sent Get Route request for the call) the routing server cannot find an appropriate route for the call or it does not respond, for example, due to connectivity loss (i.e., the routing server sends an HTTP 404 "Not Found" message), the device routes the call using its routing tables. If the Get Route request is not the first one sent for the call (e.g., in call forwarding or alternative routing) and the routing server responds with an HTTP 404 "Not Found" message, the device rejects the call.

This HTTP request-response transaction for the routing path occurs between routing server and each device in the route path (hops) as the call traverses the devices to its final destination. Each device in the call path connects to the routing server, which responds with the next hop in the route path. Each device considers the call as an incoming call from an IP Group. The session ID (SID) is generated by the first device in the path and then passed unchanged down the route path, enabling the routing server to uniquely identify requests belonging to the same call session.

Communication between the device and the routing server is through the device's embedded Representational State Transfer (RESTful) API. The RESTful API is used to manage the routing-related information exchanged between the routing server (RESTful server) and the device (RESTful client). When you have configured the device with connection settings of the Routing sever and the device starts-up, it connects to the routing server and activates the RESTful API, which triggers the routing-related API commands.

The following figure provides an example of information exchange between devices and a routing server for routing calls:



The routing server can also manipulate call data such as calling name, if required. It can also create new IP Groups and associated configuration entities, if necessary for routing. Multiple routing servers can also be employed, whereby each device in the chain path can use a specific routing server. Alternatively, a single routing server can be employed and used for all devices ("stateful" routing server).

The device automatically updates (sends) the routing server with its' configuration topology regarding SIP routing-related entities (SRDs, SIP Interfaces, and IP Groups) that have been configured for use by the routing server. For example, if you add a new IP Group and enable it for use by the routing server, the device sends this information to the routing server. Routing of calls associated with routing-related entities that are disabled for use by the routing server (default) are handled only by the device (not the routing server).

In addition to regular routing, the routing server functionality also supports the following:

- **Alternative Routing:** If a call fails to be established, the device "closest" to the failure and configured to send "additional" routing requests (through REST API - "additionalRoute" attribute in HTTP Get Route request) to the routing server, sends a new routing request to the routing server. The routing server may respond with a new route destination, thereby implementing alternative routing. Alternatively, it may enable the device to return a failure response to the previous device in the route path chain and respond with an alternative route to this device. Therefore, alternative routing can be implemented at any point in the route path. If the routing server sends an HTTP 404 "Not Found" message for an alternative route request, the device rejects the call. If the routing server is configured to handle alternative routing, the device does not make any alternative routing decisions based on its alternative routing tables.
- **Call Status:** The device can report call status to the routing server to indicate whether a call has successfully been established and/or failed (disconnected). The device can also report when an IP Group (Proxy Set) is unavailable, detected by the keep-alive mechanism, or when the CAC thresholds permitted per IP Group have been crossed. For Trunk Groups, the device reports when the trunk's physical state indicates that the trunk is unavailable.
- **Credentials for Authentication:** The Routing Server can provide user (e.g., IP Phone caller) credentials (username-password) in the Get Route response, which can be used by the device to authenticate outbound SIP requests if challenged by the outbound peer, for example, Microsoft Skype for Business (per RFC 2617 and RFC 3261). If multiple devices exist in the call routing path, the routing server sends the credentials only to the last device ("node") in the path.

➤ **To configure routing based on Routing server:**

1. For each configuration entity (e.g., IP Group) that you want routing done by the Routing server, configure the entity's 'Used By Routing Server' parameter to **Used**.
2. Configure an additional Security Administrator user account in the Local Users table (see "Configuring Web User Accounts" on page 67), which is used by the Routing server (REST client) to log in to the device's management interface.
3. Configure the address and connection settings of the Routing server, referred to as a Remote Web Service and HTTP remote host. You must configure the 'Type' parameter of the Remote Web Service to **Routing**.
4. In the IP-to-IP Routing table, configure the 'Destination Type' parameter of the routing rule to **Routing Server** (see Configuring SBC IP-to-IP Routing Rules on page 480).

15.7 HTTP-based Proxy Services

You can configure the device for the following HTTP-based proxy services:

■ HTTP Reverse Proxy for Managing Equipment behind NAT:

You can configure the device to function as a reverse HTTP proxy server. This functionality is required to enable administrators to manage communication equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and the administrator is located in a public domain (e.g., in the WAN). Thus, this functionality resolves NAT issues, enabling the administrator to access the IP Phone's management interface (e.g., embedded Web server).

To support the functionality, the following configuration is required:

1. Enable the HTTP Proxy application (see 'Enabling the HTTP Proxy Application' on page 284).
2. Define a local, listening HTTP interface for the leg interfacing with the administrator (see 'Configuring HTTP Interfaces' on page 284).



Note: It is recommended **not** to use port 80 as this is the default port used by IP Phones for their Web-based management interface.

3. Define each HTTP-based managed equipment:
 - a. Define the URL prefix for accessing the equipment's management interface (see 'Configuring HTTP Proxy Services' on page 286). To access the equipment's management interface, the administrator needs to enter the following URL in a Web browser:
http://<device's WAN IP address:port>/url prefix/
 - b. Define the IP address of the managed equipment (see 'Configuring HTTP Proxy Hosts' on page 287).



Note: For this feature, no special configuration is required on the managed equipment.

■ HTTP-based EMS Services for AudioCodes Equipment behind NAT:

You can configure the device to act as an HTTP Proxy that enables AudioCodes EMS to manage AudioCodes equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and EMS is located in a public domain (e.g., in the WAN). Thus, the feature resolves NAT traversal issues. The IP Phones register with the device in order to allow communication between the IP Phones and the EMS.

To support the functionality, the following configuration is required:

1. Enable the HTTP Proxy application (see 'Enabling the HTTP Proxy Application' on page 284).
2. Configure two local, listening HTTP interfaces - one for the EMS and one for the IP Phones (see 'Configuring HTTP Interfaces' on page 284).
3. Configure the address of the EMS server (see 'Configuring an HTTP-based EMS Service' on page 289).

15.7.1 Enabling the HTTP Proxy Application

Before you can configure HTTP-based proxy services, you must enable the HTTP Proxy application, as described in the following procedure. Once enabled, the Web interface displays menus in the Navigation pane that are relevant to the HTTP Proxy application.

➤ **To enable the HTTP Proxy application:**

1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** > **Applications Enabling**).



2. From the 'HTTP Proxy 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.

15.7.2 Configuring HTTP Interfaces

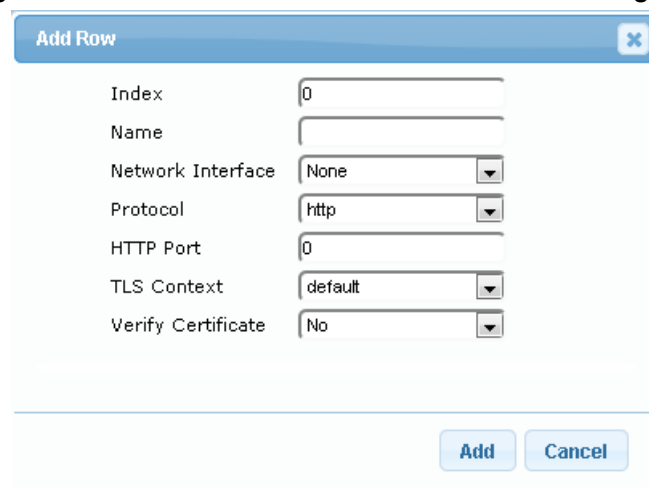
The HTTP Interfaces table lets you configure up to 10 HTTP Interfaces. An HTTP Interface represents a local, listening interface for receiving HTTP/S requests from HTTP-based (Web) clients such as managed equipment (e.g., IP Phones) and/or the EMS management tool for HTTP/S-based services.

The following procedure describes how to configure HTTP Interfaces through the Web interface. You can also configure it through ini file (HTTPInterface) or CLI (configure system > http-proxy > http-interface).

➤ **To configure an HTTP Interface:**

1. Open the HTTP Interfaces table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **HTTP Interfaces**).
2. Click **Add**; the following dialog box appears:

Figure 15-39: HTTP Interfaces Table - Add Row Dialog Box



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

Table 15-20: HTTP Interfaces Table Parameter Descriptions

Parameter	Description
Index [HTTPInterface_Index]	Defines an index number for the new table row. Notes: - Each row must be configured with a unique index. - The parameter is mandatory.
Name interface-name [HTTPInterface_InterfaceName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no value is defined. Notes: - Each row must be configured with a unique name. - The parameter is mandatory.
Network Interface network-interface [HTTPInterface_NetworkInterface]	Assigns a local, network interface to the HTTP interface. By default, no value is defined (None). For configuring network interfaces, see Configuring IP Network Interfaces on page 138. Note: The parameter is mandatory.
Protocol protocol [HTTPInterface_Protocol]	Defines the protocol type. [0] HTTP (default) [1] HTTPS
HTTP Port http-port [HTTPInterface_Port]	Defines the local, listening HTTP port. The valid value is 0 to 65534. The default is 0. Note: The parameter is mandatory.
TLS Context tls-context [HTTPInterface_TLSContext]	Assigns a TLS Context for the connection with the HTTP Proxy service. By default, the default TLS Context (Index 0) is assigned. For configuring TLS Contexts, see Configuring TLS Certificate Contexts on page 109. Note: The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above).
Verify Certificate verify-cert [HTTPInterface_VerifyCert]	Enables TLS certificate verification when the connection with the proxy service is based on HTTPS. [0] No = (Default) No certificate verification is done. [1] Yes = The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context. Note: The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above).

15.7.3 Configuring HTTP Proxy Services

The HTTP Proxy Services table lets you configure up to 10 HTTP Proxy Services.

The following procedure describes how to configure HTTP Proxy Services through the Web interface. You can also configure it through ini file (HTTPProxyService) or CLI (configure system > http-proxy > http-proxy-serv).

➤ **To configure an HTTP Proxy Service:**

1. Open the HTTP Proxy Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **HTTP Proxy Services**).
2. Click **Add**; the following dialog box appears:

Figure 15-40: HTTP Proxy Services Table - Add Row Dialog Box

The dialog box titled 'Add Row' contains the following fields and controls:

- Index:** A text input field containing the value '0'.
- Name:** An empty text input field.
- Listening Interface:** A dropdown menu with 'None' selected.
- URL Prefix:** A text input field containing the value '/'.
- Keep-Alive Mode:** A dropdown menu with 'Options' selected.
- Buttons:** 'Add' and 'Cancel' buttons located at the bottom right of the dialog.

3. Configure an HTTP Proxy service according to the parameters described in the table below.
4. Click **Add**, and then save ("burn") your settings to flash memory.

Table 15-21: HTTP Proxy Services Table Parameter Descriptions

Parameter	Description
Index [HTTPProxyService_Index]	Defines an index number for the new table row. Notes: - Each row must be configured with a unique index. - The parameter is mandatory.
Name service-name [HTTPProxyService_ServiceName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no value is defined. Notes: - Each row must be configured with a unique name. - The parameter is mandatory.
Listening Interface listening-int [HTTPProxyService_ListeningInterface]	Assigns an HTTP Interface to the HTTP Proxy service. To configure HTTP Interfaces, see 'Configuring HTTP Interfaces' on page 284. Note: The parameter is mandatory.

Parameter	Description
URL Prefix url-prefix [HTTPProxyService_URLPrefix]	Defines the URL prefix that is used to access the managed equipment's embedded Web server. The URL prefix is matched against the target of the HTTP requests sent by the client (such as GET and POST). If a match is located in the table, the device removes the prefix from the request and then forwards the HTTP request to the managed equipment without the prefix. For example, for the URL of GET /home/index.html HTTP/1.1 (which is part of the URL http://10.20.30.40/home/index.html), a URL prefix of "/home" can be configured. To match all URLs, configure the parameter to "/" (default).
Keep-Alive Mode keep-alive-mode [HTTPProxyService_KeepAliveMode]	Enables a keep-alive mechanism with the managed equipment: ▪ [0] Disable ▪ [1] Options = (Default) Enables keep-alive by sending HTTP OPTIONS messages. If no response is received from the HTTP host, the device stops forwarding HTTP requests to the host and raises an SNMP alarm (acHTTPProxyServiceAlarm). If you configured the address of the host as an FQDN (see 'Configuring HTTP Proxy Hosts' on page 287) and the DNS resolution results in multiple IP addresses, when no response is received from the keep-alive, the device checks connectivity with the next resolved IP address and so on, until a response is received.

15.7.4 Configuring HTTP Proxy Hosts

The HTTP Proxy Hosts table lets you configure up to 50 HTTP Proxy hosts (up to 5 per HTTP Proxy Service). The table is a "child" of the HTTP Proxy Services table (see 'Configuring HTTP Proxy Services' on page 286). An HTTP Proxy Host represents the HTTP-based managed equipment (e.g., IP Phone).

The following procedure describes how to configure HTTP Remote hosts through the Web interface. You can also configure it through ini file (HTTPProxyHost) or CLI (configure system > http-proxy > http-proxy-host).

➤ **To configure an HTTP Proxy Host:**

1. Open the HTTP Proxy Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **HTTP Proxy Services**).
2. In the table, select the required HTTP Proxy Service index row, and then click the **HTTP Proxy Hosts** link, located below the table; the HTTP Proxy Hosts table appears.
3. Click **Add**; the following dialog box appears:

Figure 15-41: HTTP Proxy Hosts Table - Add Row Dialog Box

4. Configure an HTTP Proxy Host according to the parameters described in the table below.
5. Click **Add**, and then save ("burn") your settings to flash memory.

Table 15-22: HTTP Proxy Hosts Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table row. Notes: - Each row must be configured with a unique index. - The parameter is mandatory.
Network Interface network-interface [HTTPProxyHost_NetworkInterface]	Assigns a local, network interface to the HTTP Proxy Host. By default, no value is defined (None). For configuring network interfaces, see Configuring IP Network Interfaces on page 138. Note: The parameter is mandatory.
Proxy Address proxy-address [HTTPProxyHost_IpAddresses]	Defines the address of the managed equipment (host). The valid value is an IP address in dotted-decimal notation or an FQDN (up to 100 characters). If the address is an FQDN, the device uses DNS to resolve it into an IP address. If the DNS resolution results in multiple IP addresses, the device uses the first available address (i.e., that responds to the keep-alive).
Protocol protocol [HTTPProxyHost_Protocol]	Defines the protocol type. [0] HTTP (default) [1] HTTPS

Parameter	Description
HTTP Port http-port [HTTPProxyHost_Port]	Defines the port of the managed equipment. The default is 0. Note: The parameter is mandatory.
TLS Context tls-context [HTTPProxyHost_TLSContext]	Assigns a TLS Context for the TLS connection with the HTTP Proxy host. By default, the default TLS Context (Index 0) is assigned. For configuring TLS Contexts, see Configuring TLS Certificate Contexts on page 109. Note: The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above).
Verify Certificate verify-cert [HTTPProxyHost_VerifyCert]	Enables TLS certificate verification when the connection with the host is based on HTTPS. [0] No = No certificate verification is done. [1] Yes = (Default) The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context. Note: The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above).

15.7.5 Configuring an HTTP-based EMS Service

The EMS Services table lets you configure a single HTTP-based EMS service. For a description of the EMS service, see 'HTTP-based Proxy Services' on page 15.7.

The following procedure describes how to configure an EMS Service through the Web interface. You can also configure it through ini file (EMSService) or CLI (configure system > http-proxy > ems-serv).

➤ **To configure an EMS Service:**

1. Open the EMS Services table (**Configuration** tab > **VoIP** menu > **Services** > **HTTP Proxy** > **EMS Services**).

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

Figure 15-42: EMS Services Table - Add Row Dialog Box

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

Table 15-23: EMS Services Table Parameter Descriptions

Parameter	Description
Index [EMSService_Index]	Defines an index number for the new table row. Notes: - Each row must be configured with a unique index. - The parameter is mandatory.
Name service-name [EMSService_ServiceName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no value is defined. Notes: - Each row must be configured with a unique name. - The parameter is mandatory.
EMS Primary Server primary-server [EMSService_PrimaryServer]	Defines the address of the primary EMS server. Note: The parameter is mandatory.
EMS Secondary Server secondary-server [EMSService_SecondaryServer]	Defines the address of the secondary EMS server.
Listening Interface to devices dev-login-int [EMSService_DeviceLoginInterface]	Assigns an HTTP Interface (local, listening HTTP interface:port) for communication with the client. To configure HTTP Interfaces, see 'Configuring HTTP Interfaces' on page 284. By default, no value is defined (None). Note: The parameter is mandatory.

Parameter	Description
Listening to EMS Interface ems-int [EMSService_EMSSInterface]	Assigns an HTTP Interface (local, listening HTTP interface:port) for communication with the EMS. To configure HTTP Interfaces, see 'Configuring HTTP Interfaces' on page 284. By default, no value is defined (None). Note: The parameter is mandatory.

15.8 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 480

If an incoming call matches the characteristics of a routing rule, the device **first** runs the assigned Call Setup Rules Set ID before routing the call according to the rule. 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.

You can also configure a Call Setup rule that determines whether the device must discontinue with the Call Setup Rules Set ID and route the call accordingly. This is done using the **Exit** optional value of the 'Action Type' parameter. When used, the 'Action Value' parameter can be configured to one of the following strings:

- "true": Indicates that if the condition is met, the device routes the call according to the selected routing rule. Note that if the condition is not met, the device also uses the selected routing rule, unless the next Call Setup rule in the Set ID has an **Exit** option configured to "false" for an empty condition.
- "false": Indicates that if the condition is met, the device attempts to route the call to the next matching routing rule (if configured). If the condition is not met, the device routes the call according to the selected routing rule.

As the default result of a Call Setup rule is always "true", please adhere to the following guidelines when configuring the 'Action Type' field to **Exit**: If, for example, you want to exit the Call Setup Rule Set ID with "true" when LDAP query result is found and "false" when LDAP query result is not found:

- Incorrect -this rule will always exit with result = True:

Condition: ldap.found exists	Action Type: Exit	Action Value: True

- Correct:

- Single rule:

Condition: ldap.found !exists	Action Type: Exit	Action Value: False

- Set of rules:

Condition: ldap.found exists	Action Type: Exit	Action Value: True

Condition: <leave it blank>	Action Type: Exit	Action Value: False



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 through the Web interface. You can also configure it through ini file (CallSetupRules) or CLI (configure voip > services call-setup-rules).

➤ **To configure a Call Setup rule:**

1. Open the Call Setup Rules table (**Configuration** tab > **VoIP** menu > **Services** > **LDAP** > **Call Setup Rules**).

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

Figure 15-43: Call Setup Rules Table - Add Row Dialog Box

The dialog box titled "Add Row" contains the following fields and values:

- Index: 4
- Rules Set ID: 0
- Query Target: (empty)
- Attributes To Query: (empty)
- Attributes To Get: (empty)
- Row Role: Use Current Condition (dropdown)
- Condition: (empty)
- Action Subject: (empty)
- Action Type: Add (dropdown)
- Action Value: (empty)

Buttons: Add, Cancel

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

Table 15-24: 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 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.
Query Target query-target [CallSetupRules_QueryTarget]	Specifies an LDAP server (LDAP Server Group) on which to perform an LDAP query. To configure LDAP Server Groups, see Configuring LDAP Server Groups on page 243.
Attributes To Query 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 attr-to-get	Defines the attributes of the queried LDAP record that the device must handle (e.g., retrieve value).

Parameter	Description
[CallSetupRules_AttributesToGet]	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 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 condition [CallSetupRules_Condition]	Defines the condition that must exist for the device to perform the action. The valid value is a string of up to 200 characters (case-insensitive). Regular Expression (regex) can also be used, for example: ▪ ldap.attr.mobile exists (attribute "mobile" exists in AD) ▪ param.call.dst.user == ldap.attr.msRTCSIP-PrivateLine (called number is the same as the number in the attribute "msRTCSIP-PrivateLine") ▪ ldap.found !exists (LDAP record not found) ▪ ldap.err exists (LDAP error exists)
Action Subject 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)

Parameter	Description
Action Type 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 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)

15.8.1 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)

15.9 Enhanced 9-1-1 Support for Lync Server

The Enhanced 9-1-1 (E9-1-1) service is becoming the mandatory emergency service required in many countries around the world. The E9-1-1 service, based on its predecessor 911, enables emergency operators to pinpoint the location (granular location) of callers who dial the 9-1-1 emergency telephone number.

Today, most enterprises implement an IP-based infrastructure providing a VoIP network with fixed and nomadic users, allowing connectivity anywhere with any device. This, together with an often deployed multi-line telephone system (MLTS) poses a challenge for E9-1-1 due to the difficulty in accurately locating the E9-1-1 caller.

This section describes the E9-1-1 solution provided by Microsoft Lync Server (hereafter referred to as *Lync Server*) and AudioCodes' device's ELIN interworking capabilities, which provides the SIP Trunk to the E9-1-1 emergency service provider. This section also describes the configuration of the device for interoperating between the Lync Server environment and the E9-1-1 emergency provider.

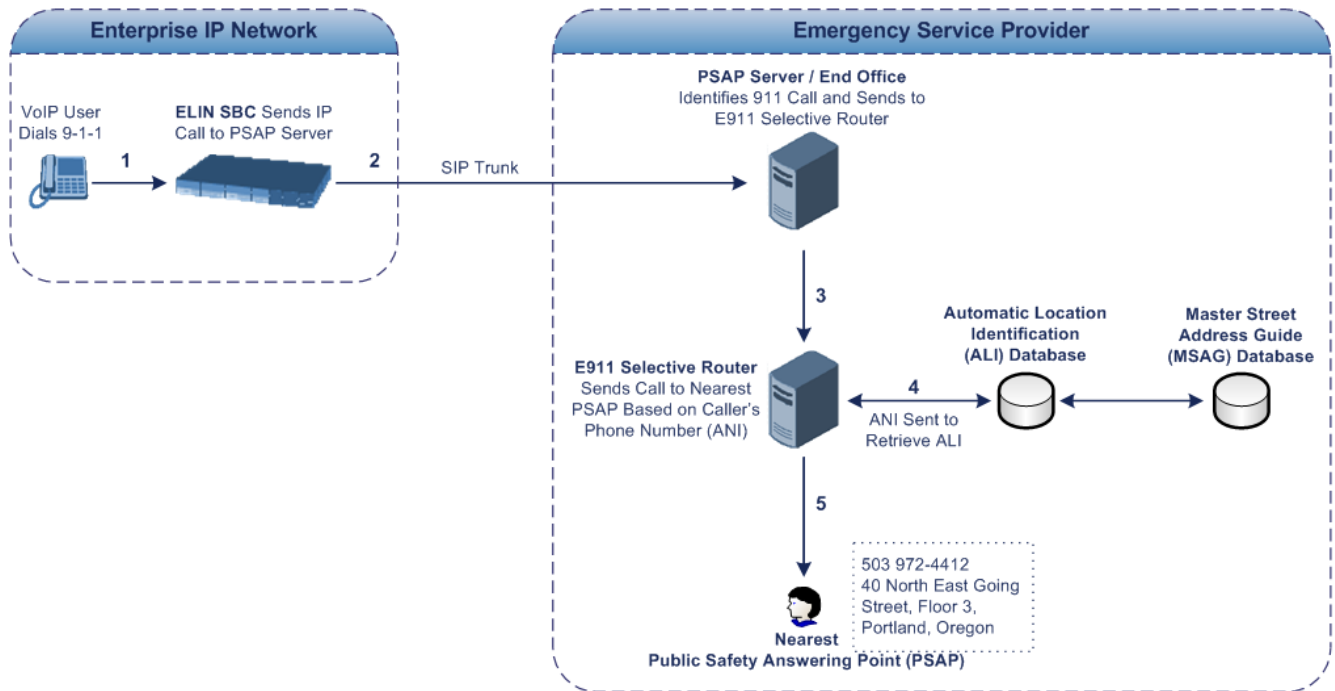


Note: The ELIN feature for E9-1-1 is a license-dependent feature and is available only if it is included in the Software License Key installed on the device. For ordering the feature, please contact your AudioCodes sales representative. For installing a new Software License Key, see Software License Key on page 596.

15.9.1 About E9-1-1 Services

E9-1-1 is a national emergency service for many countries, enabling E9-1-1 operators to automatically identify the geographical location and phone number of a 911 caller. In E9-1-1, the 911 caller is routed to the nearest E9-1-1 operator, termed *public safety answering point* (PSAP) based on the location of the caller. Automatic identification of the caller's location and phone number reduces the time spent on requesting this information from the 911 caller. Therefore, the E9-1-1 service enables the PSAP to quickly dispatch the relevant emergency services (for example, fire department or police) to the caller's location. Even if the call prematurely disconnects, the operator has sufficient information to call back the 911 caller.

The figure below illustrates the routing of an E9-1-1 call to the PSAP:



1. The VoIP user dials 9-1-1.
2. AudioCodes' ELIN device sends the call to the emergency service provider over the SIP Trunk (PSAP server).
3. The emergency service provider identifies the call is an emergency call and sends it to an E9-1-1 Selective Router in the Emergency Services provider's network.
4. The E9-1-1 Selective Router determines the geographical location of the caller by requesting this information from an Automatic Location Identification (ALI) database based on the phone number or Automatic Number Identifier (ANI) of the 911 caller. Exact location information is also supplied by the Master Street Address Guide (MSAG) database, which is a companion database to the ALI database. Phone companies and public safety agencies collaborate beforehand to create master maps that match phone numbers, addresses and cross streets to their corresponding PSAP. This MSAG is the official record of valid streets (with exact spelling), street number ranges, and other address elements with which the service providers are required to update their ALI databases.
5. The E9-1-1 Selective Router sends the call to the appropriate PSAP based on the retrieved location information from the ALI.
6. The PSAP operator dispatches the relevant emergency services to the E9-1-1 caller.

15.9.2 Microsoft Lync Server and E9-1-1

Microsoft Lync Server enables Enterprise voice users to access its unified communications platform from virtually anywhere and through many different devices. This, together with a deployed MLTS, poses a challenge for E9-1-1 due to the difficulty in accurately locating the E9-1-1 caller. However, Lync Server offers an innovative solution to solving Enterprises E9-1-1 location problems.

15.9.2.1 Gathering Location Information of Lync Clients for 911 Calls

When a Microsoft® Lync™ client (hereafter referred to as *Lync client*) is enabled for E9-1-1, the location data that is stored on the client is sent during an emergency call. This stored location information is acquired automatically from the Microsoft Location Information Server (LIS). The LIS stores the location of each network element in the enterprise. Immediately after the Lync client registration process or when the operating system detects a network connection change, each Lync client submits a request to the LIS for a location. If the LIS is able to resolve a location address for the client request, it returns the address in a location response. Each client then caches this information. When the Lync client dials 9-1-1, this location information is then included as part of the emergency call and used by the emergency service provider to route the call to the correct PSAP.

The gathering of location information in the Lync Server network is illustrated in the figure below:

1. The Administrator provisions the LIS database with the location of each network element in the Enterprise. The location is a civic address, which can include contextual in-building and company information. In other words, it associates a specific network entity (for example, a WAP) with a physical location in the Enterprise (for example, Floor 2, Wing A, and the Enterprise's street address). For more information on populating the LIS database, see "Adding ELINs to the Location Information Server" on page 301.
2. The Administrator validates addresses with the emergency service provider's MSAG – a companion database to the ALI database. This ensures that the civic address is valid as an official address (e.g., correct address spelling).
3. The Lync client initiates a location request to the LIS under the following circumstances:
 - Immediately after startup and registering the user with Lync Server
 - Approximately every four hours after initial registration
 - Whenever a network connection change is detected (such as roaming to a new WAP)

The Lync client includes in its location request the following known network connectivity information:

- Always included:
 - ◆ IPv4 subnet
 - ◆ Media Access Control (MAC) address
- Depends on network connectivity:
 - ◆ Wireless access point (WAP) Basic Service Set Identifier (BSSID)
 - ◆ Link Layer Discovery Protocol-Media Endpoint Discovery (LLDP-MED) chassis ID and port ID

For a Lync client that moves inside the corporate network such as a soft phone on a laptop that connects wirelessly to the corporate network, Lync Server can determine which subnet the phone belongs to or which WAP / SSID is currently serving the soft-client.

4. The LIS queries the published locations for a location and if a match is found, returns the location information to the client. The matching order is as follows:
 - WAP BSSID
 - LLDP switch / port
 - LLDP switch
 - Subnet
 - MAC address

This logic ensures that for any client that is connected by a wireless connection, a match is first attempted based on the hardware address of its connected access point. The logic is for the match to be based on the most detailed location. The subnet generally provides the least detail. If no match is found in the LIS for WAP BSSID, LLDP switch / port, LLDP switch, or subnet, the LIS proxies the MAC address to an integrated Simple Network Management Protocol (SNMP) scanning application. Using SNMP may benefit some organizations for the following reasons:

- LLDP is not supported by Lync Server so this provides a mechanism for soft phones to acquire detailed location information.
- Installed Layer-2 switches may not support LLDP.

If there is no match and the LIS cannot determine the location, the user may be prompted to manually enter the location. For example, the client may be located in an undefined subnet, at home, in a coffee shop or anywhere else outside the network. When a user manually provides a location, the location is mapped based on the MAC address of the default gateway of the client's network and stored on the client. When the client returns to any previously stored location, the client is automatically set to that location. A user can also manually select any location stored in the local users table and manage existing entries.

15.9.2.2 Adding ELINs to the Location Information Server

As mentioned in the previous section, the administrator needs to populate the Location Information Server (LIS) database with a network wire map, which maps the Enterprise's network elements to civic addresses. Once done, it can automatically locate clients within a network. You can add addresses individually to the LIS or in a batch using a comma-separated value (CSV) file containing the column formats listed in the table below.

Table 15-25: Columns in the LIS Database

Network Element	Columns
Wireless access point	<BSSID>,<Description>,<Location>,< CompanyName >,<HouseNumber>,<HouseNumberSuffix>,<PreDirectional>,...<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>
Subnet	<Subnet>,<Description>,<Location>,<CompanyName>,<HouseNumber>,<HouseNumberSuffix>,<PreDirectional>,...<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>
Port	<ChassisID>,<PortIDSubType>,<PortID>,<Description>,<Location>,<CompanyName>,<HouseNumber>,<HouseNumberSuffix>,...<PreDirectional>,<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>
Switch	<ChassisID>,<Description>,<Location>,<CompanyName>,<HouseNumber>,<HouseNumberSuffix>,<PreDirectional>,...<StreetName>,<StreetSuffix>,<PostDirectional>,<City>,<State>,<PostalCode>,<Country>

For the ELIN number to be included in the SIP INVITE (XML-based PIDF-LO message) sent by the Mediation Server to the ELIN device, the administrator must add the ELIN number to the <CompanyName> column (shown in the table above in **bold** typeface). As the ELIN device supports up to five ELINs per PIDF-LO, the <CompanyName> column can be populated with up to this number of ELINs, each separated by a semicolon. The digits of

each ELIN can be separated by hyphens (xxx-xxx-xxx) or they can be adjacent (xxxxxxxxxx).

When the ELIN device receives the SIP INVITE, it extracts the ELINs from the NAM field in the PIDF-LO (e.g., <ca:NAM>1111-222-333; 1234567890 </ca:NAM>), which corresponds to the <CompanyName> column of the LIS.

If you do not populate the location database, and the Lync Server location policy, Location Required is set to **Yes** or **Disclaimer**, the user will be prompted to enter a location manually.

15.9.2.3 Passing Location Information to the PSTN Emergency Provider

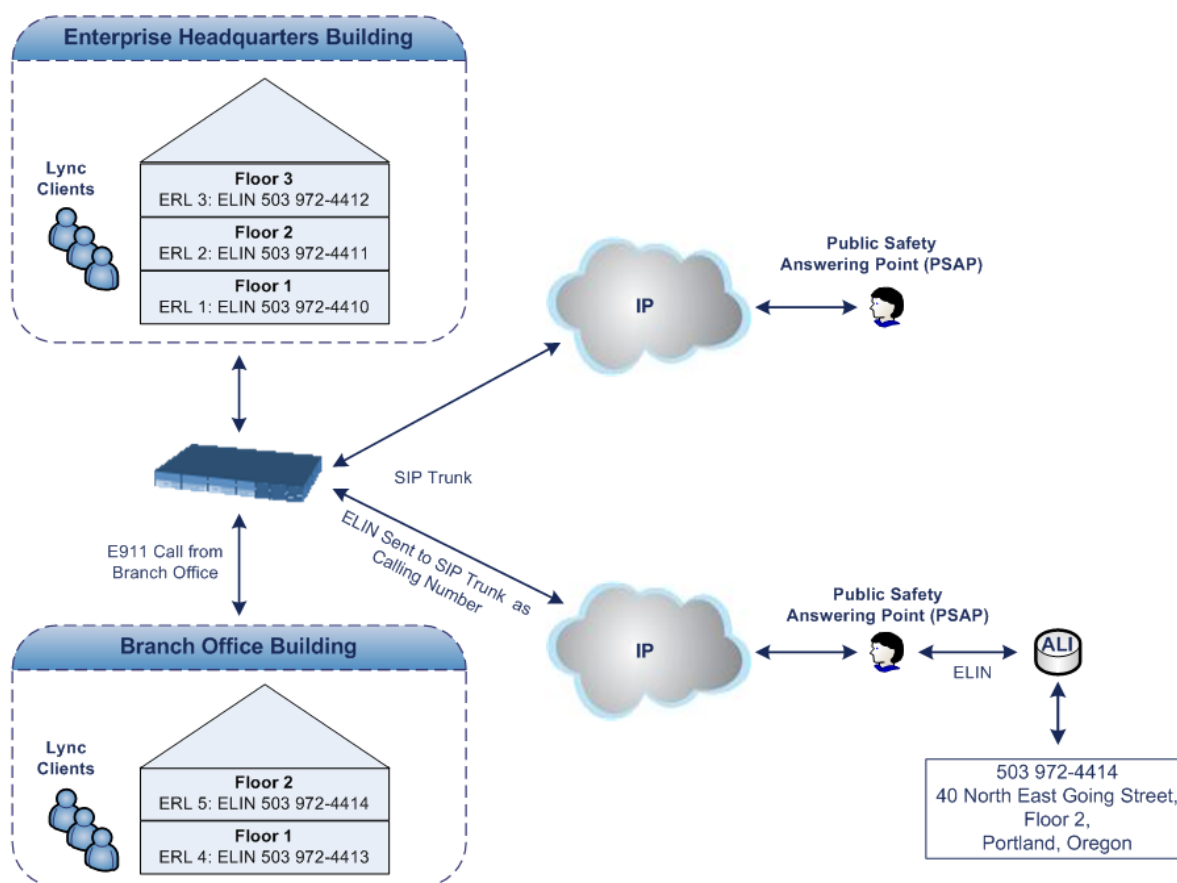
When a Lync client, enabled for E9-1-1 emergency services, dials 9-1-1, the location data and callback information stored on the client is sent with the call through the Mediation Server to a SIP Trunk-based emergency service provider. The emergency service provider then routes the call to the nearest and most appropriate PSAP based on the location information contained within the call.

Lync Server passes the location information of the Lync client in an IETF-standard format - Presence Information Data Format - Location Object (PIDF-LO)—in a SIP INVITE message. However, this content cannot be sent on the SIP Trunk due to protocol limitations. To overcome this, Enterprises deploying the device can divide their office space into Emergency Response Locations (ERLs) and assign a dedicated Emergency Location Identification Number (ELIN) to each ERL (or zone). When Lync Server sends a SIP INVITE message with the PIDF-LO to the device, it can parse the content and translate the calling number to an appropriate ELIN. The device then sends the call to the SIP Trunk with the ELIN number as the calling number. The ELIN number is sent to the emergency service provider, which sends it on to the appropriate PSAP according to the ELIN address match in the ALI database lookup.

The ERL defines a specific location at a street address, for example, the floor number of the building at that address. The geographical size of an ERL is according to local or national regulations (for example, less than 7000 square feet per ERL). Typically, you would have an ERL for each floor of the building. The ELIN is used as the phone number for 911 callers within this ERL.

The figure below illustrates the use of ERLs and ELINs, with an E9-1-1 call from floor 2 at the branch office:

Figure 15-44: Implementing ERLs and ELINs for E9-1-1 in Lync



The table below shows an example of designating ERLs to physical areas (floors) in a building and associating each ERL with a unique ELIN.

Table 15-26: Designating ERLs and Assigning to ELINs

ERL Number	Physical Area	IP Address	ELIN
1	Floor 1	10.13.124.xxx	503 972-4410
2	Floor 2	10.15.xxx.xxx	503 972-4411
3	Floor 3	10.18.xxx.xxx	503 972-4412

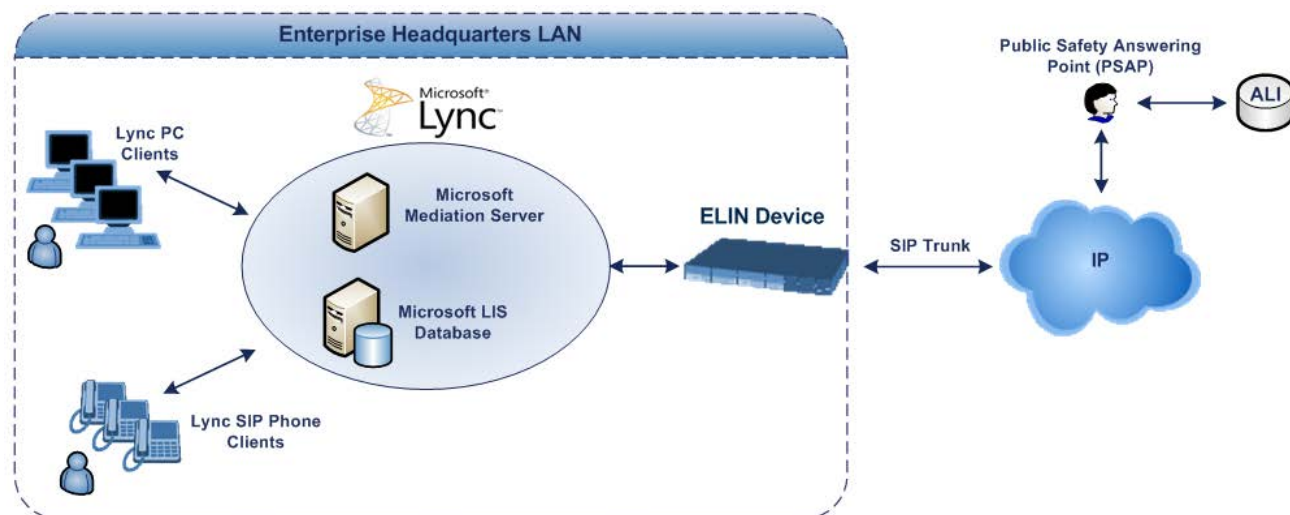
In the table above, a unique IP subnet is associated per ERL. This is useful if you implement different subnets between floors. Therefore, IP phones, for example, on a specific floor are in the same subnet and therefore, use the same ELIN when dialing 9-1-1.

15.9.3 AudioCodes ELIN Device for Lync Server E9-1-1 Calls to PSTN

Microsoft Mediation Server sends the location information of the E9-1-1 caller in the XML-based PIDF-LO body contained in the SIP INVITE message. However, this content cannot be sent on the SIP Trunk due to protocol limitations. To solve this issue, Lync Server requires a device (*ELIN SBC*) to send the E9-1-1 call to the SIP Trunk. When Lync Server sends the PIDF-LO to the device, it parses the content and translates the calling number to an appropriate ELIN. This ensures that the call is routed to an appropriate PSAP, based on ELIN-address match lookup in the emergency service provider's ALI database.

The figure below illustrates an AudioCodes ELIN device deployed in the Lync Server environment for handling E9-1-1 calls between the Enterprise and the emergency service provider.

Figure 15-45: ELIN SBC for E9-1-1 in Lync Environment



15.9.3.1 Detecting and Handling E9-1-1 Calls

The ELIN device identifies E9-1-1 calls and translates their incoming E9-1-1 calling numbers into ELIN numbers, sent toward the PSAP. The device handles the received E9-1-1 calls as follows:

1. The device identifies E9-1-1 calls if the incoming SIP INVITE message contains a PIDF-LO XML message body. This is indicated in the SIP *Content-Type* header, as shown below:

```
Content-Type: application/pidf+xml
```

2. The device extracts the ELIN number(s) from the "NAM" field in the XML message. The "NAM" field corresponds to the <CompanyName> column in the Location Information Server (LIS). The device supports up to five ELIN numbers per XML message. The ELINs are separated by a semicolon. The digits of the ELIN number can be separated by hyphens (xxx-xxx-xxx) or they can be adjacent (xxxxxxxxxx), as shown below:

```
<ca:NAM>1111-222-333; 1234567890 </ca:NAM>
```

3. The device saves the *From* header value of the SIP INVITE message in its ELIN database table (**Call From** column). The ELIN table is used for PSAP callback, as discussed later in "PSAP Callback to Lync Clients for Dropped E9-1-1 Calls" on page 306. The ELIN table also stores the following information:

- **ELIN:** ELIN number
- **Time:** Time at which the original E9-1-1 call was terminated with the PSAP

- **Count:** Number of E9-1-1 calls currently using the ELIN

An example of the ELIN database table is shown below:

ELIN	Time	Count	Index	Call From
4257275678	22:11:52	0	2	4258359333
4257275999	22:11:57	0	3	4258359444
4257275615	22:12:03	0	0	4258359555
4257275616	22:11:45	0	1	4258359777

The ELIN table stores this information for a user-defined period (see "Configuring the E9-1-1 Callback Timeout" on page 308), starting from when the E9-1-1 call, established with the PSAP, terminates. After this time expires, the table entry with its ELIN is disregarded and no longer used (for PSAP callback). Therefore, table entries of only the most recently terminated E9-1-1 callers are considered in the ELIN table. The maximum entries in the ELIN table is 300.

4. The device uses the ELIN number as the E9-1-1 calling number and sends it in the SIP INVITE message (as an ANI / Calling Party Number) to the SIP Trunk.

An example of a SIP INVITE message received from an E9-1-1 caller is shown below. The SIP *Content-Type* header indicating the PIDF-LO, and the NAM field listing the ELINs are shown in **bold** typeface.

```
INVITE sip:911;phone-context=Redmond@192.168.1.12;user=phone
SIP/2.0
From:
"voip_911_user1"<sip:voip_911_user1@contoso.com>;epid=1D19090AED;t
ag=d04d65d924
To: <sip:911;phone-context=Redmond@192.168.1.12;user=phone>
CSeq: 8 INVITE
Call-ID: e6828be1-1cdd-4fb0-bdda-cda7faf46df4
VIA: SIP/2.0/TLS 192.168.0.244:57918;branch=z9hG4bK528b7ad7
CONTACT:
<sip:voip_911_user1@contoso.com;opaque=user:epid:R4bCDaUj51a06PUbk
raS0QAA;gruu>;text;audio;video;image
PRIORITY: emergency
CONTENT-TYPE: multipart/mixed; boundary= -----
=_NextPart_000_4A6D_01CAB3D6.7519F890
geolocation: <cid:voip_911_user1@contoso.com>;inserted-
by="sip:voip_911_user1@contoso .com"
Message-Body:
-----=_NextPart_000_4A6D_01CAB3D6.7519F890
Content-Type: application/sdp ; charset=utf-8
v=0
o=- 0 0 IN IP4 Client
s=session
c=IN IP4 Client
t=0 0
m=audio 30684 RTP/AVP 114 111 112 115 116 4 3 8 0 106 97
c=IN IP4 172.29.105.23
a=rtcp:60423
a=label:Audio
a=rtpmap:3 GSM/8000/1
```

```

a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-16
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=ptime:20

-----=_NextPart_000_4A6D_01CAB3D6.7519F890
Content-Type: application/pidf+xml
Content-ID: <voip_911_user1@contoso.com>
<?xml version="1.0" encoding="utf-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
xmlns:bp="urn:ietf:params:xml:ns:pidf:geopriv10:basicPolicy"
xmlns:ca="urn:ietf:params:xml:ns:pidf:geopriv10:civicAddr"
xmlns:ms="urn:schema:Rtc.LIS.msftE911PidfExtn.2008"
entity="sip:voip_911_user1@contoso.com"><tuple
id="0"><status><gp:geopriv><gp:location-
info><ca:civicAddress><ca:country>US</ca:country><ca:A1>WA</ca:A1>
<ca:A3>Redmond</ca:A3><ca:RD>163rd</ca:RD><ca:STS>Ave</ca:STS><ca:
POD>NE</ca:POD><ca:HNO>3910</ca:HNO><ca:LOC>40/4451</ca:LOC>
<ca:NAM>1111-222-333; 1234567890 </ca:NAM>
<ca:PC>98052</ca:PC></ca:civicAddress></gp:location-
info><gp:usage-rules><bp:retransmission-
allowed>true</bp:retransmission-allowed></gp:usage-
rules></gp:geopriv><ms:msftE911PidfExtn><ms:ConferenceUri>sip:+142
55550199@contoso.com;user=phone</ms:ConferenceUri><ms:ConferenceMo
de>twoway</ms:ConferenceMode><LocationPolicyTagID
xmlns="urn:schema:Rtc.Lis.LocationPolicyTagID.2008">user-
tagid</LocationPolicyTagID
></ms:msftE911PidfExtn></status><timestamp>1991-09-
22T13:37:31.03</timestamp></tuple></presence>
-----=_NextPart_000_4A6D_01CAB3D6.7519F890--

```

15.9.3.2 Pre-empting Existing Calls for E9-1-1 Calls

If the ELIN device receives an E9-1-1 call from the IP network and there are unavailable channels (for example, all busy), the device immediately terminates one of the non-E9-1-1 calls (arbitrary) and accepts the E9-1-1 call on the freed channel:

- The preemption is done only on a call pertaining to the same source IP Group from which the E9-1-1 call is received, or the same destination IP Group (i.e., PSAP Server).

This feature is initiated only if the received SIP INVITE message contains a *Priority* header set to "emergency", as shown below:

PRIORITY: emergency

15.9.3.3 PSAP Callback to Lync Clients for Dropped E9-1-1 Calls

As the E9-1-1 service automatically provides all the contact information of the E9-1-1 caller to the PSAP, the PSAP operator can call back the E9-1-1 caller. This is especially useful in cases where the caller disconnects prematurely. However, as the Enterprise sends ELINs to the PSAP for E9-1-1 calls, a callback can only reach the original E9-1-1 caller using the device to translate the ELIN number back into the E9-1-1 caller's extension number.

In the ELIN table of the device, the temporarily stored *From* header value of the SIP INVITE message originally received from the E9-1-1 caller is used for PSAP callback. When the PSAP makes a callback to the E9-1-1 caller, the device translates the called

number (i.e., ELIN) received from the PSAP to the corresponding E9-1-1 caller's extension number as matched in the ELIN table.

The handling of PSAP callbacks by the device is as follows:

1. When the device receives a call from the emergency service provider, it searches the ELIN table for an ELIN that corresponds to the received called party number in the incoming message.
2. If a match is found in the ELIN table, it routes the call to the Mediation Server by sending a SIP INVITE, where the values of the *To* and *Request-URI* are taken from the value of the original *From* header that is stored in the ELIN table (in the **Call From** column).
3. The device updates the Time in the ELIN table. (The Count is not affected).

The PSAP callback can be done only within a user-defined period (see "Configuring the E9-1-1 Callback Timeout" on page 308), started from after the original E9-1-1 call established with the PSAP is terminated. After this time expires, the table entry with its ELIN is disregarded and no longer used (for PSAP callback). Therefore, table entries of only the most recently terminated E9-1-1 callers are considered in the ELIN table. If the PSAP callback is done after this timeout expires, the device is unable to route the call to the E9-1-1 caller and instead, either sends it as a regular call or most likely, rejects it if there are no matching routing rules. However, if another E9-1-1 caller has subsequently been processed with the same ELIN number, the PSAP callback is routed to this new E9-1-1 caller.

In scenarios where the same ELIN number is used by multiple E9-1-1 callers, upon receipt of a PSAP callback, the device sends the call to the most recent E9-1-1 caller. For example, if the ELIN number "4257275678" is being used by three E9-1-1 callers, as shown in the table below, then when a PSAP callback is received, the device sends it to the E9-1-1 caller with phone number "4258359555".

Table 15-27: Choosing Caller of ELIN

ELIN	Time	Call From
4257275678	11:00	4258359333
4257275678	11:01	4258359444
4257275678	11:03	4258359555

15.9.3.4 Selecting ELIN for Multiple Calls within Same ERL

The device supports the receipt of up to five ELIN numbers in the XML message of each incoming SIP INVITE message. As discussed in the preceding sections, the device sends the ELIN number as the E9-1-1 calling number to the emergency service provider. If the XML message contains more than one ELIN number, the device chooses the ELIN according to the following logic:

- If the first ELIN in the list is not being used by other active calls, it chooses this ELIN.
- If the first ELIN in the list is being used by another active call, the device skips to the next ELIN in the list, and so on until it finds an ELIN that is not being used and sends this ELIN.
- If all the ELINs in the list are in use by active calls, the device selects the ELIN number as follows:
 1. The ELIN with the lowest count (i.e., lowest number of active calls currently using this ELIN).

2. If the count between ELINs is identical, the device selects the ELIN with the greatest amount of time passed since the original E9-1-1 call using this ELIN was terminated with the PSAP. For example, if E9-1-1 caller using ELIN 4257275678 was terminated at **11:01** and E9-1-1 caller using ELIN 4257275670 was terminated at **11:03**, then the device selects ELIN 4257275678.

In this scenario, multiple E9-1-1 calls are sent with the same ELIN.

15.9.4 Configuring AudioCodes ELIN Device

This section describes E9-1-1 configuration of the AudioCodes ELIN Gateway deployed in the Lync Server environment.

15.9.4.1 Enabling the E9-1-1 Feature

By default, the ELIN device feature for E9-1-1 emergency call handling in a Lync environment is disabled.

➤ To enable the ELIN feature:

- Configure the 'SBC PSAP Mode' parameter to **Enable** for the IP Group through which you want to communicate with the public-safety answering point (PSAP). For more information on IP Groups, see "Configuring IP Groups" on page 348.

15.9.4.2 Configuring the E9-1-1 Callback Timeout

The PSAP can use the ELIN to call back the E9-1-1 caller within a user-defined time interval (in minutes) from when the initial call established with the PSAP has been terminated. By default, an ELIN can be used for PSAP callback within 30 minutes after the call is terminated. You can change this to any value between 0 and 60:

➤ To configure the E9-1-1 callback timeout

1. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).
2. In the 'E911 Callback Timeout' field (E911CallbackTimeout), enter the required callback timeout.

15.9.4.3 Configuring the SIP Release Cause Code for Failed E9-1-1 Calls

When a Lync client makes an emergency call, the call is routed through the Microsoft Mediation Server to the ELIN device, which sends it on to the SIP Trunk. In some scenarios, the call may not be established due to either the destination (for example, busy or not found) or the ELIN device (for example, lack of resources or an internal error). In such a scenario, the Mediation Server requires that the ELIN device "reject" the call with the SIP release cause code 503 "Service Unavailable" instead of the designated release call. Such a release cause code enables the Mediation Server to issue a failover to another entity (for example, another ELIN device), instead of retrying the call or returning the release call to the user.

To support this requirement, you can configure the ELIN device to send a 503 "Service Unavailable" release cause code instead of SIP 4xx if an emergency call cannot be established:

➤ To enable SIP response 503 upon failed E911:

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

2. From the 'Emergency Special Release Cause' drop-down list (EmergencySpecialReleaseCause), select **Enable**.

15.9.4.4 Configuring SBC IP-to-IP Routing Rule for E9-1-1

To route incoming E9-1-1 calls to the emergency service provider's PSAP server, you need to configure routing rules in the IP-to-IP Routing table for routing between the emergency callers' IP Group and the PSAP server's IP Group. The only special configuration is to define the emergency number (e.g., 911) in the 'Destination Username Prefix' parameter of the IP Group belonging to the E9-1-1 callers. The following example shows IP-to-IP routing rules for E9-1-1 in a Lync environment:

Figure 15-46: Example of IP-to-IP Routing Rules for Lync E9-1-1

Index	Name	Routing Policy	Alternative Route Options	Source IP Group	Request Type	Source Username Prefix	Destination Username Prefix	Destination Type	Destination IP Group
1	E911 > PSAP	Default_SBCRc	Route Row	LAN IP PBX	All	*	911	IP Group	PSAP Server
2	PSAP > E911	Default_SBCRc	Route Row	PSAP Server	All	*	*	IP Group	LAN IP PBX

15.9.4.5 Viewing the ELIN Table

To view the ELIN table:

- CLI

```
# show voip e911
ELIN          Time    Count Index Call From
-----
4257275678    22:11:52  0    2    4258359333
4257275999    22:11:57  0    3    4258359444
4257275615    22:12:03  0    0    4258359555
4257275616    22:11:45  0    1    4258359777
----- Current Time: 22:12:40
```

- Using Syslog, by invoking the following Web command shell:

```
SIP / GateWay / E911Dump
```

This page is intentionally left blank.

16 Quality of Experience

This chapter describes how to configure the Quality of Experience feature.

16.1 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 *SEM User's Manual*.

16.1.1 Configuring the SEM Server

The device can be configured to report QoE voice metrics to a single SEM server or to two SEM 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 normal HA mode, when both SEM/EMS servers are located in the same subnet, a single SEM/EMS 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.

You can also configure the device to use a TLS connection with the SEM server. Before you can do this, configure a TLS Context (certificates) in the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 109). If no TLS Context is specified, the device uses the default TLS Context (ID 0).

You can also configure at what stage of the call the device must send the report to the SEM server. The report can be sent during the call or only at the end of the call. Reporting at the end of the call may be beneficial when network congestion occurs, as this reduces bandwidth usage over time.



Note: If a QoE traffic overflow is experienced between SEM and the device, the device sends the QoE data only at the end of the call, regardless of your settings.

For a detailed description of the SEM parameters, see "Quality of Experience Parameters" on page 763.

➤ To configure the SEM server address and other related features:

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

Figure 16-1: Session Experience Manager Server Page

Session Experience Manager Server		
Server IP	0.0.0.0	
Redundant Server IP	0.0.0.0	
Interface Name	OAMP	
QoE Report Mode	Report QoE During Call	▼
QoE Connection by TLS	Disable	▼
QoE TLS Context Name	MED	▼

2. Configure the address of the SEM server:
 - a. In the 'Server IP' field, enter the primary SEM server's IP address.
 - b. If Geographical-Redundancy HA mode exists, in the 'Redundant Server IP' field, enter the secondary SEM server's IP address.
 - c. In the 'Interface Name' field, enter the device's IP network interface from which the device sends the reports to the SEM server.
3. From the 'QoE Report Mode' drop-down list, select when you want the device to send reports of a call to the SEM.
4. (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).
5. Click **Submit**, and then save ("burn") your settings to flash memory.

16.1.2 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 125.

16.1.3 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 649. For configuring what to report to the SEM, see "Configuring Quality of Experience Profiles" on page 313.

16.2 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 348), Media Realms (see "Configuring Media Realms" on page 323), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 326).

The monitored voice 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 let you configure 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 the SEM Server" on page 311.
- 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 320.

- 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 491).



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 through the Web interface. You can also configure it through other management platforms:

- Quality of Experience Profile table:** *ini* file (QoEProfile) or CLI (configure voip/qoe qoe-profile)
- Quality of Experience Color Rules table:** *ini* file (QOECOLORRules) or CLI (configure voip/qoe qoe-profile qoe-color-rules)

➤ **To configure a QoE Profile:**

- Open the Quality of Experience Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Quality of Experience Profile**).
- Click **Add**; the following dialog box appears:

Figure 16-2: Quality of Experience Profile Table - Add Row Dialog Box

- Configure a QoE Profile according to the parameters described in the table below.
- Click **Add**.

Table 16-1: Quality of Experience Profile Table Parameter Descriptions

Parameter	Description
Index [QoEProfile_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Profile Name name [QoEProfile_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 20 characters.

Parameter	Description
Sensitivity Level 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 = (Default) 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 16-3: Quality of Experience Table - Add Row Dialog Box

The dialog box titled "Add Row" contains a red error message: "You have exceeded the maximum number of rows in this table". Below the message are several input fields:

- Index: -1
- Monitored Parameter: MOS
- Direction: Device Side
- Sensitivity Level: Medium
- Green Yellow Threshold: 0
- Green Yellow Hysteresis: 0
- Yellow Red Threshold: 0
- Yellow Red Hysteresis: 0

At the bottom right, there are two buttons: "Add" and "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 **Add**, and then save ("burn") your settings to flash memory.

Table 16-2: Quality of Experience Color Rules Table Parameter Descriptions

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

Parameter	Description
Monitored Parameter 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 direction [QOECOLORRules_direction]	Defines the monitoring direction. [0] Device Side (default) [1] Remote Side
Sensitivity Level 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 are 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 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 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 yellow-red-threshold [QOECOLORRules_YellowRedThreshold]	Defines the parameter threshold values between Yellow (medium quality) and Red (poor quality) states. The valid threshold values are as follows: - MOS values are in multiples of 10. For example, to denote a MOS of 3.2, the value 32 (i.e., 3.2*10) must be entered. - Delay values are in msec. - Packet Loss values are in percentage (%). - Jitter is in msec. - Echo measures the Residual Echo Return Loss (RERL) in dB.

Parameter	Description
Yellow Red Hysteresis 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.

16.3 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 348), Media Realms (see "Configuring Media Realms" on page 323), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 326).

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 320.
- 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 491).
- 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 through the Web interface. You can also configure it through ini file (BWProfile) or CLI (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 16-4: Bandwidth Profile Table - Add Row Dialog Box

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 **Add**, and then reset the device with a save ("burn") to flash memory.

Table 16-3: Bandwidth Profile Table Parameter Descriptions

Parameter	Description
Index [BWProfile_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [BWProfile_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 20 characters.
Egress Audio Bandwidth egress-audio-bandwidth [BWProfile_EgressAudioBandwidth]	Defines the outgoing audio traffic threshold (in Kbps).
Ingress Audio Bandwidth ingress-audio-bandwidth [BWProfile_IngressAudioBandwidth]	Defines the incoming audio traffic threshold (in Kbps).
Egress Video Bandwidth egress-video-bandwidth [BWProfile_EgressVideoBandwidth]	Defines the outgoing video traffic threshold (in Kbps).

Parameter	Description
Ingress Video Bandwidth ingress-video-bandwidth [BWProfile_IngressVideoBandwidth]	Defines the incoming video traffic threshold (in Kbps).
Total Egress Bandwidth total-egress-bandwidth [BWProfile_TotalEgressBandwidth]	Defines the total (video and audio) outgoing bandwidth threshold (in Kbps).
Total Ingress Bandwidth total-ingress-bandwidth [BWProfile_TotalIngressBandwidth]	Defines the total (video and audio) incoming bandwidth threshold (in Kbps).
Warning Threshold 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 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 the 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 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).

16.4 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 313) and bandwidth utilization (configured in "Configuring Bandwidth Profiles" on page 317). 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 through the Web interface. You can also configure it through other management platforms:

- **Media Enhancement Profile table:** *ini* file (MediaEnhancementProfile) or CLI (configure voip/qoe media-enhancement)
- **Media Enhancement Rules table:** *ini* file (MediaEnhancementRules) or CLI (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 16-5: Media Enhancement Profile Table - Add Row Dialog Box

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

Table 16-4: Media Enhancement Profile Table Parameter Descriptions

Parameter	Description
Index [MediaEnhancementProfile_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name profile-name [MediaEnhancementProfile_Profil eName]	Defines an arbitrary name to easily identify the row. 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 16-6: Media Enhancement Rules Table - Add Row Dialog Box

7. Configure a Media Enhancement Rule according to the parameters described in the table below.

8. Click **Add**, and then reset the device with a save ("burn") to flash memory.

Table 16-5: Media Enhancement Rules Table Parameter Descriptions

Parameter	Description
Index rule-index [MediaEnhancementRules_RuleIndex]	Defines the index of the table row entry.
Trigger 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 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 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 is used, as configured in the 'Value' field (below). Notes: ▪ If the 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 the 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.
Alternative IP Profile ID value [MediaEnhancementRules_ActionValue]	Defines an alternative IP Profile ID for the IP Group that is associated with this rule, if this rule is applied. The parameter is applicable only if the 'Rule Action' parameter is set to Alternative IP Profile .

17 Control Network

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

17.1 Configuring Media Realms

The Media Realm table lets you configure a pool of up to 64 SIP media interfaces, termed *Media Realms*. Media Realms lets you divide a Media-type interface (configured in the Interface table) into several media realms, where each realm is specified by a UDP port range. Media Realms also define the maximum number of permitted media sessions.

Once configured, to apply Media Realms to specific calls, you need to assign them to any of the following configuration entities:

- IP Groups (see "Configuring IP Groups" on page 348)
- SIP Interfaces (see "Configuring SIP Interfaces" on page 342)

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 313.
- **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 317.

The Media Realm table provides sub-tables ("child" tables) that let you configure the following:

- Remote Media Subnets: Defines remote destination subnets per Media Realm and assigns each subnet a Quality of Experience Profile and Bandwidth Profile. For more information, see "Configuring Remote Media Subnets" on page 326.
- Media Realm Extensions: Defines port ranges for multiple Media-type interfaces per Media Realm. For more information, see "Configuring Media Realm Extensions" on page 329.



Notes:

- The Media Realm assigned to an IP Group overrides any other Media Realm assigned to any other configuration entity associated with the call.
- If you modify a Media Realm that is currently being used by a call, the device does not perform Quality of Experience for the call.
- If you delete a Media Realm that is currently being used by a call, the device maintains the call until the call parties end the call.

The following procedure describes how to configure Media Realms through the Web interface. You can also configure it through ini file (CpMediaRealm) or CLI (configure voip > voip-network realm).

➤ **To configure a Media Realm:**

1. Open the Media Realm table (**Configuration** tab > **VoIP** menu > **VoIP Network** > **Media Realm Table**).
2. Click **Add**; the following dialog box appears:

Figure 17-1: Media Realm Table - Add Row Dialog Box

3. Configure the Media Realm according to the parameters described in the table below.
4. Click **Add**.

Table 17-1: Media Realm Table Parameter Descriptions

Parameter	Description
Index [CpMediaRealm_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [CpMediaRealm_Media RealmName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. Notes: - The parameter is mandatory. - Each row must be configured with a unique name.
IPv4 Interface Name ipv4 [CpMediaRealm_IPv4I F]	Assigns an IPv4 network interface to the Media Realm. This is the name of the interface as configured in the 'Interface Name' parameter in the Interface table. By default, no value is defined (None).
IPv6 Interface Name ipv6if [CpMediaRealm_IPv6I F]	Assigns an IPv6 network interface to the Media Realm. This is the name of the interface as configured for the 'Interface Name' parameter in the Interface table. By default, no value is defined (None).

Parameter	Description
Port Range Start port-range-start [CpMediaRealm_PortRangeStart]	Defines the starting port for the range of media interface UDP ports. By default, no value is defined. Notes: - You must either configure all your 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 201. - 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 342). 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. - 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.
Number of Media Session Legs session-leg [CpMediaRealm_MediaSessionLeg]	Defines the number of media sessions for the configured port range. By default, no value is defined.
Port Range End 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.

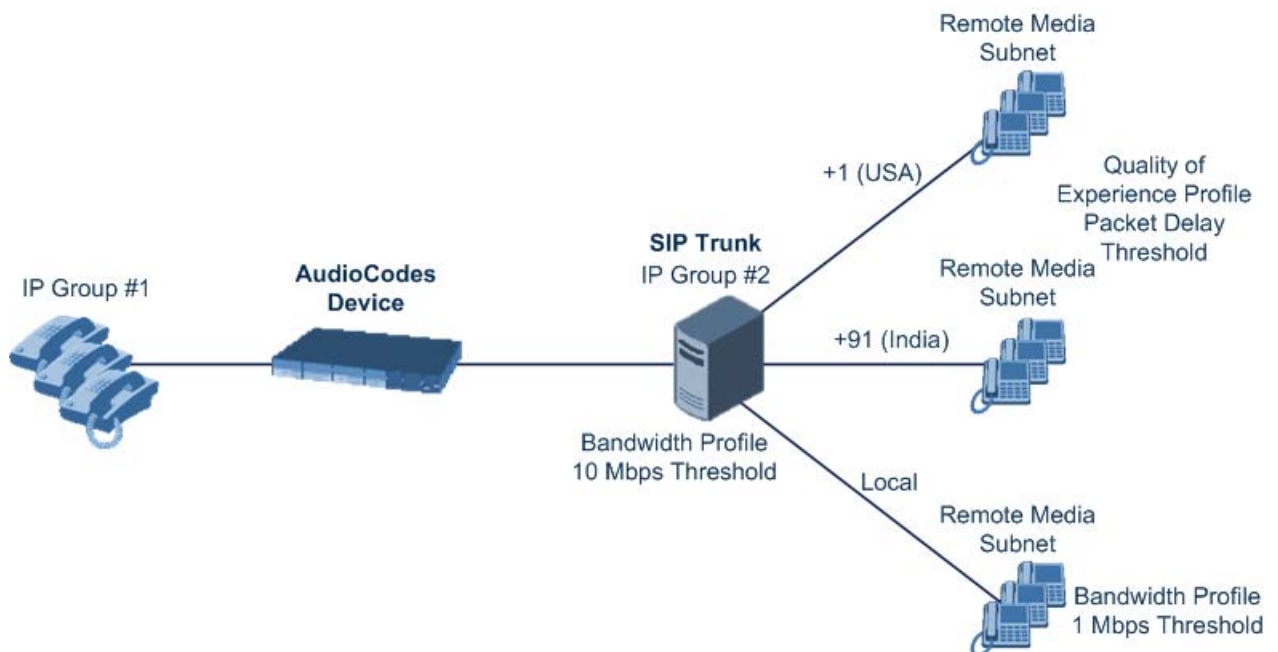
Parameter	Description
Default Media Realm is-default [CpMediaRealm_IsDefault]	Defines the Media Realm as the default Media Realm. The default Media Realm is used for SIP Interfaces and IP Groups for which you have not assigned a Media Realm. ▪ [0] No (default) ▪ [1] Yes Notes: ▪ You can configure the parameter to Yes for only one Media Realm; all the other Media Realms must be configured to No. ▪ If you do not configure the parameter (i.e., the parameter is No for all Media Realms), the device uses the first Media Realm in the table as the default. ▪ If the table is not configured, the default Media Realm includes all configured media interfaces.
QoE Profile qoe-profile [CpMediaRealm_QoeProfile]	Assigns a QoE Profile to the Media Realm. By default, no value is defined (None). For configuring QoE Profiles, see "Configuring Quality of Experience Profiles" on page 313.
BW Profile bw-profile [CpMediaRealm_BWProfile]	Assigns a Bandwidth Profile to the Media Realm. By default, no value is defined (None). For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 317.

17.1.1 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 313 and "Configuring Bandwidth Profiles" on page 317, 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 17-2: Remote Media Subnets Example



The following procedure describes how to configure Remote Media Subnets through the Web interface. You can also configure it through ini file (RemoteMediaSubnet) or CLI (configure voip > voip-network realm remote-media-subnet).

➤ **To configure a Remote Media Subnet:**

1. Open the Media Realm table (see "Configuring Media Realms" on page 323).
2. Select the Media Realm 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 17-3: Remote Media Subnet Table - Add Row Dialog Box

The dialog box titled "Add Row" contains the following fields and values:

- Index: 0
- Name: (empty)
- Prefix Length: 16
- Address Family: IPv4
- Destination IP: 0.0.0.0
- QoE Profile: None
- BW Profile: None

Buttons: Add, Cancel

4. Configure the Remote Media Subnet according to the parameters described in the table below.
5. Click **Add**.

Table 17-2: Remote Media Subnet Table Parameter Descriptions

Parameter	Description
Index [RemoteMediaSubnet_ RemoteMediaSubnetIndex]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [RemoteMediaSubnet_ RemoteMediaSubnetName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 20 characters. Note: Each row must be configured with a unique name.
Prefix Length 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 address-family [RemoteMediaSubnet_ AddressFamily]	Defines the IP address protocol. [2] IPv4 (default) [10] IPv6
Destination IP dst-ip-address [RemoteMediaSubnet_ DstIPAddress]	Defines the IP address of the destination. The default is 0.0.0.0.
QOE Profile Name qoe-profile [RemoteMediaSubnet_ QOEProfileName]	Assigns a Quality of Experience Profile to the Remote Media Subnet. By default, no value is defined (None). For configuring QoE Profiles, see "Configuring Quality of Experience Profiles" on page 313.

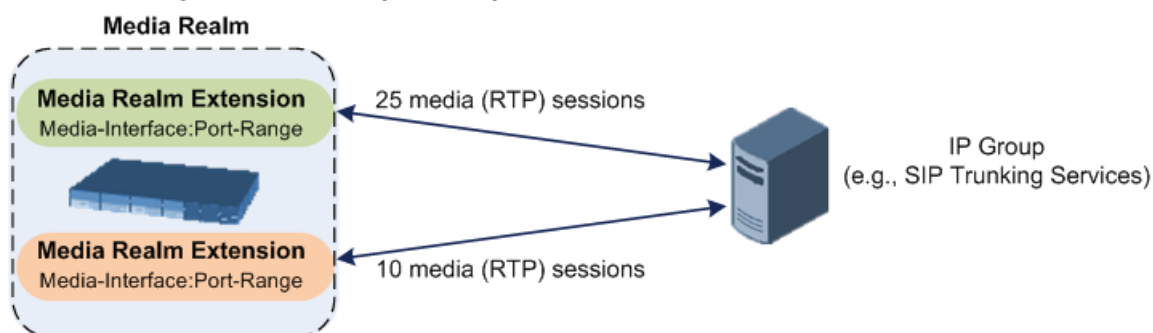
Parameter	Description
BW Profile Name bw-profile	Assigns a Bandwidth Profile to the Remote Media Subnet. By default, no value is defined (None).
[RemoteMediaSubnet_ BWProfileName]	For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 317.

17.1.2 Configuring Media Realm Extensions

The Media Realm Extension table lets you configure 2 Media Realm Extensions. A Media Realm Extension is associated with a specific Media Realm and defines a port range and the number of media sessions for a specific Media-type network interface (configured in the IP Interfaces table). Therefore, a Media Realm Extension enhances a Media Realm by allowing you to define different port ranges, media sessions, and network interface than is defined by the associated Media Realm (i.e., the Media Realm is distributed across multiple interfaces).

Media Realm Extensions can be useful, for example, to overcome limitations of the maximum number of media ports supported per interface. Instead of configuring only a single Media Realm in the Media Realm table (see "Configuring Media Realms" on page 323), you can also configure additional "Media Realms" in the Media Realm Extensions table associated with the single Media Realm. An IP Group that is associated with a Media Realm configured with Media Realm Extensions, allocates its media sessions / ports between the different interfaces, as configured by the Media Real and its associated Media Realm Extensions. For example, two Media Realm Extensions could be configured, whereby one allocates 25 media sessions on interface "LAN-1" and another, 10 sessions on interface "LAN-2". The Media Realm associated with these Media Realm Extensions would be assigned to the relevant IP Group.

Figure 17-4: Example of Implementation of Media Realm Extensions



The following procedure describes how to configure Media Realm Extensions through the Web interface. You can also configure it through ini file (MediaRealmExtension).

➤ **To configure a Media Realm Extension:**

1. Open the Media Realm table (see "Configuring Media Realms" on page 323).
2. Select the Media Realm row for which you want to add Remote Media Extensions, and then click the **Media Realm Extension** link located below the table; the Media Realm Extension table appears.

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

Figure 17-5: Media Realm Extension Table - Add Row Dialog Box

4. Configure the Media Realm Extension according to the parameters described in the table below.
5. Click **Add**.

Table 17-3: Media Realm Extension Table Parameter Descriptions

Parameter	Description
Index [MediaRealmExtension_ExtensionIndex]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
IPv4 Interface Name [MediaRealmExtension_IPv4IF]	Assigns an IPv4 network interface (configured in the Interface table) to the Media Realm Extension. By default, no value is defined (None). For configuring IP network interfaces, see "Configuring IP Network Interfaces" on page 138. Note: - The parameter is mandatory. - You must configure the Media Realm Extension with an IP network interface that has the same IP version(s) as the Media Realm to which the Media Realm Extension is associated. If the associated Media Realm is assigned both an IPv4 and IPv6 network interface, you also need to assign the Media Realm Extension with both an IPv4 and IPv6 network interface. For example, if the associated Media Realm is assigned only an IPv4 network interface, you also need to assign the Media Realm Extension with an IPv4 network interface.
IPv6 Interface Name [MediaRealmExtension_IPv6IF]	Assigns an IPv6 network interface (configured in the Interface table) to the Media Realm Extension. By default, no value is defined (None). Note: - The parameter is mandatory. - You must configure the Media Realm Extension with an IP network interface that has the same IP version(s) as the Media Realm to which the Media Realm Extension is associated. If the associated Media Realm is assigned both an IPv4 and IPv6 network interface, you also need to assign the Media Realm Extension with both an IPv4 and IPv6 network interface. For example, if the associated Media Realm is assigned only an IPv6 network interface, you also need to assign the Media Realm Extension with an IPv6 network interface.

Parameter	Description
Port Range Start [MediaRealmExtension _PortRangeStart]	Defines the first (lower) port in the range of media UDP ports for the Media Realm Extension. By default, no value is defined. Notes: ▪ You must either configure all your 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 201.
Port Range End [MediaRealmExtension _PortRangeEnd]	Defines the last (upper) port in the range of media UDP ports for the Media Realm Extension. Note: It is unnecessary to configure the parameter. The device automatically populates the parameter with a value, calculated by the summation of the 'Number of Media Session Legs' parameter (multiplied by the port chunk size) and the 'Port Range Start' parameter. After you have added the Media Realm Extension row to the table, the parameter is displayed with the calculated value.
Number Of Media Session Legs [MediaRealmExtension _MediaSessionLeg]	Defines the number of media sessions for the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10. By default, no value is defined. Note: The parameter is mandatory.

17.2 Configuring SRDs

The SRD table lets you configure up to 250 signaling routing domains (SRD). The SRD is a logical representation of an entire SIP-based VoIP network (Layer 5) consisting of groups of SIP users and servers. The SRD is associated with all the configuration entities (e.g., SIP Interfaces and IP Groups) required for routing calls within the network. Typically, only a **single** SRD is required (recommended) for most deployments. Multiple SRDs are only required for multi-tenant deployments, where the physical device is "split" into multiple logical devices. For more information on multi-tenant architecture, see "Multiple SRDs for Multi-tenant Deployments" on page 337.

As the device is shipped with a default SRD ("DefaultSRD" at Index 0), if your deployment requires only one SRD, you can use the default SRD instead of creating a new one. When only one SRD is employed and you create other related configuration entities (e.g., SIP Interfaces), the default SRD is automatically assigned to the new configuration entity. Therefore, when employing a single-SRD configuration topology, there is no need to handle SRD configuration (i.e., transparent).

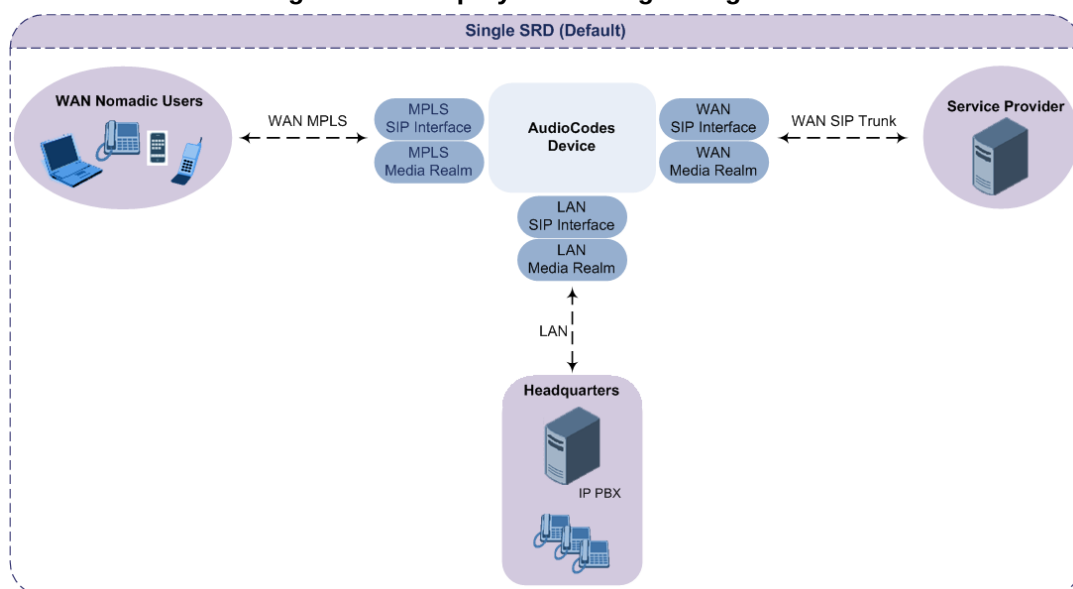
SRDs are associated with the following configuration entities:

- SIP Interface (mandatory) - see "Configuring SIP Interfaces" on page 342
- IP Group (mandatory) - see "Configuring IP Groups" on page 348
- Proxy Set (mandatory) - see "Configuring Proxy Sets" on page 360
- Admission Control rule - see Configuring Admission Control Table on page 463
- Classification rule - see Configuring Classification Rules on page 471

As mentioned previously, if you use only a single SRD, the device automatically assigns it to the above-listed configuration entities.

As each SIP Interface defines a different Layer-3 network (see "Configuring SIP Interfaces" on page 342 for more information) on which to route or receive calls and as you can assign multiple SIP Interfaces to the same SRD, for most deployment scenarios (even for multiple Layer-3 network environments), you only need to employ a single SRD to represent your VoIP network (Layer 5). For example, if your VoIP deployment consists of an Enterprise IP PBX (LAN), a SIP Trunk (WAN), and far-end users (WAN), you would only need a single SRD. The single SRD would be assigned to three different SIP Interfaces, where each SIP Interface would represent a specific Layer-3 network (IP PBX, SIP Trunk, or far-end users) in your environment. The following figure provides an example of such a deployment:

Figure 17-6: Deployment using a Single SRD



**Notes:**

- It is recommended to use a single-SRD configuration topology, unless you are deploying the device in a multi-tenant environment, in which case, multiple SRDs are required.
- Each SIP Interface, Proxy Set, and IP Group can be associated with only one SRD.
- If you have upgraded your device to Version 7.0 and your device was configured with multiple SRDs but not operating in a multi-tenant environment, it is recommended to gradually change your configuration to a single SRD topology.
- If you upgrade the device from an earlier release to Version 7.0, your previous SRD configuration is fully preserved regarding functionality. The same number of SRDs is maintained, but the configuration elements are changed to reflect the configuration topology of Version 7.0. Below are the main changes in configuration topology when upgrading to Version 7.0:
 - ✓ The SIP Interface replaces the associated SRD in several tables (due to support for multiple SIP Interfaces per SRD).
 - ✓ Some fields in the SRD table were duplicated or moved to the SIP Interface table.
 - ✓ Indices used for associating configuration entities in tables are changed to row pointers (using the entity's name).
 - ✓ Some tables are now associated (mandatory) with an SRD (SIP Interface, IP Group, Proxy Set, and Classification).
 - ✓ Some fields used for associating configuration entities in tables now have a value of **Any** to distinguish between **Any** and **None** (deleted entity or not associated).

The following procedure describes how to configure SRDs through the Web interface. You can also configure it through ini file (SRD) or CLI (configure voip > voip-network srd).

➤ **To configure an SRD:**

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

Figure 17-7: SRD Table - Add Row Dialog Box

3. Configure an SRD according to the parameters described in the table below.
4. Click **Add**.

Table 17-4: SRD Table Parameter Descriptions

Parameter	Description
Index [SRD_Index]	Defines an index for the new table row. Note: Each row must be configured with a unique index.
Name name [SRD_Name]	Defines an arbitrary name to easily identify the row. The valid value can be a string of up to 40 characters. Notes: ▪ The parameter is mandatory. ▪ Each row must be configured with a unique name.
Sharing Policy type [SRD_SharingPolicy]	Defines the sharing policy of the SRD, which determines whether the SRD shares its SIP resources (SIP Interfaces, Proxy Sets, and IP Groups) with all other SRDs (Shared and Isolated). ▪ [0] Shared = (Default) SRD shares its resources with other SRDs (Isolated and Shared) and calls can thus be routed between the SRD and other SRDs. ▪ [1] Isolated = SRD does not share its resources with other SRDs and calls cannot be routed between the SRD and other Isolated SRDs. However, calls can be routed between the SRD and other Shared SRDs. For more information on SRD Sharing Policy, see Multiple SRDs for Multi-tenant Deployments on page 337.

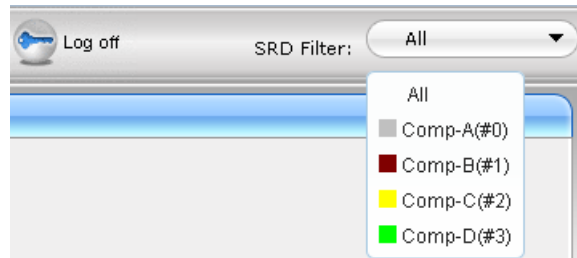
Parameter	Description
SBC Operation Mode sbc-operation-mode [SRD_SBCOperationMode]	Defines the device's operational mode for the SRD. [0] B2BUA = (Default) Device operates as a back-to-back user agent (B2BUA), changing the call identifiers and headers between the inbound and outbound legs. [1] Call Stateful Proxy = Device operates as a Stateful Proxy, passing the SIP message transparently between inbound and outbound legs. In other words, the same SIP dialog identifiers (tags, Call-Id and CSeq) occur on both legs (as long as no other configuration disrupts the CSeq compatibility). For more information on B2BUA and Stateful Proxy modes, see B2BUA and Stateful Proxy Operating Modes on page 425. Notes: - The settings of the parameter also determines the default behavior of related parameters in the IP Profile table (SBCRemoteRepresentationMode, SBCKeepVIAHeaders, SBCKeepUserAgentHeader, SBCKeepRoutingHeaders, SBCRemoteMultipleEarlyDialogs). - If the 'SBC Operation Mode' parameter is configured in the IP Group table, the 'SBC Operation Mode' parameter in the SRD table is ignored.
SBC Routing Policy sbc-routing-policy-name [SRD_SBCRoutingPolicyName]	Assigns an SBC Routing Policy to the SRD. By default, no value is defined (None) if you have configured multiple SBC Routing Policies. If you have configured only one SBC Routing Policy, the device assigns it to the SRD by default. For more information on SBC Routing Policies, see Configuring SBC Routing Policy Rules on page 493. Notes: - If you have assigned an SBC Routing Policy to a Classification rule that is associated with the SRD, the SBC Routing Policy assigned to the SRD is ignored. - You can assign the same Routing Policy to multiple SRDs.
Max. Number of Registered Users max-reg-users [SRD_MaxNumOfRegUsers]	Defines the maximum number of users belonging to the SRD that can register with the device. The default is -1, which means that the number of allowed user registrations is unlimited.
Block Unregistered Users block-un-reg-users [SRD_BlockUnRegUsers]	Enables the device to block (reject) incoming calls (INVITE requests) from unregistered users belonging to the SRD. [0] No = (Default) Calls from unregistered users are allowed. [1] Yes = Calls from unregistered users are blocked. Notes: - When the device blocks a call, it sends a SIP 500 "Server Internal Error" response to the remote end. - The parameter applies to calls belonging to a User-type IP Group. - When the corresponding parameter in the SIP Interface table (SIPInterface_BlockUnRegUsers) is configured to Yes or No for a SIP Interface that is associated with the SRD, the parameter in the SRD table is ignored for calls belonging to the SIP Interface.
Enable Un-Authenticated	Enables the device to accept REGISTER requests and register them in its

Parameter	Description
Registrations enable-un-auth-registrs [SRD_EnableUnAuthenticatedRegistrations]	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. Notes: - Regardless of the parameter, the device always accepts registration refreshes from users that are already registered in its database. - For a SIP Interface that is associated with the SRD, if the corresponding parameter in the SIP Interface table (SIPInterface_EnableUnAuthenticatedRegistrations) is configured to Disable or Enable, the parameter in the SRD is ignored for calls belonging to the SIP Interface.
Used By Routing Server used-by-routing-server [SIPInterface_UsedByRoutingServer]	Enables the SRD to be used by a third-party routing server for call routing decisions. [0] Not Used (default) [1] Used For more information on the third-party routing server feature, see "Centralized Third-Party Routing Server" on page 280.

17.2.1 Filtering Tables in Web Interface by SRD

When your configuration includes multiple SRDs, you can filter tables in the Web interface by a specific SRD. The filter is configured in the SRD Filter drop-down list, located on the Web interface's toolbar. The filter is applied throughout the GUI. The following figure shows the SRD Filter with an example of configured SRDs in its' drop-down list.

Figure 17-8: SRD Filter



When you select an SRD for filtering, the Web interface displays only table rows associated with the filtered SRD. In addition, if you add a new row to a table, the filtered SRD is automatically selected as the associated SRD (in the 'SRD' parameter of the Add Row dialog box). For example, if your SRD filter is set to "Comp-A" and you then add a new Proxy Set, the Proxy Set is automatically associated with SRD "Comp-A" (i.e., the 'SRD' parameter is set to "Comp-A"). All other parameters in the Add Row dialog box are also automatically set to values associated with the filtered SRD.

SRD filtering is especially useful in multi-tenant setups where multiple SRDs may be configured. In such a setup, SRD filtering eliminates configuration clutter by "hiding" SRDs that are irrelevant to the current configuration and facilitates configuration by automatically associating the filtered SRD, and other configuration elements associated with the filtered SRD, wherever applicable.

17.2.2 Multiple SRDs for Multi-tenant Deployments

The device can be deployed in a multi-tenant architecture, serving multiple customers (tenants) from a single, shared physical entity. The device's multi-tenant feature is fully scalable, offering almost "non-bleeding" partition per tenant, whereby users of one tenant can't infringe on the space of users of another tenant. The device provides per tenant configuration, monitoring, reporting, analytics, alarms and interfacing. The device is a real-time multi-tenant system that provides each tenant with optimal real-time performance, as each session received by the device is classified and processed only through the tenant's "orbit".

While some enterprises are large enough to justify a dedicated standalone device, many enterprises require only a fraction of the device's capacity and capabilities. Service providers offering SIP Trunking services can funnel multiple enterprises into a single device and thereby, reap significant cost improvements over a device-per-customer model. Tenant size in a multi-tenant architecture can vary and therefore, the instance CPU, memory and interface allocations should be optimized so as not to waste resources for small-sized tenants on the one hand, and not to allocate too many instances for a single tenant/customer on the other. For example, it would be a waste to allocate a capacity of 100 concurrent sessions to a small tenant for which 10 concurrent sessions suffice.

In a multi-tenant deployment, each tenant is represented by a dedicated SRD. The different Layer-3 networks (e.g., LAN IP-PBX users, WAN SIP Trunk, and WAN far-end users) of the tenant are represented by SIP Interfaces, which are all associated with the tenant's SRD. As related configuration entities (SIP Interfaces, IP Groups, Proxy Sets, Classification rules, and IP-to-IP Routing rules) are associated with the specific SRD, each SRD has its own logically separated configuration tables (although configured in the same

tables). Therefore, full logical separation (on the SIP application layer) between tenants is achieved by SRD.

To create a multi-tenant configuration topology that is as non-bleeding as possible, you can configure an SRD (tenant) as *Isolated* and *Shared*:

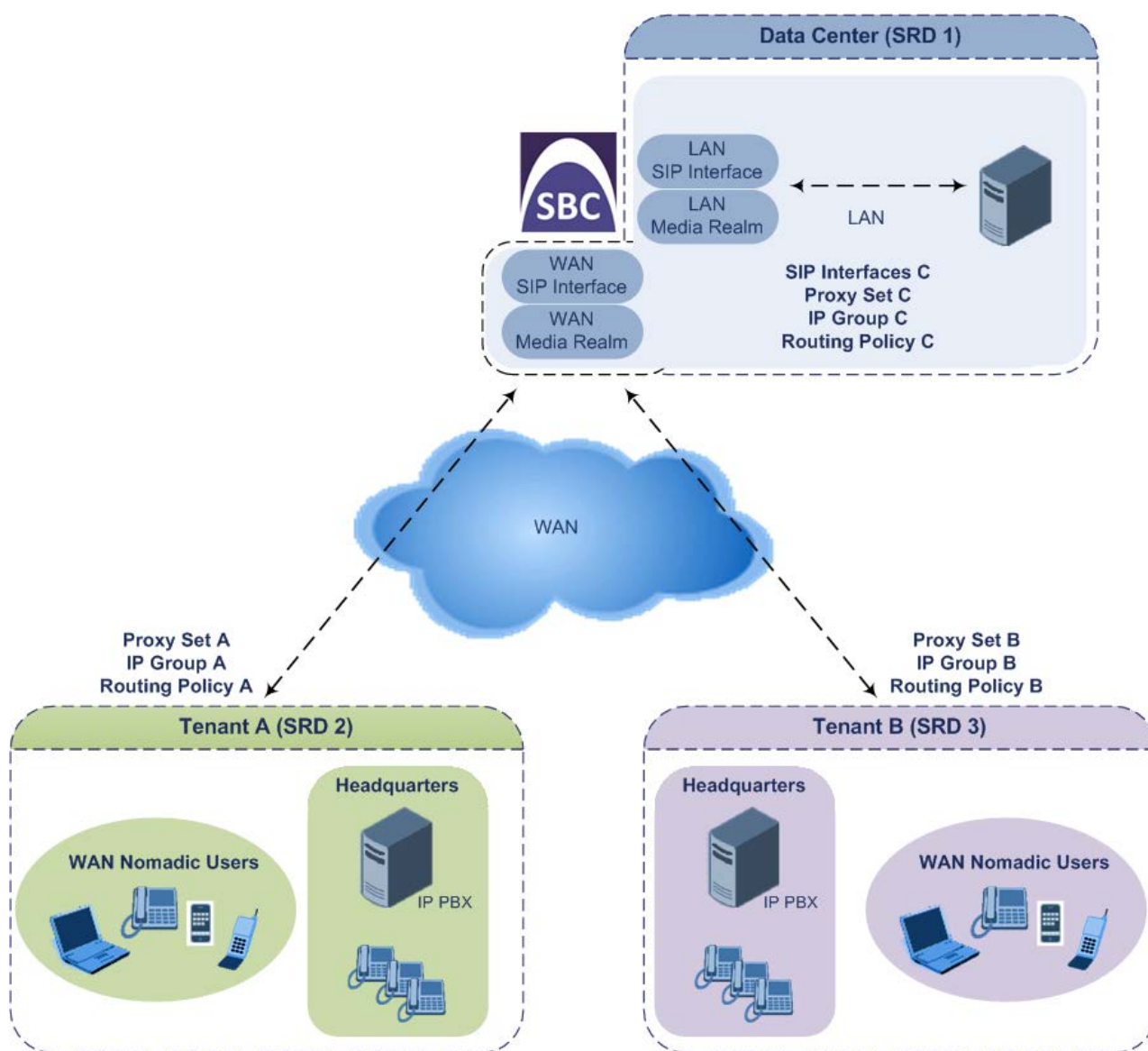
- **Isolated SRD:** An Isolated SRD has its own dedicated SIP resources (SIP Interfaces, Proxy Sets, and IP Groups). No other SRD can use the SIP resources of an Isolated SRD. Thus, call traffic of an Isolated SRD is kept separate from other SRDs (tenants), preventing any risk of traffic "leakage" with other SRDs.

Isolated SRDs are more relevant when each tenant needs its own separate (dedicated) routing "table" for non-bleeding topology. Separate routing tables are implemented using Routing Policies. In such a non-bleeding topology, routing between Isolated SRDs is not possible. This enables accurate and precise routing per SRD, eliminating any possibility of erroneous call routing between SRDs, restricting routing to each tenant's (SRD's) sphere. Configuring only one Routing Policy that is shared between Isolated SRDs is not best practice for non-bleeding environments, since it allows routing between these SRDs.

- **Shared SRD:** Isolated SRDs have their own dedicated SIP resources (SIP Interfaces, Proxy Sets, and IP Groups). This may not be possible in some deployments. For example, in deployments where all tenants use the same SIP Trunking service, or use the same SIP Interface due to limited SIP interface resources (e.g., multiple IP addresses cannot be allocated and SIP port 5060 must be used). In contrast to Isolated SRDs, a Shared SRD can share its' SIP resources with all other SRDs (Shared and Isolated). This is typically required when tenants need to use common resources. In the SIP Trunk example, the SIP Trunk would be associated with a Shared SRD, enabling all tenants to route calls with the SIP Trunk.

Another configuration entity that can be used for multi-tenant deployments is the Routing Policy. Routing Policies allow each SRD (or tenant) to have its own routing rules, manipulation rules, Least Cost Routing (LCR) rules, and/or LDAP-based routing configuration. However, not all multi-tenant deployments need multiple Routing Policies and typically, their configuration is not required. Isolated SRDs are more relevant only when each tenant requires its own dedicated Routing Policy to create separate, dedicated routing "tables"; for all other scenarios, SRDs can be Shared. For more information on Routing Policies, see "Configuring SBC Routing Policy Rules" on page 493.

The figure below illustrates a multi-tenant architecture with Isolated SRD tenants ("A" and "B") and a Shared SRD tenant ("Data Center") serving as a SIP Trunk:



To facilitate multi-tenant configuration through CLI, you can access a specific tenant "view". Once in a specific tenant view, all configuration commands apply only to the currently viewed tenant. Only table rows (indexes) belonging to the viewed tenant can be modified. New table rows are automatically associated with the viewed tenant (i.e., SRD name). The display of tables and show running-configuration commands display only rows relevant to the viewed tenant (and shared tenants). The show commands display only information relevant to the viewed tenant. To support this CLI functionality, use the following commands:

- To access a specific tenant view:

```
# srd-view <SRD name>
```

Once accessed, the tenant's name (i.e., SRD name) forms part of the CLI prompt, for example:

```
# srd-view datacenter
(srd-datacenter)#
```

- To exit the tenant view:

```
# no srd-view
```


17.2.3 Cloning SRDs

You can clone (duplicate) existing SRDs. This is especially useful when operating in a multi-tenant environment and you need to add new tenants (SRDs). The new tenants can quickly and easily be added by simply cloning one of the existing SRDs. Once cloned, all you need to do is tweak configuration entities associated with the SRD clone.

When an SRD is cloned, the device adds the new SRD clone to the next available index row in the SRD table. The SRD clone is assigned a unique name in the following syntax format: <unique ID>_<original SRD index>_CopyOf_<name or index if no name of original SRD>. For example, if you clone SRD "SIP-Trunk" at index 2, the new SRD clone is assigned the name, "36454371_2_CopyOf_SIP-Trunk".

The SRD clone has identical settings as the original SRD. In addition, all configuration entities associated with the original SRD are also cloned and these clones are associated with the SRD clone. The naming convention of these entities is the same as the SRD clones (see above) and all have the same unique clone ID ("36454371" in the example above) as the cloned SRD. These configuration entities include IP Groups, SIP Interfaces, Proxy Sets (without addresses), Classification rules, and Admission Control rules. If the Routing Policy associated with the original SRD is not associated with any other SRD, the Routing Policy is also cloned and its' clone is associated with the SRD clone. All configuration entities associated with the original Routing Policy are also cloned and these clones are associated with the Routing Policy clone. These configuration entities include IP-to-IP Routing rules, Inbound Manipulation rules, and Outbound Manipulation rules.

When any configuration entity is cloned (e.g., an IP-to-IP Routing rule) as a result of a cloned SRD, all fields of the entity's row which "point" to other entities (e.g., SIP Interface, Source IP Group, and Destination IP Group) are replaced by their corresponding clones.



Note: For some cloned entities such as SIP Interfaces, some parameter values may change. This occurs in order to avoid the same parameter having the same value in more than one table row (index), which would result in invalid configuration. For example, a SIP Interface clone will have an empty Network Interface setting. After the clone process finishes, you thus need to update the Network Interface for valid configuration.

➤ To clone an SRD:

- Web interface: In the SRD table, select an SRD to clone, and then click the **Clone** button.
- CLI:


```
(config-voip)# voip-network srd clone <SRD index that you want cloned>
```


17.2.4 Color-Coding of SRDs in Web Interface

To easily identify your configured SRDs, the Web interface displays each SRD in a unique color. The color is automatically and randomly assigned to new SRDs, and is displayed in a box alongside the name of the SRD, in tables where the SRD is configured or assigned. This is applied throughout the Web interface's GUI. The following example shows SRDs assigned with unique color codes.

Figure 17-9: Color-Coding of SRDs

Index ↕	Name
0	 Comp-A (#0)
1	 Comp-B (#1)
2	 Comp-C (#2)
3	 Comp-D (#3)



17.2.5 Automatic Configuration based on SRD

To facilitate configuration and eliminate possible flaws in configuration due to invalid associations between configuration entities, the Web interface automatically configures configuration entities based on SRD:

- If you delete an SRD (in the SRD table) that is associated with other configuration entities in other tables, the device automatically deletes the associated table rows. For example, if you delete an SRD that is associated with a Proxy Set, the device automatically deletes the Proxy Set.
- If you associate an SRD with a configuration entity in another table (i.e., other than the SRD table), the device automatically configures certain parameters of the configuration entity according to the SRD or associated SRD. For example, if you add a rule in the IP-to-IP Routing table and you select a Routing Policy, the 'Source IP Group' and 'Destination IP Group' parameters list only IP Groups that re associated with the SRD to which the Routing Policy is assigned (and IP Groups belonging to a Shared SRD, if exists).
- If your configuration setup includes only a single SRD, the device automatically selects the SRD when adding related configuration entities. For example, when adding an IP Group, the single SRD is automatically selected in the Add Row dialog box.

17.3 Configuring SIP Interfaces

The SIP Interface table lets you configure up to 500 SIP Interfaces. A SIP Interface represents a Layer-3 network in your deployment environment, by defining a local, listening port number and type (e.g., UDP), and assigning an IP network interface for SIP signaling traffic. For example, if your deployment consists of an IP PBX in the LAN, a SIP Trunk in the WAN, and remote far-end users in the WAN, you would need to configure a SIP Interface for each of these SIP entities. You can also configure various optional features for the SIP Interface such as assigning it a Media Realm, blocking calls received on the SIP Interface from users not registered with the device, and enabling direct media.

Each SIP Interface can be associated with only one SRD. As the SRD configuration entity represents your VoIP deployment SIP network (Layer 5), you need to associate your SIP Interfaces with a specific SRD in order to represent your Layer-3 networks. For most deployments (except multi-tenant deployments), your SRD represents your entire network and thus, only one SRD is required. The device provides a default SRD and in such scenarios where only a single SRD is required, your SIP Interfaces are automatically assigned to the default SRD. Therefore, there is no need to even handle SRD configuration entity.

Once configured, you can apply SIP Interfaces to calls, by assigning them to the following configuration entities in their respective tables:

- (Mandatory) Proxy Set to specify the SIP Interface for communication with the proxy server (i.e., IP Group). For more information, see "Configuring Proxy Sets" on page 360.
- IP-to-IP Routing rules for specifying the destination SIP Interface to where you want to route the call. For more information, see "Configuring SBC IP-to-IP Routing Rules" on page 480.
- Classification rules for specifying the SIP Interface as a matching characteristic of the incoming call. This is especially useful for the single SRD-configuration topology, where each SIP Interface represents a Layer-3 network (SIP entity). Therefore, classification of calls to IP Groups (SIP entities) can be based on SIP Interface. For more information, see "Configuring Classification Rules" on page 471.
- Admission Control rules to apply call admission control per SIP Interface. For more information, see "Configuring Admission Control" on page 463.
- Intrusion Detection System (IDS) for applying the IDS policy to a specific SIP Interface. For more information, see "Configuring IDS Policies" on page 176.



Note: The device terminates active calls associated with a SIP Interface in the following scenarios:

- If you delete the associated SIP Interface.
- If you edit any of the following fields of the associated SIP Interface: 'Application Type', 'UDP Port', 'TCP Port', 'TLS Port' or 'SRD' fields.
- If you edit or delete a network interface in the Interface table that is associated with the SIP Interface.

The following procedure describes how to configure SIP interfaces through the Web interface. You can also configure it through ini file (SIPInterface) or CLI (configure voip > voip-network sip-interface).

➤ **To configure a SIP Interface:**

1. Open the SIP Interface table (**Configuration** tab > **VoIP** menu > **VoIP Network** > **SIP Interface Table**).
2. Click **Add**; the following dialog box appears:

Parameter	Value
Index	1
SRD	DefaultSRD
Name	
Network Interface	None
Application Type	GW
UDP Port	5060
TCP Port	5060
TLS Port	5061
Encapsulating Protocol	No encapsulation
Media Realm	None
SBC Direct Media	Disable
TLS Context Name	default
TLS Mutual Authentication	
Block Unregistered Users	Not Configured
Max. Number of Registered Users	-1
Enable Un-Authenticated Registrations	Not configured
Enable TCP Keepalive	Disable

Buttons: Add, Cancel

3. Configure a SIP Interface according to the parameters described in the table below.
4. Click **Add**.

Table 17-5: SIP Interface Table Parameter Descriptions

Parameter	Description
Index [SIPInterface_Index]	Defines an index for the new table row. Note: Each row must be configured with a unique index.
SRD srd [SIPInterface_SRDName]	Assigns an SRD to the SIP Interface. If only one SRD is configured in the SRD table, the SRD is assigned to the SIP Interface by default. If multiple SRDs are configured in the SRD table, no value is defined. For configuring SRDs, see "Configuring SRDs" on page 332. Notes: ▪ The parameter is mandatory. ▪ You can assign the same SRD to multiple SIP Interfaces.
Name interface-name [SIPInterface_InterfaceName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, if you do not configure a name, the device automatically assigns the name "SIPInterface_<row index>" (e.g., "SIPInterface_1" when added to Index 1).

Parameter	Description
Network Interface network-interface [SIPInterface_NetworkInterface]	Assigns a Control-type IP network interface to the SIP Interface. By default, no value is defined (None). For configuring network interfaces, see "Configuring IP Network Interfaces" on page 138. Note: The parameter is mandatory.
Application Type application-type [SIPInterface_ApplicationType]	Defines the application for which the SIP Interface is used. [2] SBC = SBC application.
UDP Port udp-port [SIPInterface_UDPPort]	Defines the device's listening and source port for SIP signaling traffic over UDP. The valid range is 1 to 65534. The default is 5060. Notes: - The port must be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). 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) 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 201. - 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 tcp-port [SIPInterface_TCPPort]	Defines the device's listening port for SIP signaling traffic over TCP. The valid range is 1 to 65534. The default is 5060. Notes: - The port must be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999. - 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 tls-port [SIPInterface_TLSPort]	Defines the device's listening port for SIP signaling traffic over TLS. The valid range is 1 to 65534. The default is 5061. Notes: - The port must be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999. - Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
Encapsulating Protocol encapsulating-protocol [SIPInterface_Encapsulat]	Defines the type of incoming traffic (SIP messages) expected on the SIP Interface. [0] No Encapsulation (default) = Regular (non-WebSocket) traffic. [1] WebSocket = Traffic received on the SIP Interface is identified by

Parameter	Description
ingProtocol]	the device as WebSocket signaling traffic (encapsulated by WebSocket frames). For outgoing traffic, the device encapsulates the traffic using the WebSocket protocol (frames) on the TCP/TLS ports. For more information on WebSocket, see SIP over WebSocket on page 526. Note: WebSocket encapsulation is not supported for UDP ports.
Media Realm media-realm-name [SIPInterface_MediaRealm]	Assigns a Media Realm to the SIP Interface. By default, no value is defined (None). For configuring Media Realms, see "Configuring Media Realms" on page 323. Note: If you later delete the assigned Media Realm in the Media Realm table, this value becomes None.
SBC Direct Media intra-srd-media-anchoring [SIPInterface_SBCDirectMedia]	Enables direct media (RTP/SRTP) flow (i.e., no Media Anchoring) between endpoints associated with the SIP Interface. ▪ [0] Disable = (Default) Media Anchoring is employed, whereby the media stream traverses the device (and each leg uses a different coder or coder parameters). ▪ [1] Enable = No Media Anchoring. Media stream flows directly between endpoints (i.e., does not traverse the device - no Media Anchoring). ▪ [2] Enable when Same NAT = No Media Anchoring. Media stream flows directly between endpoints if they are located behind the same NAT. Notes: ▪ If the parameter is enabled for direct media and the two endpoints belong to the same SIP Interface, calls cannot be established if the following scenario exists: a. One of the endpoints is defined as a foreign user (for example, "follow me service") b. and one endpoint is located on the WAN and the other on the LAN. The reason for the above is that in direct media, the device does not interfere in the SIP signaling such as manipulation of IP addresses, which is necessary for calls between LAN and WAN. ▪ To enable direct media for all calls, use the global parameter SBCDirectMedia. If enabled, even if the SIP Interface is disabled for direct media, direct media is employed for calls belonging to the SIP Interface. ▪ If you enable direct media for the SIP Interface, make sure that your Media Realm provides sufficient ports, as media may traverse the device for mid-call services (e.g., call transfer). ▪ For more information on direct media, see Direct Media on page 436.

Parameter	Description
TLS Context Name tls-context-name [SIPInterface_TLSContext]	Assigns a TLS Context (SSL/TLS certificate) to the SIP Interface. The default TLS Context ("default" at Index 0) is assigned to the SIP Interface by default. Notes: - For incoming calls: The assigned 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. - For outgoing calls: The assigned TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call. For more information on TLS Contexts, see "Configuring SSL/TLS Certificates" on page 109.
TLS Mutual Authentication tls-mutual-auth [SIPInterface_TLSMutualAuthentication]	Enables TLS mutual authentication for the SIP Interface (when the device acts as a server). [-1] Not Configured = (Default) The SIPRequireClientCertificate global parameter setting is applied. [0] Disable = Device does not request the client certificate for TLS connection on the SIP Interface. [1] Enable = Device requires receipt and verification of the client certificate to establish the TLS connection on the SIP Interface.
Block Unregistered Users block-un-reg-users [SIPInterface_BlockUnRegUsers]	Enables the device to block (reject) incoming calls (INVITE requests) from unregistered users belonging to the SIP Interface. [-1] Not Configured = (Default) The corresponding parameter in the SRD table (SRD_BlockUnRegUsers) of the SRD that is associated with the SIP Interface is applied. [0] No = Calls from unregistered users are allowed. [1] Yes = Calls from unregistered users are blocked. Notes: - When the device blocks a call, it sends a SIP 500 "Server Internal Error" response to the remote end. - The parameter applies to calls belonging to a User-type IP Group. - If configured to Yes or No, the parameter overrides the 'Block Unregistered Users' parameter of the associated SRD in the SRD table.
Max. Number of Registered Users max-reg-users [SIPInterface_MaxNumberOfRegUsers]	Defines the maximum number of users belonging to the SIP Interface that can register with the device. By default, no value is defined (i.e., the number of allowed user registrations is unlimited).
Enable Un-Authenticated Registrations enable-un-auth-registrs [SIPInterface_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

Parameter	Description
	Routing table where the destination is the user's User-type IP Group (i.e., call survivability scenarios) and if the parameter is enabled. ▪ [-1] Not Configured = (Default) The corresponding parameter in the SRD table (SRD_EnableUnAuthenticatedRegistrations) of the SRD associated with the SIP Interface is applied. ▪ [0] Disable = The device rejects REGISTER requests from new users that were not authenticated by a proxy server. ▪ [1] Enable = 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. Notes: ▪ Regardless of the parameter, the device always accepts registration refreshes from users that are already registered in its database. ▪ If configured to Disable or Enable, the parameter overrides the 'Enable Un-Authenticated Registrations' parameter settings of the SRD (in the SRD table) that is associated with the SIP Interface.
Enable TCP Keepalive tcp-keepalive-enable [SIPInterface_TCPKeepaliveEnable]	Enables the TCP Keep-Alive mechanism with the IP entity on this SIP Interface. TCP keep-alive can be used, for example, to keep a NAT entry open for clients located behind a NAT server, or simply to check that the connection to the IP entity is available. ▪ [0] Disable (default) ▪ [1] Enable Note: For configuring TCP keepalive, use the following ini file parameters: TCPKeepAliveTime, TCPKeepAliveInterval, and TCPKeepAliveRetry.
Classification Failure Response Type classification_failure_response_type [SIPInterface_ClassificationFailureResponseType]	Defines the SIP response code that the device sends if a received SIP request (OPTIONS, REGISTER, or INVITE) fails 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: The 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).

Parameter	Description
Pre Classification Manipulation Set ID 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 rules, see Configuring SIP Message Manipulation on page 377. 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.
Message Policy message-policy [SIPInterface_MessagePolicyName]	Assigns a SIP message policy to the SIP interface. For configuring SIP Message Policy rules, see "Configuring SIP Message Policy Rules".
Used By Routing Server used-by-routing-server [SIPInterface_UsedByRoutingServer]	Enables the SIP Interface to be used by a third-party routing server for call routing decisions. [0] Not Used (default) [1] Used For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 280.

17.4 Configuring IP Groups

The IP Group table lets you configure up to 625 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 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 360).

You can use IP Groups 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 480). 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 369).
- Included in routing decisions by a third-party routing server. If deemed necessary for routing, the routing server can even create an IP Group. For more information, see Centralized Third-Party Routing Server on page 280.

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 313.
- **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 317.



Notes: If you delete an IP Group or modify the 'Type' or 'SRD' parameters, the device immediately terminates currently active calls associated with the IP Group. In addition, all users belonging to this IP Group are removed from the device's users database.

The following procedure describes how to configure IP Groups through the Web interface. You can also configure it through ini file (IPGroup) or CLI (configure voip > control-network ip-group).

➤ **To configure an IP Group:**

1. Open the IP Group table (**Configuration** tab > **VoIP** menu > **VoIP Network** > **IP Group Table**).
2. Click **Add**; the following dialog box appears:

The 'Add Row' dialog box is shown with the following fields and values:

Field	Value
Index	0
SRD	SRD0
Name	
Type	Server
Proxy Set	None
IP Profile	None
Media Realm	None
SIP Group Name	
QoE Profile	None
Media Enhancement Profile	None
Bandwidth Profile	None
Always Use Src Address	No
Contact User	
Local Host Name	

3. Configure an IP Group according to the parameters described in the table below.
4. Click **Add**.

Table 17-6: IP Group Table Parameter Descriptions

Parameter	Description
Common Parameters	
Index [IPGroup_Index]	Defines an index for the new table row. Note: Each row must be configured with a unique index.
SRD srd-name [IPGroup_SRDName]	Assigns an SRD to the IP Group. If only one SRD is configured in the SRD table, the SRD is assigned by default. If multiple SRDs are configured in the SRD table, no value is assigned by default. For configuring SRDs, see Configuring SRDs on page 332. Notes: - The parameter is mandatory. - For the parameter to take effect, a device reset is required.
Name name [IPGroup_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. Note: Each row must be configured with a unique name.
Type type [IPGroup_Type]	Defines the type of IP Group: [0] Server = Applicable when the destination address of the IP Group (e.g., ITSP, Proxy, IP-PBX, or Application server) is known. The address is configured by the Proxy Set that is associated with the IP Group. [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). 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 registration database with the AOR and contacts of the users. Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group. The device forwards these responses directly to the SIP users. To route a call to a registered user, a rule must be configured in the SBC IP-to-IP Routing table. The device searches the dynamic database (by using the Request-URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained from this entry and a SIP request is sent to the destination. The device also supports NAT traversal for the SIP clients located behind NAT. In this case, the device must be defined with a global IP address. [2] Gateway = In scenarios where the device receives requests to and from a gateway representing multiple users. This IP Group type is necessary for any of the following scenarios: ✓ The IP Group cannot be defined as a Server-type since its address is initially unknown and therefore, a Proxy Set cannot be configured for it. ✓ The IP Group cannot be defined as a User since the SIP Contact

Parameter	Description
	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 (i.e., IP Group is registered with the device). 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. You can view the registration status of the Gateway-type IP Group in the 'GW Group Registered Status' field, and view the IP address of the IP Group in the 'GW Group Registered IP Address' field if it is registered with the device.
Proxy Set proxy-set-id [IPGroup_ProxySetName]	Assigns a Proxy Set to the IP Group. All INVITE messages destined to the IP Group are sent to the IP address configured for the Proxy Set. For configuring Proxy Sets, see "Configuring Proxy Sets" on page 360. Notes: ▪ The Proxy Set must be associated with the same SRD as that assigned to the IP Group. ▪ You can assign the same Proxy Set to multiple IP Groups. ▪ Proxy Sets are used for Server-type IP Groups, but may in certain scenarios also be used for User-type IP Groups. For example, this is required in deployments where the device mediates between an IP PBX and a SIP Trunk, and the SIP Trunk requires SIP registration for each user that requires service. In such a scenario, the device must register all the users to the SIP Trunk on behalf of the IP PBX. This is done by using the User Info table where each user is associated with the source IP Group (i.e., the IP PBX). For configuring the User Info table, see SBC User Information for SBC User Database on page 591.
IP Profile ip-profile-name [IPGroup_ProfileName]	Assigns an IP Profile to the IP Group. By default, no value is defined (None). For configuring IP Profiles, see "Configuring IP Profiles" on page 391.
Media Realm Name media-realm-name [IPGroup_MediaRealm]	Assigns a Media Realm to the IP Group. The Media Realm determines the UDP port range and maximum sessions on a specific interface for media traffic associated with the IP Group. By default, no value is defined (None). For configuring Media Realms, see Configuring Media Realms on page 323. Notes: ▪ For the parameter to take effect, a device reset is required. ▪ If you delete a Media Realm from the Media Realm table that is assigned to the IP Group, the parameter value reverts to None.
SIP Group Name sip-group-name	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

Parameter	Description
[IPGroup_SIPGroupName]]	replaces the original host name. The valid value is a string of up to 100 characters. By default, no value is defined. Notes: - If the parameter is not configured, the value of the global parameter, ProxyName is used instead (see "Configuring Proxy and Registration Parameters" on page 373). - 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.
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: User-to-User: 00FA080019001038F725B3;encoding=hex Note: To define the Network Node Identifier of the device for Avaya UCID, use the 'Network Node ID' (NetworkNodeId) parameter.
QoE Profile qoe-profile [IPGroup_QOEProfile]	Assigns a Quality of Experience Profile rule. By default, no value is defined (None). For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 313.
Media Enhancement Profile media-enhancement-profile [IPGroup_MediaEnhancementProfile]	Assigns a Media Enhancement Profile rule. By default, no value is defined (None). For configuring Media Enhancement Profiles, see "Configuring Media Enhancement Profiles" on page 320.
Bandwidth Profile bandwidth-profile [IPGroup_BWProfile]	Assigns a Bandwidth Profile rule. By default, no value is defined (None). For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 317.
Always Use Src Address	Enables the device to always send SIP requests and responses, within a

Parameter	Description
always-use-source-addr [IPGroup_AlwaysUseSourceAddr]	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 more information on NAT traversal, see "Remote UA behind NAT" on page 159.
Contact User 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. By default, no value is defined. Notes: - The parameter is applicable only to Server-type IP Groups. - The parameter is overridden by the 'Contact User' parameter in the Account table (see "Configuring Registration Accounts" on page 369).
Local Host Name local-host-name [IPGroup_ContactName]	Defines the host name (string) that the device uses in the SIP message's Via and Contact headers. This is typically used to define an FQDN as the host name. The device uses this string for Via and Contact headers in outgoing INVITE messages sent to a specific IP Group, and the Contact header in SIP 18x and 200 OK responses for incoming INVITE messages received from a specific IP Group. The Inbound IP Routing table can be used to identify the source IP Group from where the INVITE message was received. If the parameter is not configured, these headers are populated with the device's dotted-decimal IP address of the network interface on which the message is sent. By default, no value is defined. Note: To ensure proper device handling, the parameter should be a valid FQDN.
Used By Routing Server used-by-routing-server [IPGroup_UsedByRoutingServer]	Enables the IP Group to be used by a third-party routing server for call routing decisions. [0] Not Used (default) [1] Used For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 280.
Created By Routing Server [IPGroup_CreatedByRoutingServer]	(Read-only) Indicates whether the IP Group was created by a third-party routing server: [0] No [1] Yes For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 280.
SBC Tab (SBC Application)	
SBC Operation Mode	Defines the device's operational mode for the IP Group.

Parameter	Description
sbc-operation-mode [IPGroup_SBCOperationMode]	[-1] Not Configured = (Default) [0] B2BUA = Device operates as a back-to-back user agent (B2BUA), changing the call identifiers and headers between the inbound and outbound legs. [1] Call Stateful Proxy = Device operates as a Stateful Proxy, passing the SIP message transparently between inbound and outbound legs. In other words, the same SIP dialog identifiers (tags, Call-Id and CSeq) occur on both legs (as long as no other configuration disrupts the CSeq compatibility). For more information on B2BUA and Stateful Proxy modes, see B2BUA and Stateful Proxy Operating Modes on page 425. Note: If configured, the parameter overrides the 'SBC Operation Mode' parameter in the SRD table.
Classify By Proxy Set 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 471). 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). - The parameter is applicable only to Server-type IP Groups.
SBC Client Forking Mode	Defines call forking of INVITE messages to up to five separate SIP

Parameter	Description
enable-sbc-client-forking [IPGroup_EnableSBCClientForking]	outgoing legs for User-type IP Groups. This occurs if multiple contacts are registered under the same AOR in the device's registration database. ▪ [0] Sequential = (Default) Sequentially sends the INVITE to each contact. If there is no answer from the first contact, it sends the INVITE to the second contact, and so on until a contact answers. If no contact answers, the call fails or is routed to an alternative destination, if configured. ▪ [1] Parallel = Sends the INVITE simultaneously to all contacts. The call is established with the first contact that answers. ▪ [2] Sequential Available Only = Sequentially sends the INVITE only to available contacts (i.e., not busy). If there is no answer from the first available contact, it sends the INVITE to the second contact, and so on until a contact answers. If no contact answers, the call fails or is routed to an alternative destination, if configured. Note: The device can also fork INVITE messages received for a Request-URI of a specific contact (user) registered in the database to all other users located under the same AOR as the specific contact. This is configured using the SBCSendInviteToAllContacts parameter.
Inbound Message Manipulation Set inbound-mesg-manipulation-set [IPGroup_InboundManSet]	Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the inbound leg. For configuring Message Manipulation rules, see Configuring SIP Message Manipulation on page 377. 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 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 outbound-mesg-manipulation-set [IPGroup_OutboundManSet]	Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the outbound leg. For configuring Message Manipulation rules, see Configuring SIP Message Manipulation on page 377. 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.
Msg Man User Defined String1 msg-man-user-defined-string1 [IPGroup_MsgManUserDef1]	Defines a value for the SIP user part that can be used in Message Manipulation rules configured in the Message Manipulations table. The Message Manipulation rule obtains this value from the IP Group, by using the following syntax: param.ipg.<src dst>.user-defined.<0>. The valid value is a string of up to 30 characters. By default, no value is defined. For configuring Message Manipulation rules, see Configuring SIP Message Manipulation on page 377.
Msg Man User Defined String2 msg-man-user -	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

Parameter	Description
defined-string2 [IPGroup_MsgManUserDef2]IPGroup_MsgManUserDef2]	using the following syntax: param.ipg.<src dst>.user-defined.<1>. The valid value is a string of up to 30 characters. By default, no value is defined. For configuring Message Manipulation rules, see Configuring SIP Message Manipulation on page 377.
Registration Mode 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.
Max. Number of Registered Users max-num-of-reg-users [IPGroup_MaxNumOfRegUsers]	Defines the maximum number of users in this IP Group that can register with the device. The default is -1, meaning that no limitation exists for registered users. Note: The parameter is applicable only to User-type IP Groups.
Authentication Mode authentication-mode [IPGroup_AuthenticationMode]	Defines the authentication mode. ▪ [0] User Authenticates = (Default) The device does not handle the authentication, but simply forwards 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 configured 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). For more information on Accounts, see Configuring Registration Accounts on page 369. ▪ [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 591). This is applicable only to User-type IP Groups. ✓ Authenticates SIP servers. This is applicable only to Server-type IP Groups.
Authentication Method List 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 no methods are configured, the device doesn't challenge any methods. By default, no value is defined. To define multiple SIP methods, use the backslash (\) to separate each method (e.g., INVITE\REGISTER). Note: The parameter is applicable only if the 'Authentication Mode' parameter is set to SBC as Server [2].
Username 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.

Parameter	Description
	Notes: - The parameter is applicable only to Server-type IP Groups and when the 'Authentication Mode' parameter is set to SBC as Server (i.e., authentication of servers). - To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.
Password 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: - The 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.
Source URI Input 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: - 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, 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 the parameter setting.
Destination URI Input 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 characteristic (input) to classify the incoming INVITE to an IP Group (source IP Group) in the Classification table.

Parameter	Description
	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 the 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.
SBC PSAP Mode sbc-psap-mode [IPGroup_SBCPSAPMode]]	Enables E9-1-1 emergency call routing in a Microsoft Lync Server environment. ▪ [0] Disable (default) ▪ [1] Enable For more information, see Enhanced 9-1-1 Support for Lync Server on page 298.
DTLS Context dtls-context [IPGroup_DTLSContext]	Assigns a TLS Context (certificate) to the IP Group, which is used for DTLS sessions (handshakes) with the IP Group. By default, no value is defined (None). For configuring TLS Contexts, see Configuring TLS Certificate Contexts

Parameter	Description
	on page 109.
Route Using Request URI Port use-request-uri-port [IPGroup_SBCRouteUsingRequestURIPort]	Enables the device to use the port indicated in the Request-URI of the incoming message as the destination port when routing the message to the IP Group. The device uses the IP address (and not port) that is configured for the Proxy Set associated with the IP Group. The parameter thus allows the device to route calls to the same server (IP Group), but different port. ▪ [0] Disable = (Default) The port configured for the associated Proxy Set is used as the destination port. ▪ [1] Enable = The port indicated in the Request-URI of the incoming message is used as the destination port.
GW Group Status	
GW Group Registered IP Address	(Read-only field) Displays the IP address of the IP Group entity (gateway) if registered with the device; otherwise, the field is blank. Note: The field is applicable only to Gateway-type IP Groups (i.e., the 'Type' parameter is configured to Gateway).
GW Group Registered Status	(Read-only field) Displays whether the IP Group entity (gateway) is registered with the device ("Registered" or "Not Registered"). Note: The field is applicable only to Gateway-type IP Groups (i.e., the 'Type' parameter is configured to Gateway).

17.5 Configuring Proxy Sets

The Proxy Sets table lets you configure up to 625 Proxy Sets. A Proxy Set defines the address and transport type (e.g., UDP or TCP) of a SIP server (e.g., SIP proxy and SIP registrar server). The Proxy Set represents the destination (address) of the IP Group configuration entity. 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 (Proxy Hot-Swap feature) 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.) Each Proxy Set can be assigned a specific SSL/TLS certificate the (TLS Context), enabling you to use different TLS certificates per SIP entity (IP Group). In addition, each Proxy Set must be assigned a SIP Interface (and SRD), which determines, amongst others, the device's local network interface through which communication with the Proxy Set is done.

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 645.

To use a configured Proxy Set, you need to assign it to an IP Group in the IP Group table (see "Configuring IP Groups" on page 348). When the device sends INVITE messages to an IP Group, it sends it to the address configured for the Proxy Set. You can assign the same Proxy Set to multiple IP Groups (belonging to the same SRD).

You can also enable the device to classify 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, the device classifies the SIP dialog as belonging to the IP Group that is associated with the Proxy Set.



Notes:

- It is recommended to classify incoming SIP dialogs to IP Groups, based on the Classification table (see Configuring Classification Rules on page 471) instead of based on Proxy Set.
- You can view the device's connectivity status with proxy servers in the Tel-to-IP Routing table, for Tel-to-IP routing rules whose destination is an IP Group that is associated with a Proxy Set. The status is only displayed for Proxy Sets enabled with the Proxy Keep-Alive feature.

The Proxy Set is configured using two tables, one a "child" of the other:

- Proxy Sets table: Defines the attributes of the Proxy Set such as associated SIP Interface and redundancy features - ini file parameter, ProxySet or CLI command, configure voip > voip-network proxy-set

- Proxy Set Address table ("child"): Defines the addresses of the Proxy Set - table ini file parameter, ProxyIP or CLI command, configure voip > voip-network proxy-ip > proxy-set-id

➤ **To configure a Proxy Set:**

1. Open the Proxy Sets table (**Configuration** tab > **VoIP** menu > **VoIP Network** > **Proxy Sets Table**).
2. Click **Add**; the following dialog box appears:

Figure 17-10: Proxy Sets Table - Add Row Dialog Box

3. Configure a Proxy Set according to the parameters described in the table below.
4. Click **Add**.
5. Select the index row of the Proxy Set that you added, and then click the **Proxy Address Table** link located below the table; the Proxy Address table opens.
6. Click **Add**; the following dialog box appears:

Figure 17-11: Proxy Address Table - Add Row Dialog Box

7. Configure the address of the Proxy Set according to the parameters described in the

table below.

8. Click **Add**.

Table 17-7: Proxy Sets Table and Proxy Address Table Parameter Description

Parameter	Description
Proxy Sets Table	
Index configure voip > voip-network proxy-set [ProxySet_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
SRD voip-network proxy-set > srd- id [ProxySet_SRDName]	Assigns an SRD to the Proxy Set. Notes: - The parameter is mandatory and must be configured first before you can configure the other parameters in the table. - To configure SRDs, see Configuring SRDs on page 332.
Name proxy-name [ProxySet_ProxyName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. Note: - Each row must be configured with a unique name. - The value cannot include a "/" forward slash.
SBC IPv4 SIP Interface sbcipv4-sip-int- name [ProxySet_SBCIPv4SI PInterfaceName]	Assigns an IPv4-based SIP Interface for SBC calls to the Proxy Set. Notes: - At least one SIP Interface must be assigned to the Proxy Set. - The parameter appears only if you have configured a network interface with an IPv4 address in the Interface table (see Configuring IP Network Interfaces on page 138). - To configure SIP Interfaces, see "Configuring SIP Interfaces" on page 342.
SBC IPv6 SIP Interface sbcipv6-sip-int- name [ProxySet_SBCIPv6SI PInterfaceName]	Assigns an IPv6-based SIP Interface for SBC calls to the Proxy Set. Notes: - At least one SIP Interface must be assigned to the Proxy Set. - The parameter appears only if you have configured a network interface with an IPv6 address in the Interface table.
Proxy Keep-Alive proxy-enable- keep-alive [ProxySet_EnableProx yKeepAlive]	Enables the device's Proxy Keep-Alive feature, which checks communication with the proxy server. [0] Disable (default). [1] Using OPTIONS = Enables the Proxy Keep-Alive feature using SIP OPTIONS messages. The device sends an OPTIONS message every user-defined interval, configured by the 'Proxy Keep-Alive Time' parameter (in this table). If the device receives a SIP response code that is configured in the 'Keep-Alive Failure Responses' parameter (in this table), 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 feature using SIP REGISTER messages. The device sends a REGISTER message every user-defined interval, configured by the SBCProxyRegistrationTime parameter. Any SIP response from the proxy - success (200 OK) or

Parameter	Description
	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).
Proxy Keep-Alive Time proxy-keep-alive-time [ProxySet_ProxyKeepAliveTime]	Defines the interval (in seconds) between keep-alive messages sent by the device when the Proxy Keep-Alive feature is enabled (see the 'Proxy Keep-Alive' parameter in this table). The valid range is 5 to 2,000,000. The default is 60. Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options.
Success Detection Retries success-detect-retries [ProxySet_SuccessDetectionRetries]	Defines the minimum number of consecutive, successful keep-alive messages that the device sends to an offline proxy, before 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.
Redundancy Mode proxy-redundancy-mode [ProxySet_ProxyRedundancyMode]	Determines whether the device switches from a redundant proxy to the primary proxy when the primary proxy becomes available again. [-1] Not configured = (Default) Proxy redundancy method is according to the settings of the global parameter, ProxyRedundancyMode. [0] Parking = The device continues operating with the redundant (now active) proxy even if the primary proxy returns to service. If the redundant proxy subsequently becomes unavailable, the device operates with the next configured redundant proxy. [1] Homing = The device always attempts to operate with the primary proxy. The device switches back to the primary proxy whenever it

Parameter	Description
	becomes available. Notes: - To enable this functionality, you must also enable the Proxy Keep-Alive feature (see the 'Proxy Keep-Alive' parameter in this table). - The Homing option can only be used if the 'Proxy Keep-Alive' parameter is set to Using Options.
Proxy Load Balancing Method proxy-load-balancing-method [ProxySet_ProxyLoadBalancingMethod]	Enables load balancing between proxy servers of the Proxy Set. [0] Disable = (Default) Disables proxy load balancing. [1] Round Robin = A list of all possible proxy IP addresses is compiled. This list includes all IP addresses of the Proxy Set after DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive feature (enabled by the 'Proxy Keep-Alive' and 'Proxy Keep-Alive Time' parameters in this table) 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 address list is refreshed every user-defined interval (see the ProxyIPListRefreshTime parameter). If a change in the order of the IP address 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: ✓ More than one IP address has been configured for the Proxy Set. ✓ The proxy address is not configured as an FQDN (only IP address). ✓ SRV is disabled (see the DNSQueryType parameter). ✓ The SRV response includes several records with a different Priority value.
Min. Active Servers for Load Balancing min-active-serv-lb [ProxySet_MinActiveServersLB]	Defines the minimum number of proxies in the Proxy Set that must be online for the device to consider the Proxy Set as online, when proxy load balancing is used. The valid value is 1 to 15. The default is 1. Note: The parameter is applicable only if proxy load balancing is enabled (see the 'Proxy Load Balancing Method' parameter, above).

Parameter	Description
DNS Resolve Method dns-resolve-method [ProxySet_DNSResolveMethod]	Defines the DNS query record type for resolving the proxy server's host name (FQDN) into an IP address(es). ▪ [-1] = DNS resolution method is according to the settings of the global parameter, ProxyDNSQueryType. ▪ [0] A-Record = (Default) DNS A-record query is used to resolve DNS to IP addresses. ▪ [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. ▪ [3] MS-Lync = SRV query as required by Microsoft when the device is deployed in a Microsoft Lync environment. The device sends a special SRV query to the DNS server according to the transport protocol configured in the 'Transport Type' parameter (described later in this section): ✓ TLS: "_sipinternaltls_tcp.<domain>" and "_sip_tls.<domain>". For example, if the configured domain name (in the 'Proxy Address' parameter) is "ms-server.com", the device queries for "_sipinternaltls_tcp.ms-server.com" and "_sip_tls.ms-server.com". ✓ TCP: "_sipinternal_tcp.<domain>" and "_sip_tcp.<domain>". ✓ Undefined: "_sipinternaltls_tcp.<domain>", "_sipinternal_tcp.<domain>", "_sip_tls.<domain>" and "_sip_tcp.<domain>". 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) to resolve into IP addresses. Note: An SRV query can return up to four host names. For each host name, the subsequent DNS A-record query can resolve 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 IP addresses, it uses the first 30 IP addresses in the received list and ignores the rest.
Proxy Hot Swap is-proxy-hot-swap [ProxySet_IsProxyHotSwap]	Enables the Proxy Hot-Swap feature, whereby the device switches to a redundant proxy upon a failure in the primary proxy (no response is received). ▪ [0] No = (Default) Disables the Proxy Hot-Swap feature. If a failure occurs in the primary proxy, the device does not connect with any other address (proxy) configured for the Proxy Set. ▪ [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 491) for SBC, the device assumes that the proxy is down and sends the message to the next available proxy (address) in the list.

Parameter	Description
Keep-Alive Failure Responses 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 response code is defined. If no response code is configured, or if response codes received are not those configured, the proxy is considered "alive". Note: The SIP 200 response code is not supported for this feature.
Classification Input classification-input [ProxySet_ClassificationInput]	Defines how the device classifies incoming IP calls to the Proxy Set. [0] IP Only = (Default) Classifies calls to the Proxy Set according to IP address only. [1] IP + Port + Transport = Classifies calls to the Proxy Set according to IP address, port, and transport type. Note: The parameter is applicable only if the IP Group table's parameter, 'Classify by Proxy Set' is set to Enable (see Configuring IP Groups on page 348).
TLS Context Index tls-context-index [ProxySet_TLSContextName]	Assigns a TLS Context (SSL/TLS certificate) to the Proxy Set. By default, no TLS Context is assigned. If you assign a TLS Context, the TLS Context is used as follows: Incoming calls: If the 'Transport Type' parameter (in this table) is set to TLS and the incoming call is successfully classified to an IP Group based on the Proxy Set, this TLS Context is used. If the 'Transport Type' parameter is set to UDP or classification to this Proxy Set fails, the TLS Context is not used. Instead, the device uses the TLS Context configured for the SIP Interface (see "Configuring SIP Interfaces" on page 342) used for the call; otherwise, the default TLS Context (ID 0) is used. Outgoing calls: If the 'Transport Type' parameter is set to TLS and the outgoing call is sent to an IP Group that is associated with this Proxy Set, this TLS Context is used. Instead, the device uses the TLS Context configured for the SIP Interface 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 configuring TLS Contexts, see "Configuring TLS Certificate Contexts" on page 109.
Proxy Address Table configure voip > voip-network proxy-ip > proxy-set-id	
Index proxy-ip-index [ProxyIp_ProxyIpIndex]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Proxy Address proxy-address [ProxyIp_IpAddress]	Defines the address of the proxy. 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 using 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
	Note: You can configure the device to use the port indicated in the Request-URI of the incoming message, instead of the port configured for the parameter. To enable this, use the IPGroup_SBCRouteUsingRequestURIPort parameter for the IP Group that is associated with the Proxy Set (Configuring IP Groups on page 348).
Transport Type transport-type [ProxyIp_TransportType]	Defines the transport type for communicating with the proxy. ▪ [0] UDP ▪ [1] TCP ▪ [2] TLS ▪ [-1] = (Default) The transport type is according to the settings of the global parameter, SIPTransportType.

18 SIP Definitions

This section describes configuration of various SIP-related functionalities.

18.1 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 719.

18.2 Configuring Registration Accounts

The Account table lets you configure up to 625 Accounts. An Account defines registration information for registering and authenticating (digest) IP Groups (e.g., IP PBX) with a "serving" IP Group (e.g., ITSP).

The device initiates registration with a "serving" IP Group on behalf of the "served" IP Group. Therefore, Accounts are typically required when the "served" IP Group is unable to register or authenticate itself for whatever reason. Registration information includes 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 IP Group can register to more than one 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, username/password, host name, and contact user values.

Authentication is typically required for INVITE messages sent to the "serving" IP Group. If the device receives a SIP 401 (Unauthorized) in response to a sent INVITE, the device checks for a matching "serving" and "served" entry in the Account table. If a matching row exists, the device authenticates the INVITE by providing the corresponding MD5 authentication username and password to the "serving" IP Group.



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 through the Web interface. You can also configure it through ini file (Account) or CLI (configure voip > sip-definition account).

➤ **To configure an Account:**

1. Open the Account table (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Account Table**).

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

Figure 18-1: Account Table - Add Row Dialog Box

3. Configure an account according to the parameters described in the table below.

4. Click **Add**.

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 645.

Table 18-1: Account Table Parameter Descriptions

Parameter	Description
Index	Defines an index for the new table row. Note: Each row must be configured with a unique index.
Served IP Group served-ip-group-name [Account_ServedIPGroupName]	Defines the IP Group (e.g., IP-PBX) that you want to register and/or authenticate upon its behalf. Note: By default, all IP Groups are displayed. However, if you filter the Web display by SRD (using the SRD Filter box), only IP Groups associated with the filtered SRD are displayed.
Serving IP Group serving-ip-group-name [Account_ServingIPGroupName]	Defines the IP Group (<i>Serving IP Group</i>) to where the device sends the SIP REGISTER requests (if enabled) for registration and authentication (of the Served IP Group). Note: By default, only IP Groups associated with the SRD to which the Served IP Group is associated are displayed, as well as IP Groups of Shared SRDs. However, if you filter the Web display by SRD (using the SRD Filter box), only IP Groups associated with the filtered SRD are displayed, as well as IP Groups of Shared SRDs.

Parameter	Description
User Name user - name [Account_Username]	Defines the digest MD5 Authentication username. The valid value is a string of up to 50 characters.
Password password [Account_Password]	Defines the digest MD5 Authentication password. The valid value is a string of up to 50 characters.
Host Name 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 the parameter is not configured or if registration fails, the 'SIP Group Name' parameter value configured in the IP Group table is used instead.
Register register [Account_Register]	Enables registration. [0] No= (Default) The device only performs authentication (not registration). Authentication is typically done for INVITE messages sent to the "serving" IP Group. If the device receives a SIP 401 (Unauthorized) in response to a sent INVITE, the device checks for a matching "serving" and "served" entry in the table. If a matching row exists, the device authenticates the INVITE by providing the corresponding MD5 authentication username and password to the "serving" IP Group. [1] Regular = Regular registration process. For more information, see "Regular Registration Mode" on page 372. [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 372. Note: The account registration is not affected by the IsRegisterNeeded parameter.
Contact User 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 the 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 application - type [Account_ApplicationType]	Defines the application type: [2] SBC = SBC application.

18.2.1 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.227.005
Content-Length: 0
```

18.2.2 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.

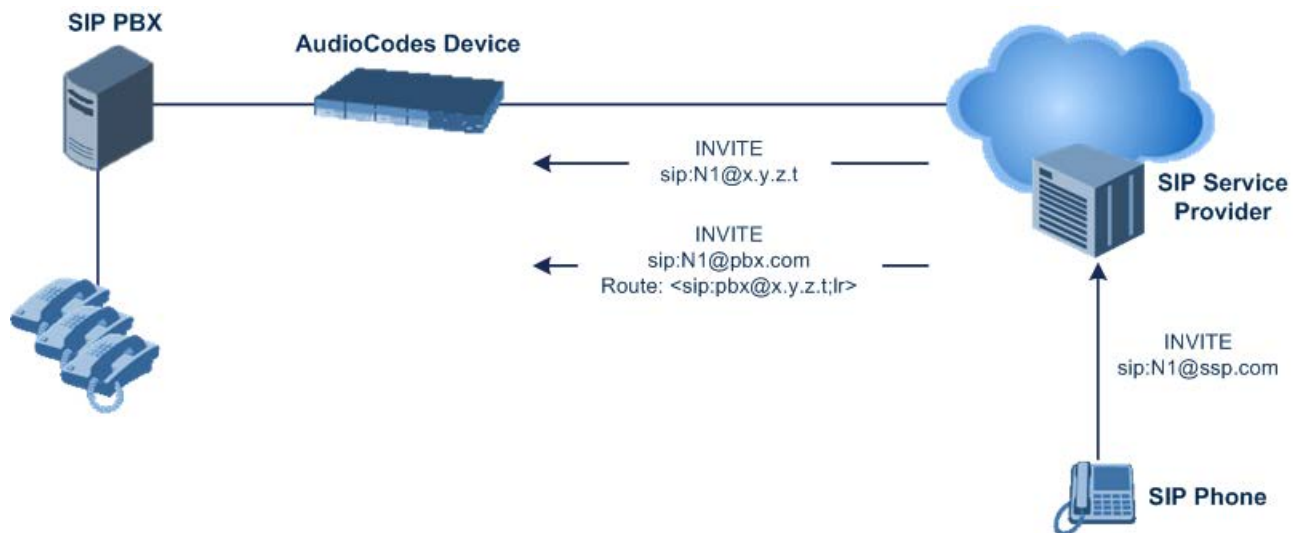
The SIP REGISTER request sent by the device for GIN registration with a SIP server provider contains the Require and Proxy-Require headers. These headers contain the token 'gin'. The Supported header contains the token 'path' and the URI in the Contact header contains the parameter 'bnc' without a user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



18.3 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 719.



Note: To view the registration status of endpoints with a SIP Registrar/Proxy server, see "Viewing Registration Status" on page 645.

➤ **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 369)

Click the **Proxy Set Table**  button to Open the Proxy Sets table 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 360 for a description of this page).

18.3.1 SIP Message Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261. 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 2600 SBC/v.6.80A.227.005
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 2600
SBC/v.6.80A.227.005
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
```

18.4 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 then 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 342).
 - Post-classification Process: Message manipulation can be done on inbound and/or outbound SIP messages after the call has been successfully classified. Manipulation occurs only after the routing process - inbound message manipulation is done first, then outbound number manipulation (see Configuring IP-to-IP Outbound Manipulations on page 504), and then outbound message manipulation. For viewing the call processing flow, see Call Processing of SIP Dialog Requests on page 428. 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 348).

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

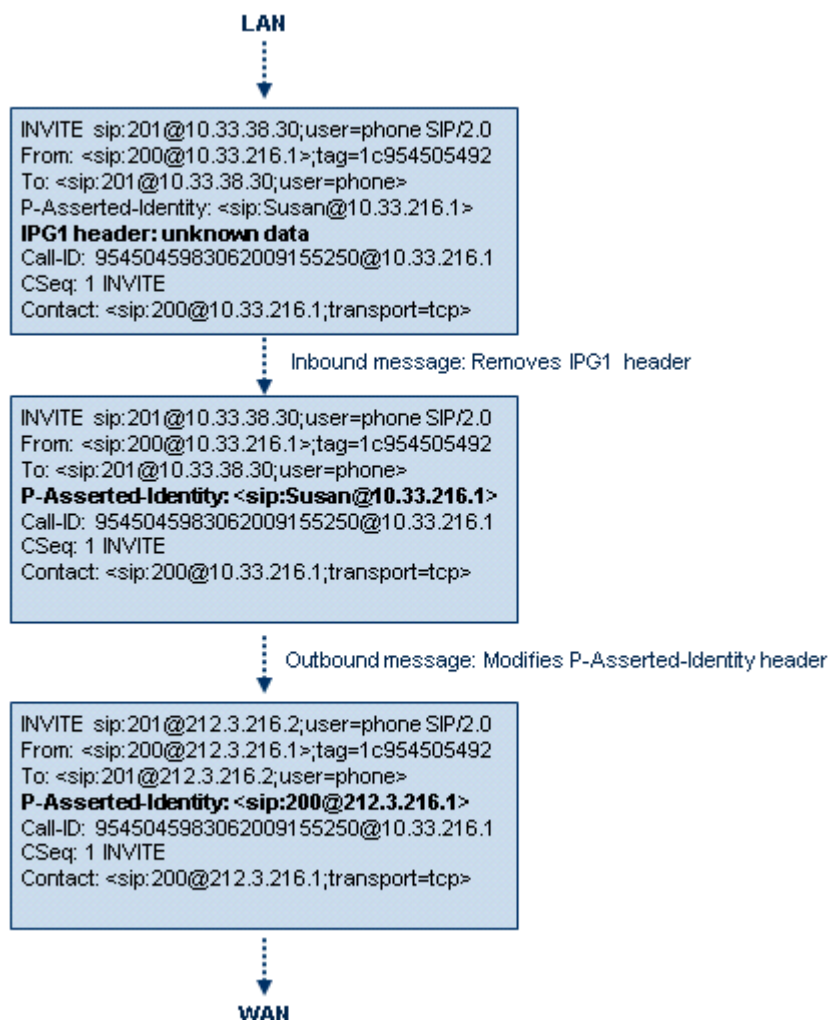
- Multiple manipulation rules using the same condition. The following figure shows a configuration example where rules 1 and 2 ('Row Rule' configured to **Use Previous Condition**) use the condition configured for rule 0 ('Row Rule' configured to **Use Current Condition**). For more information, see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

Figure 18-2: Configuration Example of Message Manipulation Rules using Same Condition

Index	Name	Manipulation Set ID	Message Type	Condition	Action Subject	Action Type	Action Value	Row Role
0	To header urgent	0	invite.request	header.request-uri, header.to		Modify	header.to + 'urgent=1'	Use Current Condition
1	Add emergency	0			header.priority	Add	'emergency'	Use Previous Condition
2	User-Agent	0			header.user-agent	Modify	'trunk-a'	Use Previous Condition

The figure below illustrates a SIP message manipulation example:

Figure 18-3: 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 348) 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 through the Web interface. You can also configure it through ini file (MessageManipulations) or CLI (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 18-4: Message Manipulations Table - Add Row Dialog Box

3. Configure a Message Manipulation rule according to the parameters described in the table below.

4. Click **Add**.

An example of configured message manipulation rules are shown in the figure below:

Figure 18-5: 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='Unkown'	header.from.url.user	Modify	param.ipg.src.user
4		2	invite.request		header.priority	Remove	
Page 1 of 1 Show 10 records per page View 1 - 5 of 5							

- Index 0: Adds the suffix ".com" to the host part of the To header.
- Index 1: Changes the user part of the From header to the user part of the P-Asserted-ID.
- Index 2: Changes the user part of the SIP From header to "200".
- Index 3: If the user part of the From header equals "unknown", then it is changed according to the srcIPGroup call's parameter.
- Index 4: Removes the Priority header from an incoming INVITE message.

Table 18-2: Message Manipulations Parameter Descriptions

Parameter	Description
Index [MessageManipulations_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name manipulation-name [MessageManipulations_ManipulationName]	Defines an arbitrary name to easily identify the rule. The valid value is a string of up to 40 characters.
Manipulation Set ID 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 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 condition	Defines the condition that must exist for the rule to apply. The valid value is a string (case-insensitive).

Parameter	Description
[MessageManipulations_ Condition]	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 action-subject [MessageManipulations_ ActionSubject]	Defines the SIP header upon which the manipulation is performed. The valid value is a string (case-insensitive).
Action Type action-type [MessageManipulations_ ActionType]	Defines the type of manipulation. ▪ [0] Add (default) = Adds new header/param/body (header or parameter elements). ▪ [1] Remove = Removes header/param/body (header or parameter elements). ▪ [2] Modify = Sets element to the new value (all element types). ▪ [3] Add Prefix = Adds value at the beginning of the string (string element only). ▪ [4] Add Suffix = Adds value at the end of the string (string element only). ▪ [5] Remove Suffix = Removes value from the end of the string (string element only). ▪ [6] Remove Prefix = Removes value from the beginning of the string (string element only). ▪ [7] Normalize = Removes unknown SIP message elements before forwarding the message.
Action Value action-value [MessageManipulations_ ActionValue]	Defines a value that you want to use in the manipulation. The default value is a string (case-insensitive) in the following syntax: ▪ string/<message-element>/<call-param> + ▪ string/<message-element>/<call-param> For example: ▪ 'itsp.com' ▪ header.from.url.user ▪ param.call.dst.user ▪ param.call.dst.host + '.com' ▪ param.call.src.user + '<' + header.from.url.user + '@' + header.p-asserted-id.url.host + '>' Note: Only single quotation marks must be used.
Row Role 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

Parameter	Description
	and 5 use the condition 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.

18.5 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 342).

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 through the Web interface. You can also configure it through ini file (MessagePolicy) or CLI (configure voip > sbc message-policy).

➤ **To configure SIP Message Policy rules:**

1. Open the Message Policy table (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Msg Policy & Manipulation** > **Message Policy Table**).
2. Click **Add**; the following dialog box appears:

Figure 18-6: Message Policy Table - Add Row Dialog Box

The 'Add Row' dialog box is shown with the following parameters:

Parameter	Value
Index	0
Name	
Max Message Length	32768
Max Header Length	512
Max Body Length	1024
Max Num Headers	32
Max Num Bodies	8
Send Rejection	Policy Reject
Method List	
Method List Type	Policy Whitelist
Body List	
Body List Type	Policy WhiteList

3. Configure a Message Policy rule according to the parameters described in the table below.
4. Click **Add**.

Table 18-3: Message Policy Table Parameter Descriptions

Parameter	Description
Index [MessagePolicy_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [MessagePolicy_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. Note: Each row must be configured with a unique name.
Max Message Length 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 max-header-length [MessagePolicy_MaxHeaderLength]	Defines the maximum SIP header length. The valid value is up to 512 characters. The default is 512.

Parameter	Description
Max Body Length 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 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 a 200 OK response. If it receives more than this, it responds with a SIP 513 'Message Too Large' response.
Max Num Bodies 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 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, 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 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 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 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 the parameter are case-sensitive.
Body List Type 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.

19 Coders and Profiles

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

19.1 Configuring Default Coders

The Coders table lets you configure up to 21 voice coders for the device. This is the default Coder Group, which is used by the device for all calls, unless a different Coder Group, configured in the Coder Group Settings table (see "Configuring Coder Groups" on page 389) is assigned to specific calls, using IP Profiles.

Each coder can be configured with packetization time (ptime), bit rate, payload type, and silence suppression. The first coder configured in the table has the highest priority and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder in the table, and so on.



Notes:

- Some coders are license-dependent and are available only if purchased from AudioCodes and included in the Software License Key installed on your device. For more information, contact your AudioCodes sales representative.
- Only the packetization time of the first coder in the coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined. The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the packetization time is assigned the default value.
- The value of several fields is hard-coded according to common standards (e.g., payload type of G.711 U-law is always 0). Other values can be set dynamically. If no value is specified for a dynamic field, a default value is assigned. If a value is specified for a hard-coded field, the value is ignored.
- The G.722 coder provides Packet Loss Concealment (PLC) capabilities, ensuring higher voice quality.
- Opus coder:
 - ✓ If one leg uses a narrowband coder (e.g., G.711) and the other leg uses the Opus coder, the device maintains the narrowband coder flavor by using the narrowband Opus coder. Alternatively, if one leg uses a wideband coder (e.g., G.722) and the other leg uses the Opus coder, the device maintains the wideband coder flavor by using the wideband Opus coder.
- For information on V.152 and implementation of T.38 and VBD coders, see "Supporting V.152 Implementation" on page 198.

The following procedure describes how to configure the Coders table through the Web interface. You can also configure it through ini file (CodersGroup) or CLI (configure voip > coders-and-profiles coders-group).

➤ **To configure coders:**

1. Open the Coders page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**).

Figure 19-1: Coders Table Page

Coder Name	Packetization Time	Rate	Payload Type	Silence Suppression	Coder Specific
G.711A-law	20	64	8	Disabled	

2. Configure coders 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 19-1: Coders Table Parameter Descriptions

Parameter	Description
Coder Name name [CodersGroup0_Name]	Defines the coder. Note: Each coder type (e.g., G.729) can be configured only once in the table.
Packetization Time p-time [CodersGroup0_pTime]	Defines the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
Rate rate [CodersGroup0_rate]	Defines the bit rate (in kbps) for the coder.
Payload Type payload-type [CodersGroup0_PayloadType]	Defines the payload type if the payload type (i.e., format of the RTP payload) for the coder is dynamic.
Silence Suppression silence-suppression [CodersGroup0_Sce]	Enables silence suppression for the coder. ▪ [0] Disable (Default) ▪ [1] Enable ▪ [2] Enable w/o Adaptation =Applicable only to G.729. Notes: ▪ If G.729 is configured with silence suppression disabled, the device includes 'annexb=no' in the SDP of the relevant SIP messages. If silence suppression is enabled or set to Enable w/o Adaptations, 'annexb=yes' is included.

Parameter	Description
Coder Specific coder-specific [CodersGroup0_CoderSpecific]	Defines additional settings specific to the coder. Currently, the parameter is applicable only to the AMR coder and is used to configure the payload format type. [0] = Bandwidth Efficient [1] = Octet Aligned (default) Note: The AMR payload type can be configured globally using the AmrOctetAlignedEnable parameter. However, the Coder Group configuration overrides the global parameter.

The table below lists the supported coders:

Table 19-2: Supported Coders

Coder Name	Packetization Time (msec)	Rate (kbps)	Payload Type	Silence Suppression
G.711 A-law [g711Alaw64k]	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	8	[0] Disable [1] Enable
G.711 U-law [g711Ulaw64k]	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	0	[0] Disable [1] Enable
G.711A-law_VBD [g711AlawVbd]	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	8 or Dynamic (default 118)	N/A
G.711U-law_VBD [g711UlawVbd]	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	0 or Dynamic (default 110)	N/A
G.722 [g722]	20 (default), 40, 60, 80, 100, 120	64 (default)	9	N/A
G.723.1 [g7231]	30 (default), 60, 90, 120, 150	[0] 5.3 (default) [1] 6.3	4	[0] Disable [1] Enable
G.726 [g726]	10, 20 (default), 30, 40, 50, 60, 80	[0] 16 [1] 24 [2] 32 (default) [3] 40	Dynamic (default 2)	[0] Disable [1] Enable
G.729 [g729]	10, 20 (default), 30, 40, 50, 60, 80, 100	8	18	[0] Disable [1] Enable [2] Enable w/o Adaptations

Coder Name	Packetization Time (msec)	Rate (kbps)	Payload Type	Silence Suppression
AMR [Amr]	20 (default)	[0] 4.75 [1] 5.15 [2] 5.90 [3] 6.70 [4] 7.40 [5] 7.95 [6] 10.2 [7] 12.2 (default)	Dynamic	[0] Disable [1] Enable
AMR-WB [Amr-WB]	20 (default)	[0] 6.6 [1] 8.85 [2] 12.65 [3] 14.25 [4] 15.85 [5] 18.25 [6] 19.85 [7] 23.05 [8] 23.85 (default)	Dynamic	[0] Disable [1] Enable
iLBC [iLBC]	20 (default), 40, 60, 80, 100, 120	15 (default)	Dynamic (default 65)	[0] Disable [1] Enable
	30 (default), 60, 90, 120	13		
silk-nb [Silk-8KHz]	20 (default), 40, 60, 80, and 100	8	Dynamic (default 76)	N/A
silk-wb [Silk-16KHz]	20 (default), 40, 60, 80, and 100	16	Dynamic (default 77)	N/A
T.38 [t38fax]	N/A	N/A	N/A	N/A
T.38 Version 3 [t38fax]	-	-	-	-
OPUS [Opus]	20 (default), 40, 60, 80, 120	N/A	Dynamic (default 111)	N/A

19.2 Configuring Coder Groups

The Coder Group Settings table lets you configure up to 21 *Coder Groups*. A Coder Group is a set of configured coders (coder type, packetization time, rate, payload type, and silence suppression). Each Coder Group can include up to 10 coders.

The first coder in the Coder Group has the highest priority and is used by the device whenever possible. If the remote side cannot use the first coder, the device attempts to use the next coder in the Coder Group, and so on.

To define coders for specific calls, you can configure a Coder Group with the necessary coders and then assign the Coder Group to the calls using IP Profiles (see "Configuring IP Profiles" on page 391).

You can also use Coder Groups for audio coder transcoding of SBC calls. If two SIP entities need to communicate, but one does not support a coder required by the other, the device can add the required coder to the SDP offer. The added coder is referred to as an extension coder. For more information on extension coders, see Coder Transcoding on page 438.

To apply a Coder Group for transcoding to a SIP entity:

1. Configure a Coder Group in the Coder Group Settings table (see description below).
2. In the IP Profile associated with the SIP entity (see Configuring IP Profiles on page 391):
 - Assign the Coder Group (using the `IpProfile_SBCExtensionCodersGroupID` parameter).
 - Enable the use of the Coder Group for transcoding (by configuring the `IpProfile_SBCAllowedCodersMode` parameter to `Restriction` or `Restriction and Preference`).

**Notes:**

- To define coders for calls that are not assigned a specific Coder Group using IP Profiles, see "Configuring Default Coders" on page 385. This group of coders is termed the *Default Coder Group*.
- For a list of supported coders, see "Configuring Default Coders" on page 385.

The following procedure describes how to configure the Coders table through the Web interface. You can also configure it through ini file (CodersGroup) or CLI (configure voip > coders-and-profiles coders-group).

➤ **To configure a Coder Group:**

1. Open the Coder Group Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders Group Settings**).

Figure 19-2: Coder Group Settings Page

Coder Group ID
1 ▼

Coder Name	Packetization Time	Rate	Payload Type	Silence Suppression	Coder Specific
G.711A-law ▼	20 ▼	64 ▼	8	Disabled ▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	
▼	▼	▼		▼	

2. Configure the Coder Group according to the parameters described in the table below.
3. Click **Add**, and then reset the device with a save ("burn") to flash memory.

Table 19-3: Coder Group Settings Table Parameter Descriptions

Parameter	Description
Coder Group ID [CodersGroupX_Index]	Defines an ID for the Coder Group.
Coder Name name [CodersGroupX_Name]	Defines the coder type. Note: Each coder type (e.g., G.729) can be configured only once in the table.
Packetization Time p-time [CodersGroupX_pTime]	Defines the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet.
Rate rate [CodersGroupX_rate]	Defines the bit rate (in kbps) for the coder.
Payload Type payload-type [CodersGroupX_PayloadType]	Defines the payload type if the payload type (i.e., format of the RTP payload) for the coder is dynamic.
Silence Suppression silence-suppression [CodersGroupX_Sce]	Enables silence suppression for the coder. ▪ [0] Disable (Default) ▪ [1] Enable ▪ [2] Enable w/o Adaptation =Applicable only to G.729. ▪

Parameter	Description
Coder Specific coder-specific [CodersGroupX_CoderSpecific]	Defines additional settings specific to the coder. Currently, the parameter is applicable only to the AMR coder and is used to configure the payload format type. [0] = Bandwidth Efficient [1] = Octet Aligned (default) Note: The AMR payload type can be configured globally using the AmrOctetAlignedEnable parameter. However, the Coder Group configuration overrides the global parameter.

19.3 Configuring IP Profiles

The IP Profile Settings table lets you configure up to 125 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. This can include, for example, transcoding or even transrating (of packetization time). 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 348

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 through the Web interface. You can also configure it through ini file (IPProfile) or CLI (configure voip > coders-and-profiles ip-profile).

➤ To configure an IP Profile:

1. Open the IP Profile Settings table (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **IP Profile Settings**).

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

Figure 19-3: IP Profile Settings Table - Add Row Dialog Box

3. Configure an IP Profile according to the parameters described in the table below.
4. Click **Add**.

Table 19-4: IP Profile Settings Table Parameter Descriptions

Parameter	Description
Common	
Index [IpProfile_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name profile-name [IpProfile_ProfileName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters.
Dynamic Jitter Buffer Minimum Delay jitter-buffer-minimum-delay [IpProfile_JitterBufMinDelay]	Defines the minimum delay (in msec) of the device's dynamic Jitter Buffer. The valid range is 0 to 150. The default delay is 10. For more information on Jitter Buffer, see Dynamic Jitter Buffer Operation on page 199. Note: The corresponding global parameter is DJBufMinDelay.

Parameter	Description
Dynamic Jitter Buffer Optimization Factor jitter-buffer-optimization-factor [IpProfile_JitterBufOptFactor]	Defines the Dynamic Jitter Buffer frame error/delay optimization factor. The valid range is 0 to 12. The default factor is 10. For more information on Jitter Buffer, see Dynamic Jitter Buffer Operation on page 199. Notes: - For data (fax and modem) calls, set the parameter to 12. - The corresponding global parameter is DJBufOptFactor.
Jitter Buffer Max Delay jitter-buffer-max-delay [IpProfile_JitterBufMaxDelay]	Defines the maximum delay and length (in msec) of the Jitter Buffer. The valid range is 150 to 2,000. The default is 250.
RTP IP DiffServ 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.
Signaling DiffServ signaling-diffserv [IpProfile_SigIPDiffServ]	Defines the DiffServ value for Premium Control CoS content (Call Control applications). The valid range is 0 to 63. The default is 40. Note: The corresponding global parameter is PremiumServiceClassControlDiffServ.
RTP Redundancy Depth 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 corresponding global parameter is RTPRedundancyDepth.
Echo Canceled echo-canceled [IpProfile_EnableEchoCanceller]	Enables the device's Echo Cancellation feature (i.e., echo from voice calls is removed). [0] Disable [1] Line (default) [2] Acoustic For a detailed description of the Echo Cancellation feature, see Configuring Echo Cancellation on page 186. Note: The corresponding global parameter is EnableEchoCanceller.
Broken Connection Mode disconnect-on-broken-	Defines the device's handling of calls when RTP packets (media) are not received within a user-defined timeout (configured by the

Parameter	Description
connection [IpProfile_DisconnectOnBrokenConnection]	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. 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.
Input Gain input-gain [IpProfile_InputGain]	Defines the pulse-code modulation (PCM) input gain control (in decibels). The valid range is -32 to 31 dB. The default is 0 dB. Note: The corresponding global parameter is InputGain.
Voice Volume voice-volume [IpProfile_VoiceVolume]	Defines the voice gain control (in decibels). The valid range is -32 to 31 dB. The default is 0 dB. Note: The corresponding global parameter is VoiceVolume.
Media IP Version Preference media-ip-version-preference [IpProfile_MediaIPVersionPreference]	Defines the preferred RTP media IP addressing version for outgoing SIP calls (according to RFC 4091 and RFC 4092). The RFCs concern Alternative Network Address Types (ANAT) semantics in the SDP to offer groups of network addresses (IPv4 and IPv6) and the IP address version preference to establish the media stream. The IP address is indicated in the "c=" field (Connection) 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 (preferred) media is IPv4. [3] Prefer IPv6 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first (preferred) media is IPv6. To indicate ANAT support, the device uses the SIP Allow header or to enforce ANAT it uses the Require header: Require: sdp-anat In the outgoing SDP, each 'm=' field is associated with an ANAT

Parameter	Description
	group. This is done using the 'a=mid:' and 'a=group:ANAT' fields. Each 'm=' field appears under a unique 'a=mid:' number, for example: <pre>a=mid:1 m=audio 63288 RTP/AVP 0 8 18 101 c=IN IP6 3000::290:8fff:fe40:3e21</pre> The 'a=group:ANAT' field shows the 'm=' fields belonging to it, using the number of the 'a=mid:' field. In addition, the ANAT group with the preferred 'm=' fields appears first. For example, the preferred group includes 'm=' fields under 'a=mid:1' and 'a=mid3': <pre>a=group:ANAT 1 3 a=group:ANAT 2 4</pre> If you configure the parameter to a "prefer" option, the outgoing SDP offer contains two medias which are the same except for the "c=" field. The first media is the preferred address type (and this type is also on the session level "c=" field), while the second media has its "c=" field with the other address type. Both medias are grouped by ANAT. For example, if the incoming SDP contains two medias, one secured and the other non-secured, the device sends the outgoing SDP with four medias: - Two secured medias grouped in the first ANAT group, one with IPv4 and the other with IPv6. The first is the preferred type. - Two non-secured medias grouped in the second ANAT group, one with IPv4 and the other with IPv6. The first is the preferred type. Note: - The parameter is applicable only when the device offers an SDP. - The IP addressing version is determined according to the first SDP "m=" field. - The feature is applicable to any type of media (e.g., audio and video) that has an IP address. - The corresponding global parameter is MediaIPVersionPreference.
Symmetric MKI 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:bnuYZnMxSfUiGitviWJZmzr70F3AiR00l5Vnh0kH 2^31</pre>

Parameter	Description
	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:R1VyA1xV/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.
MKI Size 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.
Reset SRTP Upon Re-key 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 generate-srtp-keys [IpProfile_GenerateSRTPKeys]	Enables the device to generate a new SRTP key upon receipt of a re-INVITE with the SIP entity associated with the IP Profile. ▪ [0] Only If Required= (Default) The device generates an SRTP key only if necessary. ▪ [1] Always = The device always generates a new SRTP key.
AMD Sensitivity Parameter Suite amd-sensitivity-	Defines the AMD Parameter Suite to use for the Answering Machine Detection (AMD) feature.

Parameter	Description
parameter-suite [IpProfile_AMDSensitivityParameterSuite]	[0] 0 = (Default) Parameter Suite 0 based on North American English with standard detection sensitivity resolution (8 sensitivity levels, from 0 to 7). This AMD Parameter Suite is provided by the AMD Sensitivity file, which is shipped pre-installed on the device. [1] 1 = Parameter Suite based 1 on North American English with high detection sensitivity resolution (16 sensitivity levels, from 0 to 15). This AMD Parameter Suite is provided by the AMD Sensitivity file, which is shipped pre-installed on the device. [2] 2 to [7]7 = Optional Parameter Suites that you can create based on any language (16 sensitivity levels, from 0 to 15). This requires a customized AMD Sensitivity file that needs to be installed on the device. For more information, contact your AudioCodes sales representative. Notes: - To configure the detection sensitivity level, use the 'AMD Sensitivity Level' parameter. - For more information on the AMD feature, see Answering Machine Detection (AMD) on page 206. - The corresponding global parameter is AMDSensitivityParameterSuite.
AMD Sensitivity Level amd-sensitivity-level [IpProfile_AMDSensitivityLevel]	Defines the AMD detection sensitivity level of the selected AMD Parameter Suite (using the 'AMD Sensitivity Parameter Suite' parameter). For Parameter Suite 0, the valid range is 0 to 7, where 0 is for best detection of an answering machine and 7 for best detection of a live call. For any Parameter Suite other than 0, the valid range is 0 to 15, where 0 is for best detection of an answering machine and 15 for best detection of a live call. Note: The corresponding global parameter is AMDSensitivityLevel.
AMD Max Greeting Time amd-max-greeting-time [IpProfile_AMDMaxGreetingTime]	Defines the maximum duration (in 5-msec units) that the device can take to detect a greeting message. The valid range value is 0 to 51132767. The default is 300. Note: The corresponding global parameter is AMDMaxGreetingTime.
AMD Max Post Silence Greeting Time amd-max-post-silence-greeting-time [IpProfile_AMDMaxPostSilenceGreetingTime]	Defines the maximum duration of silence from after the greeting time is over (configured by AMDMaxGreetingTime) until the device's AMD decision. Note: The corresponding global parameter is AMDMaxPostGreetingSilenceTime.
SBC Signaling Tab	
PRACK Mode sbc-prack-mode [IpProfile_SbcPrackMode]	Defines the device's handling of SIP PRACK messages for the SIP entity associated with the IP Profile. [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

Parameter	Description
	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.
P-Asserted-Identity Header Mode sbc-assert-identity [IpProfile_SBCAssertIdentity]	Defines the device's handling of the SIP P-Asserted-Identity header for the SIP entity associated with the IP Profile. 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: - The parameter affects only the initial INVITE request. - The corresponding global parameter is SBCAssertIdentity.
Diversion Header Mode sbc-diversion-mode [IpProfile_SBCDiversionMode]	Defines the device's handling of the SIP Diversion header for the SIP entity associated with the IP Profile. [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 SBCHistoryInfoMode parameter. For more information on interworking of the History-Info and Diversion headers, see Interworking SIP Diversion and History-Info Headers on page 449. 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.
History-Info Header Mode sbc-history-info-mode [IpProfile_SBCHistoryInfoMode]	Defines the device's handling of the SIP History-Info header for the SIP entity associated with the IP Profile. [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 SBCDiversionMode parameter. For more information on interworking of the History-Info and Diversion headers, see Interworking SIP Diversion and History-Info Headers on page 449.

Parameter	Description
Session Expires Mode sbc-session-expires-mode [IpProfile_SBCSessionExpiresMode]	Defines the required session expires mode for the SIP entity associated with the IP Profile. ▪ [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.
Remote Update Support sbc-rmt-update-supp [IpProfile_SBCRemoteUpdateSupport]	Defines whether the SIP UPDATE message is supported by the SIP entity associated with the IP Profile. ▪ [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.
Remote re-INVITE 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.
Remote Delayed Offer Support 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 the parameter to function, you need to assign extension coders to the IP Profile of the SIP entity that does not support delayed offer (using the IpProfile_SBCExtensionCodersGroupID parameter).

Parameter	Description
User Registration Time sbc-usr-reg-time [IpProfile_SBCUserRegistrationTime]	Defines the registration time (in seconds) that the device responds to SIP REGISTER requests from users belonging to the SIP entity associated with the IP Profile. The registration time is inserted in the Expires header in the outgoing response sent to the user. The Expires header determines the lifespan of the registration. For example, a value of 3600 means that the registration will timeout in one hour and at that point, the user will not be able to make or receive calls. The valid range is 0 to 2,000,000. The default is 0. If configured to 0, the Expires header's value received in the user's REGISTER request remains unchanged. If no Expires header is received in the REGISTER message and the parameter is set to 0, the Expires header's value is set to 180 seconds, by default. Note: The corresponding global parameter is SBCUserRegistrationTime.
NAT UDP Registration Time sbc-usr-udp-nat-reg-time [IpProfile_SBCUserBehindUdpNATRegistrationTime]	Defines the registration time (in seconds) that the device includes in register responses, in response to SIP REGISTER requests from users belonging to the SIP entity associated with the IP Profile. The parameter applies only to users that are located behind NAT and whose communication type is UDP. The registration time is inserted in the Expires header in the outgoing response sent to the user. The Expires header determines the lifespan of the registration. For example, a value of 3600 means that the registration will timeout in one hour, unless the user sends a refresh REGISTER before the timeout. Upon timeout, the device removes the user's details from the registration database, and the user will not be able to make or receive calls through the device. The valid value is 0 to 2,000,000. If configured to 0, the Expires header's value received in the user's REGISTER request remains unchanged. By default, no value is defined (-1). Note: If the parameter is not configured, the registration time is according to the global parameter SBCUserRegistrationTime or IP Profile parameter IpProfile_SBCUserRegistrationTime.

Parameter	Description
NAT TCP Registration Time sbc-usr-tcp-nat-reg-time [IpProfile_SBCUserBehindTcpNATRegistrationTime]	Defines the registration time (in seconds) that the device includes in register responses, in response to SIP REGISTER requests from users belonging to the SIP entity associated with the IP Profile. The parameter applies only to users that are located behind NAT and whose communication type is TCP. The registration time is inserted in the Expires header in the outgoing response sent to the user. The Expires header determines the lifespan of the registration. For example, a value of 3600 means that the registration will timeout in one hour, unless the user sends a refresh REGISTER before the timeout. Upon timeout, the device removes the user's details from the registration database, and the user will not be able to make or receive calls through the device. The valid value is 0 to 2,000,000. If configured to 0, the Expires header's value received in the user's REGISTER request remains unchanged. By default, no value is defined (-1). Note: If the parameter is not configured, the registration time is according to the global parameter SBCUserRegistrationTime or IP Profile parameter IpProfile_SBCUserRegistrationTime.
Remote REFER Mode sbc-rmt-refer-behavior [IpProfile_SBCRemoteReferBehavior]	Defines the device's handling of REFER requests for the SIP entity associated with the IP Profile. ▪ [0] Regular = (Default) Refer-To header is unchanged and the device forwards the REFER as is. ▪ [1] Database URL = Changes the Refer-To header so that the re-routed INVITE is sent through the SBC: a. Before forwarding the REFER request, the device changes the host part to the device's IP address and adds a special prefix ("T~&R_") to the Contact user part. b. The incoming INVITE is identified as a REFER-resultant INVITE according to this special prefix. c. The device replaces the host part in the Request-URI with the host from the REFER contact. The special prefix remains in the user part for regular classification, manipulation, and routing. The special prefix can also be used for specific routing rules for REFER-resultant INVITES. d. The special prefix is removed before the resultant INVITE is sent to the destination. ▪ [2] IP Group Name = Sets the host part in the REFER message to the name defined for the IP Group (in the IP Group table). ▪ [3] Handle Locally = Handles the incoming REFER request itself without forwarding the REFER. The device generates a new INVITE to the alternative destination according to the rules in the IP-to-IP Routing table (the 'Call Trigger' field must be set to REFER). Note: The corresponding global parameter is SBCReferBehavior.

Parameter	Description
Remote Replaces Mode 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 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.
Play RBT To Transferee sbc-play-rbt-to-xferee [IpProfile_SBCPlayRBTTToTransferee]	Enables the device to play a ringback tone to the transferred party (transferee) during a blind call transfer, for the SIP entity associated with the IP Profile (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

Parameter	Description
	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 588.
Remote 3xx Mode sbc-rmt-3xx-behavior [IpProfile_SBCRemote3xxBehavior]	Defines the device's handling of SIP 3xx redirect responses for the SIP entity associated with the IP Profile. 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., 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 the 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

Parameter	Description
	✓ 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.
Remote Early Media 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.
Remote Multiple 18x sbc-rmt-multiple-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 the SIP entity associated with the IP Profile. ▪ [0] Not Supported = Only the first 18x response is forwarded to the caller. ▪ [1] Supported = (Default) Multiple 18x responses are forwarded to the caller.
Remote Early Media Response Type sbc-rmt-early-media-resp [IpProfile_SBCRemoteEarlyMediaResponseType]	Defines the SIP provisional response type - 180 or 183 - for forwarding early media to the caller, for the SIP entity associated with the IP Profile. ▪ [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.
Remote Multiple Early Dialogs sbc-multi-early-dialog [IpProfile_SBCRemoteMultipleEarlyDialogs]	Defines the device's handling of To-header tags in call forking responses (i.e., multiple SDP answers) sent to the SIP entity associated with the IP Profile. When the SIP entity initiates an INVITE that is subsequently forked (for example, by a proxy server) to multiple endpoints, the endpoints respond with a SIP 183 containing an SDP answer. Typically, each endpoint's response has a different To-header tag. For example, a call initiated by the SIP entity (100@A) is forked and two endpoints respond with ringing, each with a different tag: ▪ Endpoint "tag 2": SIP/2.0 180 Ringing From: <sip:100@A>;tag=tag1 To: sip:200@B;tag=tag2 Call-ID: c2 ▪ Endpoint "tag 3": SIP/2.0 180 Ringing From: <sip:100@A>;tag=tag1 To: sip:200@B;tag=tag3 Call-ID: c2 In non-standard behavior (when the parameter is configured to Disable), the device forwards all the SDP answers with the same tag. In the example, endpoint "tag 3" is sent with the same tag as endpoint "tag 2" (i.e., To: sip:200@B;tag=tag2). ▪ [-1] According to Operation Mode = (Default) Depends on the setting of the 'Operation Mode' parameter in the IP Group or SRD table:

Parameter	Description
	✓ B2BUA: Device operates as if the parameter is set to Disable [0]. ✓ Call State-full Proxy: Device operates as if the parameter is set to Enable [1]. In addition, the device preserves the From tags and Call-IDs of the endpoints in the SDP answer sent to the SIP entity. ▪ [0] Disable = Device sends the multiple SDP answers with the same To-header tag, to the SIP entity. In other words, this option is relevant if the SIP entity does not support multiple dialogs (and multiple tags). However, non-standard, multiple answer support may still be configured by the SBCRemoteMultipleAnswersMode parameter. ▪ [1] Enable = Device sends the multiple SDP answers with different To-header tags, to the SIP entity. In other words, the SIP entity supports standard multiple SDP answers (with different To-header tags). In this case, the SBCRemoteMultipleAnswersMode parameter is ignored. Note: If the parameter and the SBCRemoteMultipleAnswersMode parameter are disabled, multiple SDP answers are not reflected to the SIP entity (i.e., the device sends the same SDP answer in multiple 18x and 200 responses).
Remote Multiple Answers Mode sbc-multi-answers [IpProfile_SBCRemoteMultipleAnswersMode]	Enables interworking multiple SDP answers within the same SIP dialog (non-standard). The parameter enables the device to forward multiple answers to the SIP entity associated with the IP Profile. The parameter is applicable only when the IpProfile_SBCRemoteMultipleEarlyDialogs parameter is disabled. ▪ [0] Disable = (Default) Device always sends the same SDP answer, which is based on the first received answer that it sent to the SIP entity, for all forked responses (even if 'Forking Handling Mode' is Sequential), and thus, may result in transcoding. ▪ [1] Enable = If the 'Forking Handling Mode' parameter is configured to Sequential, the device sends multiple SDP answers.
Remote Early Media RTP Detection Mode sbc-rmt-early-media-rtp [IpProfile_SBCRemoteEarlyMediaRTP]	Defines whether the destination UA sends RTP immediately after it sends a 18x response. ▪ [0] By Signaling = (Default) Remote client sends RTP immediately after it sends 18x response with early media. The device forwards 18x and RTP as is. ▪ [1] By Media = 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 451.
Remote RFC 3960 Support sbc-rmt-rfc3960-supp [IpProfile_SBCRemoteSupportsRFC3960]	Defines whether the destination UA is capable of receiving 18x messages with delayed RTP. ▪ [0] Not Supported = (Default) UA does not support receipt of 18x messages with delayed RTP. For the device's handling of this remote UA support, see Interworking SIP Early Media on page 451. ▪ [1] Supported = UA is capable of receiving 18x messages with delayed RTP.

Parameter	Description
Remote Can Play Ringback sbc-rmt-can-play-ringback [IpProfile_SBCRemoteCanPlayRingback]	Defines whether the destination UA can play a local ringback tone. [0] No = UA does not support local ringback tone. The device sends 18x with delayed SDP to the UA. [1] Yes = (Default) UA supports local ringback tone. For the device's handling of this remote UA support, see Interworking SIP Early Media on page 451.
Reliable Held Tone Source 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.
Play Held Tone 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 the 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.
Remote Hold Format 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 = (Default) 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.
Remote Representation Mode sbc-rmt-rprsnat [IpProfile_SBCRemoteRepresentationMode]	Enables interworking SIP in-dialog, Contact and Record-Route headers between SIP entities. The parameter defines the device's handling of in-dialog, Contact and Record-Route headers for messages sent to the SIP entity associated with the IP Profile. [-1] According to Operation Mode = (Default) Depends on the setting of the 'Operation Mode' parameter in the IP Group or SRD table: ✓ B2BUA: Device operates as if the parameter is set to

Parameter	Description
	Replace Contact [0]. ✓ Call State-full Proxy: Device operates as if the parameter is set to Add Routing Headers [1]. ▪ [0] Replace Contact = Device replaces the address in the Contact header, received in incoming messages from the other side, with its own address in the outgoing message sent to the SIP entity. ▪ [1] Add Routing Headers = Device adds a Record-Route header for itself to outgoing messages (requests/responses) sent to the SIP entity in dialog-setup transactions. The Contact header remains unchanged. ▪ [2] Transparent = Device doesn't change the Contact header and doesn't add a Record-Route header for itself. Instead, it relies on its' own inherent mechanism to remain in the route of future requests in the dialog (for example, relying on the way the endpoints are set up or on TLS as the transport type).
Keep Incoming Via Headers sbc-keep-via-headers [IpProfile_SBCKeepVIAHeaders]	Enables interworking SIP Via headers between SIP entities. The parameter defines the device's handling of Via headers for messages sent to the SIP entity associated with the IP Profile. ▪ [-1] According to Operation Mode = Depends on the setting of the 'Operation Mode' parameter in the IP Group table or SRD table: ✓ B2BUA: Device operates as if the parameter is set to Disable [0]. ✓ Call State-full Proxy: Device operates as if the parameter is set to Enable [1]. ▪ [0] Disable = Device removes all Via headers received in the incoming SIP request from the other leg and adds a Via header identifying only itself, in the outgoing message sent to the SIP entity. ▪ [1] Enable = Device retains the Via headers received in the incoming SIP request and adds itself as the top-most listed Via header in the outgoing message sent to the SIP entity.
Keep Incoming Routing Headers sbc-keep-routing-headers [IpProfile_SBCKeepRoutingHeaders]	Enables interworking SIP Record-Route headers between SIP entities. The parameter defines the device's handling of Record-Route headers for request/response messages sent to the the SIP entity associated with the IP Profile. ▪ [-1] According to Operation Mode = (Default) Depends on the setting of the 'Operation Mode' in the IP Group or SRD table: ✓ B2BUA: Device operates as if the parameter is set to Disable [0]. ✓ Call State-full Proxy: Device operates as if the parameter is set to Enable [1]. ▪ [0] Disable = Device removes the Record-Route headers received in requests and responses from the other side, in the outgoing SIP message sent to the SIP entity. The device creates a route set for that side of the dialog based on these headers, but doesn't send them to the SIP entity. ▪ [1] Enable = Device retains the incoming Record-Route headers received in requests and non-failure responses from the other side, in the following scenarios: ✓ The message is part of a SIP dialog-setup transaction.

Parameter	Description
	✓ The messages in the setup and previous transaction didn't include the Record-Route header, and therefore hadn't set the route set. Note: Record-Routes are kept only for SIP INVITE, UPDATE, SUBSCRIBE and REFER messages.
Keep User-Agent Header sbc-keep-user-agent [IpProfile_SBCKeepUserAgentHeader]	Enables interworking SIP User-Agent headers between SIP entities. The parameter defines the device's handling of User-Agent headers for response/request messages sent to the SIP entity associated with the IP Profile. ▪ [-1] According to Operation Mode = (Default) Depends on the setting of the 'Operation Mode' parameter in the IP Group or SRD table: ✓ B2BUA: Device operates as if this parameter is set to Disable [0]. ✓ Call State-full Proxy: Device operates as if this parameter is set to Enable [1]. ▪ [0] Disable = Device removes the User-Agent/Server headers received in the incoming message from the other side, and adds its' own User-Agent header in the outgoing message sent to the SIP entity. ▪ [1] Enable = Device retains the User-Agent/Server headers received in the incoming message and sends the headers as is in the outgoing message to the SIP entity.
Handle X-Detect sbc-handle-xdetect [IpProfile_SBCHandleXDetect]	Enables the detection and notification of events (AMD, CPT, and fax), using the X-Detect SIP header. ▪ [0] No (default) ▪ [1] Yes For more information, see Event Detection and Notification using X-Detect Header on page 202.
SBC Media Tab	
Transcoding Mode transcoding-mode [IpProfile_TranscodingMode]	Defines the transcoding mode (media negotiation) for the SIP entity associated with the IP Profile. ▪ [0] Only if Required = (Default) Transcoding is done only when necessary. Many of the media settings (such as gain control) are not implemented on the voice stream. The device forwards RTP packets transparently (RTP-to-RTP), without processing them. ▪ [1] Force = Transcoding is always done on the outgoing leg. The device interworks the media for the SIP entity (as both legs have different media capabilities), by implementing DSP transcoding. This enables the device to receive capabilities that are not negotiated between the SIP entities. For example, it can enforce gain control to use voice transcoding even though both legs have negotiated without the device's intervention (such as extension coders). For more information on extension coders and transcoding, see Coder Transcoding on page 438, Notes: ▪ To implement transcoding, you must configure the number of required DSP channels for transcoding (using the MediaChannels parameter). Each transcoding session uses

Parameter	Description
	two DSP resources. The corresponding global parameter is TranscodingMode.
Extension Coders sbc-ext-coders-group-id [IpProfile_SBCExtensionCodersGroupID]	Assigns a Coder Group used for extension coders, added to the SDP offer in the outgoing leg for the SIP entity associated with the IP Profile. This is used when transcoding is required between two IP entities (i.e., the SDP answer from one doesn't include any coder included in the offer previously sent by the other). For more information on extension coders and transcoding, see Coder Transcoding on page 438, To configure Coders Groups, see Configuring Coder Groups on page 389.
Allowed Coders sbc-allowed-coders-group-id [IpProfile_SBCAllowedCodersGroupID]	Assigns an Allowed Audio Coders Group. This defines audio (voice) coders that can be used for the SIP entity associated with the IP Profile. To configure Allowed Audio Coder Groups, see Configuring Allowed Audio Coder Groups on page 467. For a description of the Allowed Coders feature, see "Restricting Coders" on page 438.
Allowed Coders Mode sbc-allowed-coders-mode [IpProfile_SBCAllowedCodersMode]	Defines the mode of the Allowed Coders feature for the SIP entity associated with the IP Profile. [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). If an Extension Coders Group is also assigned (using the 'Extension Coders' parameter, above), these coders are added to the SDP offer if they also appear in Allowed coders. [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. The coders in the original SDP offer are listed after the Allowed coders. [2] Restriction and Preference = Performs both Restriction and Preference. Notes: - The parameter is applicable only if Allowed coders are assigned to the IP Profile (using the 'Allowed Coders' or 'Allowed Video Coders Group ID' parameters). - For more information on the Allowed Coders feature, see Restricting Coders on page 438.
Allowed Video Coders sbc-allowed-video-coders-group-id [IPProfile_SBCAllowedVideoCodersGroupID]	Assigns an Allowed Video Coders Group. This defines permitted video coders when forwarding video streams to the SIP entity associated with the IP Profile. 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 469.

Parameter	Description
Allowed Media Types sbc-allowed-media-types [IpProfile_SBCAllowedMediaTypes]	Defines media types permitted for the SIP entity associated with the IP Profile. 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., "audio, text" (without quotes). By default, no media types are configured (i.e., all media types are permitted).
SBC Media Security Mode sbc-media-security-behaviour [IpProfile_SBCMediaSecurityBehaviour]	Defines the handling of RTP and SRTP for the SIP entity associated with the IP Profile. ▪ [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.
Media Security Method sbc-media-security-method [IpProfile_SBCMediaSecurityMethod]	Defines the media security protocol for SRTP, for the SIP entity associated with the IP Profile. ▪ [0] SDDES = (Default) The device secures RTP using the Session Description Protocol Security Descriptions (SDDES) protocol to negotiate the cryptographic keys (RFC 4568). The keys are sent in the SDP body ('a=crypto') of the SIP message and are typically secured using SIP over TLS (SIPS). The encryption of the keys is in plain text in the SDP. SDDES implements TLS over TCP. ▪ [1] DTLS = The device uses Datagram Transport Layer Security (DTLS) protocol to secure UDP-based media streams (RFCs 5763 and 5764). For more information on DTLS, see SRTP using DTLS Protocol on page 213. Notes: ▪ To support DTLS, you must also configure the following for the SIP entity: ✓ TLS Context for DTLS (see Configuring TLS Certificate Contexts on page 109). The server cipher ('Cipher Server') must be configured to All.

Parameter	Description
	✓ IpProfile_SBCMediaSecurityBehaviourMedia configured to SRTP or Both. ✓ IpProfile_SBCRTCPMux configured to Supported. The setting is required as the DTLS handshake is done for the port used for RTP. Therefore, RTCP and RTP should be multiplexed over the same port. ▪ The device does not support forwarding of DTLS transparently between endpoints (SIP entities). ▪ As DTLS has been defined by the WebRTC standard as mandatory for encrypting media channels for SRTP key exchange, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 523.
Enforce MKI Size sbc-enforce-mki-size [IpProfile_SBCEnforceMKISize]	Enables negotiation of the Master Key Identifier (MKI) length 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 the SIP entity associated with the IP Profile. ▪ [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.
SDP Remove Crypto LifeTime sbc-sdp-remove-crypto-lifetime [IpProfile_SBCRemoveCryptoLifetimeInSDP]	Defines the handling of the lifetime field in the 'a=crypto' attribute of the SDP for the SIP entity associated with the IP Profile. The SDP field defines the lifetime of the master key as measured in maximum number of SRTP or SRTCP packets using the master key. ▪ [0] No = (Default) The device retains the lifetime field (if present) in the SDP. ▪ [1] Yes = The device removes the lifetime field from the 'a=crypto' attribute. Note: If you configure the parameter to Yes, the following IP Profile parameters must be configured as follows: ▪ IpProfile_EnableSymmetricMKI configured to Enable [1]. ▪ IpProfile_MKISize configured to 0. ▪ IpProfile_SBCEnforceMKISize configured to Enforce [1].
RFC 2833 Mode sbc-rfc2833-behavior [IpProfile_SBCRFC2833Behavior]	Defines the handling of RFC 2833 SDP offer-answer negotiation for the SIP entity associated with the IP Profile. ▪ [0] As is = (Default) The device does not intervene in the RFC 2833 negotiation. ▪ [1] Extend = Each outgoing offer-answer includes RFC 2833 in the offered SDP. The device adds RFC 2833 only if the incoming offer does not include RFC 2833. ▪ [2] Disallow = The device removes RFC 2833 from the incoming offer. Note: If the device interworks between different DTMF methods and one of the methods is in-band DTMF packets (RFC 2833), detection and generation of DTMF methods requires DSP resources.
Alternative DTMF Method	The device's first priority for DTMF method at each leg is RFC

Parameter	Description
sbc-alternative-dtmf-method [IpProfile_SBCAlternativeDTMFMethod]	2833. Thus, if the device successfully negotiates RFC 2833 for the SIP entity associated with the IP Profile, the chosen DTMF method for this leg is RFC 2833. When RFC 2833 negotiation fails, the device uses the parameter to define the DTMF method for the leg. ▪ [0] As Is = (Default) The device does not attempt to interwork any special DTMF method. ▪ [1] In Band ▪ [2] INFO - Cisco ▪ [3] INFO - Nortel ▪ [4] INFO - Lucent = INFO, Korea Note: If the device interworks between different DTMF methods and one of the methods is in-band DTMF packets (RFC 2833), detection and generation of DTMF methods requires DSP resources.
RFC 2833 DTMF Payload Type sbc-2833dtmf-payload [IpProfile_SBC2833DTMFPayloadType]	Defines the payload type of DTMF digits for the SIP entity associated with the IP Profile. 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).
Fax Coders sbc-fax-coders-group-id [IpProfile_SBCFaxCodersGroupID]	Assigns a Coders Group to define the supported fax coders for fax negotiation for the SIP entity associated with the IP Profile. For configuring Coders Groups, see Configuring Coders Groups on page 389. Note: The parameter is applicable only if you set the IpProfile_SBCFaxBehavior parameter to a value other than [0].
Fax Mode sbc-fax-behavior [IpProfile_SBCFaxBehavior]	Enables the device to handle fax offer-answer negotiations for the SIP entity associated with the IP Profile. ▪ [0] As Is = (Default) Device forwards fax transparently, without interference. ▪ [1] Handle always = Handle fax according to fax settings in the IP Profile for all offer-answer transactions (including the initial INVITE). ▪ [2] Handle on re-INVITE = Handle fax according to fax settings in the IP Profile for all re-INVITE offer-answer transactions (except for initial INVITE). Note: The fax settings in the IP Profile include IpProfile_SBCFaxCodersGroupID, IpProfile_SBCFaxOfferMode, and IpProfile_SBCFaxAnswerMode.

Parameter	Description
Fax Offer Mode sbc-fax-offer-mode [IpProfile_SBCFaxOfferMode]	Defines the coders included in the outgoing SDP offer (sent to the called "fax") for the SIP entity associated with the IP Profile. [0] All coders = (Default) Use only (and all) the coders of the selected Coders Group ID configured using the SBCFaxCodersGroupID parameter. [1] Single coder = Use only one coder. If a coder in the incoming offer (from the calling "fax") matches a coder in the SBCFaxCodersGroupID, the device uses this coder. If no match exists, the device uses the first coder listed in the Coders Group ID (SBCFaxCodersGroupID). Note: The parameter is applicable only if you set the IpProfile_SBCFaxBehavior parameter to a value other than [0].
Fax Answer Mode sbc-fax-answer-mode [IpProfile_SBCFaxAnswerMode]	Defines the coders included in the outgoing SDP answer (sent to the calling "fax") for the SIP entity associated with the IP Profile. [0] All coders = Use matched coders between the incoming offer coders (from the calling "fax") and the coders of the selected Coders Group ID (configured using the SBCFaxCodersGroupID parameter). [1] Single coder = (Default) Use only one coder. If the incoming answer (from the called "fax") includes a coder that matches a coder match between the incoming offer coders (from the calling "fax") and the coders of the selected Coders Group ID (SBCFaxCodersGroupID, then the device uses this coder. If no match exists, the device uses the first listed coder of the matched coders between the incoming offer coders (from the calling "fax") and the coders of the selected Coders Group ID. Note: The parameter is applicable only if you set the IpProfile_SBCFaxBehavior parameter to a value other than [0].
Remote Renegotiate on Fax Detection sbc-rmt-renegotiate-on-fax-detect [IPProfile_SBCRemoteRenegotiateOnFaxDetection]	Enables local handling of fax detection and negotiation by the device on behalf of the SIP entity associated with the IP Profile. This applies to faxes sent immediately upon the establishment of a voice channel (i.e., after 200 OK). The device attempts to detect the fax (CNG tone) from the originating SIP entity within a user-defined interval (see the SBCFaxDetectionTimeout parameter) immediately after the voice call is established. Once fax is detected, the device can handle the subsequent fax negotiation by sending re-INVITE messages to both SIP entities. The device also negotiates the fax coders between the two SIP entities. The negotiated coders are according to the list of fax coders assigned to each SIP entity, using the IP Profile parameter 'Fax Coders'. [0] Transparent = (Default) Device does not interfere in the fax transaction and assumes that the SIP entity fully supports fax renegotiation upon fax detection. [1] Only on Answer Side = The SIP entity supports fax renegotiation upon fax detection only if it is the terminating (answering) fax, and does not support renegotiation if it is the originating fax. [2] No = The SIP entity does not support fax re-negotiation

Parameter	Description
	upon fax detection when it is the originating or terminating fax. Notes: - This feature is applicable only when both SIP entities do not fully support fax detection (receive or send) and negotiation: one SIP entity must be assigned an IP Profile where the parameter is set to [1] or [2], while the peer SIP entity must be assigned an IP Profile where the parameter is set to [2]. - This feature is supported only if at least one of the SIP entities use the G.711 coder. - This feature utilizes DSP resources. If there are insufficient resources, the fax transaction fails.
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>
SDP Ptime Answer sbc-sdp-ptime-ans [IpProfile_SBCSDPPtimeAnswer]	Defines the packetization time (ptime) of the coder in RTP packets for the SIP entity associated with the IP Profile. This is useful when implementing transrating. [0] Remote Answer = (Default) Use ptime according to SDP answer.

Parameter	Description
	[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.
Preferred Ptime 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 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).
Use Silence Suppression sbc-use-silence-supp [IpProfile_SBCUseSilenceSupp]	Defines silence suppression support for the SIP entity associated with the IP Profile [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.
RTP Redundancy Mode sbc-rtsp-red-behav [IpProfile_SBCRTPRedundancyBehavior]	Enables interworking RTP redundancy negotiation support between SIP entities in the SDP offer-answer exchange (according to RFC 2198). The parameter defines the device's handling of RTP redundancy for the SIP entity associated with the IP Profile. According to the RTP redundancy SDP offer/answer negotiation, the device uses or discards the RTP redundancy packets. The parameter enables asymmetric RTP redundancy, whereby the device can transmit and receive RTP redundancy packets to and from a specific SIP entity, while transmitting and receiving regular RTP packets (no redundancy) for the other SIP entity involved in the voice path. The device can identify the RTP redundancy payload type in the SDP for indicating that the RTP packet stream includes redundant packets. RTP redundancy is indicated in SDP using the "red" coder type, for example: <pre>a=rtptime:<payload type> red/8000/1</pre> RTP redundancy is useful when there is packet loss; the missing information may be reconstructed at the receiver side from the redundant packets. [0] As Is = (Default) The device does not interfere in the RTP redundancy negotiation and forwards the SDP offer/answer (incoming and outgoing calls) as is without interfering in the

Parameter	Description
	RTP redundancy negotiation. [1] Enable = The device always adds RTP redundancy capabilities in the outgoing SDP offer sent to the SIP entity. Whether RTP redundancy is implemented depends on the subsequent incoming SDP answer from the SIP entity. The device does not modify the incoming SDP offer received from the SIP entity, but if RTP redundancy is required, it will be supported. Select the option if the SIP entity requires RTP redundancy. [2] Disable = The device removes the RTP redundancy payload (if present) from the SDP offer/answer for calls received from or sent to the SIP entity. Select the option if the SIP entity does not support RTP redundancy. Notes: - To enable the device to generate RFC 2198 redundant packets, use the IPProfile_RTPRedundancyDepth parameter. - To configure the payload type in the SDP offer for RTP redundancy, use the RFC2198PayloadType.
RTCP Mode sbc-rtcp-mode [IPProfile_SBCRTCPMode]	Defines how the device handles RTCP packets during call sessions for the SIP entity associated with the IP Profile. 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 (unless transcoding is done, in which case, the device generates RTCP on both legs). [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.
Jitter Compensation sbc-jitter-compensation [IPProfile_SBCJitterCompensation]	Enables the on-demand jitter buffer for SBC calls. The jitter buffer can be used when other functionality such as voice transcoding are not done on the call. The 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 Notes: - 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 functionality may require DSP resources. For more information, contact your AudioCodes sales representative.

Parameter	Description
ICE Mode ice-mode [IPProfile_SBCIceMode]	Enables Interactive Connectivity Establishment (ICE) Lite for the SIP entity associated with the IP Profile. ICE is a methodology for NAT traversal, employing the Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) protocols to provide a peer with a public IP address and port that can be used to connect to a remote peer. ▪ [0] Disable (default) ▪ [1] Lite For more information on ICE Lite, see ICE Lite. Note: As ICE has been defined by the WebRTC standard as mandatory, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 523.
SDP Handle RTCP sbc-sdp-handle-rtcp [IPProfile_SBCSDPHandleRTCP Attribute]	Enables the interworking of the RTCP attribute, 'a=rtcp' (RTCP) in the SDP, for the SIP entity associated with the IP Profile. The RTCP attribute is used to indicate the RTCP port for media when that port is not the next higher port number following the RTP port specified in the media line ('m='). The parameter is useful for SIP entities that either require the attribute or do not support the attribute. For example, Google Chrome and Web RTC do not accept calls without the RTCP attribute in the SDP. In Web RTC, Chrome (SDS) generates the SDP with 'a=rtcp', for example: <pre>m=audio 49170 RTP/AVP 0 a=rtcp:53020 IN IP6 2001:2345:6789:ABCD:EF01:2345:6789:ABCD</pre> ▪ [0] Don't Care = (Default) The device forwards the SDP as is without interfering in the RTCP attribute (regardless if present or not). ▪ [1] Add = The device adds the 'a=rtcp' attribute to the outgoing SDP offer sent to the SIP entity if the attribute was not present in the original incoming SDP offer. ▪ [2] Remove = The device removes the 'a=rtcp' attribute, if present in the incoming SDP offer received from the other SIP entity, before sending the outgoing SDP offer to the SIP entity. Note: As the RTCP attribute has been defined by the WebRTC standard as mandatory, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 523.
RTCP Mux sbc-rtcp-mux [IPProfile_SBCRTCPMux]	Enables interworking of multiplexing of RTP and RTCP onto a single local port, between SIP entities. The parameter enables multiplexing of RTP and RTCP traffic onto a single local port, for the SIP entity associated with the IP Profile. Multiplexing of RTP data packets and RTCP packets onto a single local UDP port is done for each RTP session (according to RFC 5761). If multiplexing is not enabled, the device uses different (but adjacent) ports for RTP and RTCP packets. With the increased use of NAT and firewalls, maintaining multiple NAT bindings can be costly and also complicate firewall administration since multiple ports must be opened to allow RTP traffic. To reduce these costs and session setup times, support

Parameter	Description
	for multiplexing RTP data packets and RTCP packets onto a single port is advantageous. For multiplexing, the initial SDP offer must include the "a=rtcp-mux" attribute to request multiplexing of RTP and RTCP onto a single port. If the SDP answer wishes to multiplex RTP and RTCP, it must also include the "a=rtcp-mux" attribute. If the answer does not include the attribute, the offerer must not multiplex RTP and RTCP packets. If both ICE and multiplexed RTP-RTCP are used, the initial SDP offer must also include the "a=candidate:" attribute for both RTP and RTCP along with the "a=rtcp:" attribute, indicating a fallback port for RTCP in case the answerer does not support RTP and RTCP multiplexing. ▪ [0] Not Supported = (Default) RTP and RTCP packets use different ports. ▪ [1] Supported = Device multiplexes RTP and RTCP packets onto a single port. Note: As RTP multiplexing has been defined by the WebRTC standard as mandatory, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 523.
RTCP Feedback sbcrtcp-feedback [IPProfile_SBCRTCPFeedback]	Enables RTCP-based feedback indication in outgoing SDPs sent to the SIP entity associated with the IP Profile. The parameter supports indication of RTCP-based feedback, according to RFC 5124, during RTP profile negotiation between two communicating SIP entities. RFC 5124 defines an RTP profile (S)AVPF for (secure) real-time communications to provide timely feedback from the receivers to a sender. For more information on RFC 5124, see http://tools.ietf.org/html/rfc5124. Some SIP entities may require RTP secure-profile feedback negotiation (AVPF/SAVPF) in the SDP offer/answer exchange, while other SIP entities may not support it. The device indicates whether or not feedback is supported on behalf of the SIP entity. It does this by adding an "F" or removing the "F" from the SDP media line ('m=') for AVP and SAVP. For example, the following shows "AVP" appended with an "F", indicating that the SIP entity is capable of receiving feedback <pre>m=audio 49170 RTP/SAVPF 0 96</pre> ▪ [0] Disable = (Default) The device does not send the feedback flag ("F") in SDP offers/answers that are sent to the SIP entity. If the SDP 'm=' attribute of an incoming message that is destined to the SIP entity includes the feedback flag, the device removes it before sending the message to the SIP entity. ▪ [1] Enable = The device includes the feedback flag ("F") in the SDP offer that is sent to the SIP entity. The device includes the feedback flag in the SDP answer sent to the SIP entity only if it was present in the SDP offer received from the other SIP entity. Note: As RTCP-based feedback has been defined by the WebRTC standard as mandatory, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 523.

Parameter	Description
Direct Media Tag sbc-dm-tag [IPProfile_SBCDirectMediaTag]	Defines an identification tag for enabling direct media (no Media Anchoring) for the SIP entity associated with the IP Profile. Direct media occurs between all endpoints whose IP Profiles have the same tag value (non-empty value). For example, if you set the parameter to "direct-rtp" for two IP Profiles "IP-PBX-1" and "IP-PBX-2", the device employs direct media for calls amongst endpoints associated with IP Profile "IP-PBX-1", for calls amongst endpoints associated with IP Profile "IP-PBX-2", and for calls between endpoints associated with IP Profile "IP-PBX-1" and IP Profile "IP-PBX-2". The valid value is a string of up to 16 characters. By default, no value is defined. For more information on direct media, see Direct Media on page 436. Note: If you enable direct media for the IP Profile, make sure that your Media Realm provides sufficient ports, as media may traverse the device for mid-call services (e.g., call transfer).

Part V

Session Border Controller Application

20 SBC Overview

This section provides an overview of the device's SBC application.



Notes:

- For guidelines on how to deploy your SBC device, refer to the *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 596.
- 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 837.

20.1 Feature List

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 and with far-end users located behind NAT on the WAN. The device supports this by:
 - Continually registering far-end users with its users registration database.
 - Maintaining remote NAT binding state by frequent registrations and 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 SDP offer-answer 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.
- Stateful Proxy Operation Mode: The device can act as a Stateful Proxy by enabling SIP messages to traverse it transparently (with minimal interference) between the inbound and outbound legs.
- B2BUA and 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.
- User-defined manipulation of SIP To, From, and Request-URI host names.
- Generates a new SIP Call-ID header value (different between legs).
- Changes the SIP Contact header and sets it to the device's 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.

20.2 B2BUA and Stateful Proxy Operating Modes

The device can operate in one or both of the following SBC modes:

- **Back-to-Back User Agent (B2BUA):** Maintains independent sessions toward the endpoints, processing an incoming request as a user agent server (UAS) on the inbound leg, and processing the outgoing request as a user agent client (UAC) on the outbound leg. SIP messages are modified regarding headers between the legs and all the device's interworking features may be applied.
- **Stateful Proxy Server:** SIP messages traverse the device transparently (with minimal interference) between the inbound and outbound legs, for connecting SIP endpoints.

By default, the device's B2BUA mode changes SIP dialog identifiers and topology data in SIP messages traversing through it:

- Call identifiers: Replaces the From-header tag and Call-ID header so that they are different for each leg (inbound and outbound).
- Routing headers:
 - Removes all Via headers in incoming requests and sends the outgoing message with its own Via header.
 - Doesn't forward any Record-Route headers from the inbound to outbound leg, and vice versa.
 - Replaces the address of the Contact header in the incoming message with its own address in the outgoing message.
- Replaces the User-Agent/ Server header value in the outgoing message, and replaces the original value with itself in the incoming message.

In contrast, when the device operates in Stateful Proxy mode, the device by default forwards SIP messages transparently (unchanged) between SIP endpoints (from inbound to outbound legs). The device retains the SIP dialog identifiers and topology headers received in the incoming message and sends them as is in the outgoing message. The device handles the above mentioned headers transparently (i.e., they remain unchanged) or according to configuration (enabling partial transparency), and only adds itself as the top-most Via header and optionally, to the Record-Route list. For configuring the handling of these headers for partial transparency, use the following IP Profile parameters (see "Configuring IP Profiles" on page 391):

- IpProfile_SBCRemoteRepresentationMode: Contact and Record-Route headers
- IpProfile_SBCKeepVIAHeaders: Via headers
- IpProfile_SBCKeepUserAgentHeader: User-Agent headers
- IpProfile_SBCKeepRoutingHeaders: Record-Route headers
- IpProfile_SBCRemoteMultipleEarlyDialogs: To-header tags

Thus, the Stateful Proxy mode provides full SIP transparency (no topology hiding) or asymmetric topology hiding. Below is an example of a SIP dialog-initiating request when operating in Stateful Proxy mode for full transparency, showing all the incoming SIP headers retained in the outgoing INVITE message.

Figure 20-1: Example of SIP Message Handling in Stateful Proxy Mode

Incoming INVITE	Outgoing INVITE
<pre> INVITE sip:bob@domain.com SIP/2.0 To: Bob <sip:bob@domain.com> From: Alice <sip:alice@caller.com>;tag=100 Call-ID: callid1@caller.com Contact: <sip:alice@pc1.caller.com> Via: SIP/2.0/UDP pc2.com;branch=branch2 Via: SIP/2.0/UDP pc1.com;branch=branch1 Record-Route: <pc2.com;lr> Record-Route: <pc1.com;lr> CSeq: 666 INVITE User-Agent: IPPv3.1 Max-Forwards: 70 Content-Type: application/sdp Content-Length: 142 v=0 ... </pre>	<pre> INVITE sip:bob@domain.com SIP/2.0 To: Bob <sip:bob@domain.com> From: Alice <sip:alice@caller.com>;tag=100 Call-ID: callid1@caller.com Contact: <sip:alice@pc1.caller.com> Via: SIP/2.0/UDP Proxy-IP;branch=branch3 Via: SIP/2.0/UDP pc2.com;branch=branch2 Via: SIP/2.0/UDP pc1.com;branch=branch1 Record-Route: <Proxy-IP;lr> Record-Route: <pc2.com;lr> Record-Route: <pc1.com;lr> CSeq: 666 INVITE User-Agent: IPPv3.1 Max-Forwards: 70 Content-Type: application/sdp Content-Length: 142 v=0 ... </pre>

Some of the reasons for implementing Stateful Proxy mode include:

- B2BUA typically hides certain SIP headers for topology hiding. In specific setups, some SIP servers require the inclusion of these headers to know the history of the SIP request. In such setups, the requirement may be asymmetric topology hiding, whereby SIP traffic toward the SIP server must expose these headers whereas SIP traffic toward the users must not expose these headers.
- B2BUA changes the call identifiers between the inbound and outbound SBC legs and therefore, call parties may indicate call identifiers that are not relayed to the other leg. Some SIP functionalities are achieved by conveying the SIP call identifiers either in SIP specific headers (e.g., Replaces) or in the message bodies (e.g. Dialog Info in an XML body).
- In some setups, the SIP client authenticates using a hash that is performed on one or more of the headers that B2BUA changes (removes). Therefore, implementing B2BUA would cause authentication to fail.
- For facilitating debugging procedures, some administrators require that the value in the Call-ID header remains unchanged between the inbound and outbound SBC legs. As B2BUA changes the Call-ID header, such debugging requirements would fail.

The operating mode can be configured per the following configuration entities:

- SRDs in the SRD table (see "Configuring SRDs" on page 332)
- IP Groups in the IP Group table (see "Configuring IP Groups" on page 348)

If the operation mode is configured in both tables, the operation mode of the IP Group is applied. Once configured, the device uses default settings in the IP Profile table for handling the SIP headers, as mentioned previously. However, you can change the default settings to enable partial transparency.

**Notes:**

- The To-header tag remains the same for inbound and outbound legs of the dialog, regardless of operation mode.
- If the Operation Mode of the SRD\IP Group of one leg of the dialog is set to 'Call Stateful Proxy', the device also operates in this mode on the other leg with regards to the dialog identifiers (Call-ID header, tags, CSeq header).
- It is recommended to implement the B2BUA mode, unless one of the reasons mentioned previously is required. B2BUA supports all the device's feature-rich offerings, while Stateful Proxy may offer only limited support. The following features are not supported when in Stateful Proxy mode:
 - ✓ Alternative routing
 - ✓ Call forking
 - ✓ Terminating REFER/3xx
- If Stateful Proxy mode is enabled and any one of the unsupported features is enabled, the device disables the Stateful Proxy mode and operates in B2BUA mode.
- You can configure the device to operate in both B2BUA and Stateful Proxy modes for the same users. This is typically implemented when users need to communicate with different SIP entities (IP Groups). For example, B2BUA mode for calls destined to a SIP Trunk and Stateful Proxy mode for calls destined to an IP PBX. The configuration is done using IP Groups and SRDs.
- If Stateful Proxy mode is used only due to the debugging benefits, it is recommended to configure the device to only forward the Call-ID header unchanged

20.3 Call Processing of SIP Dialog Requests

The device processes incoming SIP dialog requests (SIP methods) such as INVITE, SUBSCRIBE, OPTIONS, REFER, INFO, UNSOLICITED NOTIFY, MESSAGE, and REGISTER. The process is summarized in the following figure and subsequently described:

Figure 20-2: SBC Call Processing



The SIP dialog-initiating process consists of the following stages:

1. **Determining Source and Destination URL:** The SIP protocol has more than one URL in a dialog-initiating request that may represent the source and destination URLs. The device obtains the source and destination URLs from certain SIP headers. Once the URLs are determined, the user and host parts of the URLs can be used as matching rule characteristics for classification, message manipulation, and call routing.

- **All SIP requests (e.g., INVITE) except REGISTER:**
 - ♦ Source URL: 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: Obtained from the Request-URI.
- **REGISTER dialogs:**
 - ♦ Source URL: Obtained from the To header.
 - ♦ Destination URL: Obtained from the Request-URI.



Note: You can specify the SIP header from where you want the device to obtain the source URL in the incoming dialog request. This is configured in the IP Group table using the 'Source URI Input' parameter (see "Configuring IP Groups" on page 348).

2. **Determining SIP Interface:** The device checks the SIP Interface on which the SIP dialog is received. The SIP Interface defines the local SIP "listening" port and IP network interface. For more information, see "Configuring SIP Interfaces" on page 342.
3. **Applying SIP Message Manipulation:** Depending on configuration, the device can apply a SIP message manipulation rule (assigned to the SIP Interface) on the incoming SIP message. A SIP Message Manipulation rule defines a matching characteristics (*condition*) of the incoming SIP message and the corresponding manipulation operation (e.g., remove the P-Asserted-Identity header), which can apply to almost any aspect of the message (add, remove or modify SIP headers and parameters). For more information, see "Configuring SIP Message Manipulation" on page 377.
4. **Classifying to an IP Group:** Classification identifies the incoming SIP dialog request as belonging to a specific IP Group (i.e., from where the SIP dialog request originated). The classification process is based on the SRD to which the dialog belongs (the SRD is determined according to the SIP Interface). For more information, see "Configuring Classification Rules" on page 471.
5. **Applying IP-to-IP Inbound Manipulation:** Depending on configuration, the device can apply an IP-to-IP Inbound Manipulation rule to the incoming dialog. This manipulates the user part of the SIP URI for source (e.g., in the SIP From header) and destination (e.g., in the Request-URI line). The manipulation rule is associated with the incoming dialog, by configuring the rule with incoming matching characteristics such as source IP Group and destination host name. The manipulation rules are also assigned an SBC Routing Policy, which in turn, is assigned to IP-to-IP routing rules. As most deployments require only one SBC Routing Policy, the default Routing Policy is automatically assigned to manipulation and routing rules. For more information, see "Configuring IP-to-IP Inbound Manipulations" on page 499.
6. **SBC IP-to-IP Routing:** The device searches the IP-to-IP Routing table for a routing rule that matches the characteristics of the incoming call. If found, the device routes the call to the configured destination which can be, for example, an IP Group, the Request-URI if the user is registered with the device, and a specified IP address. For more information, see "Configuring SBC IP-to-IP Routing Rules" on page 480.
7. **Applying Inbound SIP Message Manipulation:** Depending on configuration, the device can apply a SIP message manipulation rule (assigned to the IP Group) on the incoming dialog. For more information, see Stage 3.
8. **Applying IP-to-IP Outbound Manipulation:** Depending on configuration, the device can apply an IP-to-IP Outbound Manipulation rule to the outbound dialog. This manipulates the user part of the Request-URI for source (e.g., in the SIP From

header) or destination (e.g., in the SIP To header) or calling name in the outbound SIP dialog. The manipulation rule is associated with the dialog, by configuring the rule with incoming matching characteristics such as source IP Group and destination host name. The manipulation rules are also assigned an SBC Routing Policy, which in turn, is assigned to IP-to-IP routing rules. As most deployments require only one SBC Routing Policy, the default Routing Policy is automatically assigned to manipulation rules and routing rules. For more information, see "Configuring IP-to-IP Outbound Manipulations" on page 504.

9. **Applying Outbound SIP Message Manipulation:** Depending on configuration, the device can apply a SIP message manipulation rule (assigned to the IP Group) on the outbound dialog. For more information, see Stage 3.
10. The call is sent to the configured destination.

20.4 User Registration

The device provides a registration database for registering users. Only users belonging to a User-type IP Group can register with the device. User-type IP Groups represent a group of SIP user agents that share the following characteristics:

- Perform registrations and share the same serving proxy/registrar
- Same SIP and media behavior
- Same IP Profile
- Same SIP handling configuration
- Same Call Admission Control (CAC)

Typically, the device is configured as the user's outbound proxy, routing requests (using the IP-to-IP Routing table) from the user's User-type IP Group to the serving proxy, and vice versa. Survivability can be achieved using the alternative routing feature.

The device forwards registration requests (REGISTER messages) from a Server-type IP Group, but does not save the registration binding in its' registration database.

20.4.1 Initial Registration Request Processing

A summary of the device's handling of registration requests (REGISTER messages) is as follows:

- The URL in the To header of the REGISTER message constitutes the primary Address of Record (AOR) for registration (according to standard). The device can save other AORs in its registration database as well. When the device searches for a user in its' registration database, any of the user's AORs can result in a match.
- The device's Classification process for initial REGISTER messages is slightly different than for other SIP messages. Unlike other requests, initial REGISTER requests can't be classified according to the registration database.
- If registration succeeds (replied with 200 OK by the destination server), the device adds a record to its' registration database, which identifies the specific contact of the specific user (AOR). The device uses this record to route subsequent SIP requests to the specific user (in normal or survivability modes).
- The records in the device's registration database include the Contact header. The device adds every REGISTER request to the registration database before manipulation, allowing correct user identification in the Classification process for the next received request.

- You can configure Call Admission Control (CAC) rules for incoming and outgoing REGISTER messages. For example, you can limit REGISTER requests from a specific IP Group or SRD. Note that this applies only to concurrent REGISTER dialogs and not concurrent registrations in the device's registration database.

The device provides a dynamic registration database that it updates according to registration requests traversing it. Each database entry for a user represents a binding between an AOR (obtained from the SIP To header), optional additional AORs, and one or more contacts (obtained from the SIP Contact headers). Database bindings are added upon successful registration responses from the proxy server (SIP 200 OK). The device removes database bindings 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 UA did not send a refresh registration request).

**Notes:**

- 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 message is not supported.
- One database is shared between all User-type IP Groups.

20.4.2 Classification and Routing of Registered Users

The device can classify incoming SIP dialog requests (e.g., INVITE) from registered users to an IP Group, by searching for the sender's details in the registration database. The device uses the AOR from the From header and the URL in the Contact header of the request to locate a matching registration binding. The found registration binding contains information regarding the registered user, including the IP Group to which it belongs. (Upon initial registration, the Classification table is used to classify the user to a User-type IP Group and this information is then added with the user in the registration database.)

The destination of a dialog request can be a registered user and the device thus uses its registration database to route the call. This can be achieved by various ways such as configuring a rule in the IP-to-IP Routing table where the destination is a User-type IP Group or any matching user registered in the database ('Destination Type' is configured to **All Users**). The device searches the registration database for a user that matches the incoming Request-URI (listed in chronological order):

1. Unique Contact generated by the device and sent in the initial registration request to the serving proxy.
2. AOR. The AOR is originally obtained from the incoming REGISTER request and must either match both user part and host part of the Request-URI, or only user part.
3. Contact. The Contact is originally obtained from 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.

You can configure (using the `SBCDBRoutingSearchMode` parameter) for which part of the destination Request-URI in the INVITE message the device must search in the registration database:

- Only by entire Request-URI (user@host), for example, "4709@joe.company.com".
- By entire Request-URI, but if not found, by the user part of the Request-URI, for example, "4709".

When an incoming INVITE is received for routing to a user and the user is located in the registration database, the device sends the call to the user's corresponding contact address specified in the database.



Note: If the Request-URI contains the "tel:" URI or "user=phone" parameter, the device searches only for the user part.

20.4.3 General Registration Request Processing

The device's general handling of registration requests (REGISTER messages) for unregistered users is as follows:

- The device routes REGISTER requests according to 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 (replies with a SIP 4xx) the request according to the user's IP Group configuration.
- Alternative routing can be configured for REGISTER requests, in the IP-to-IP Routing table.
- By default, the Expires header has the same value in incoming and outgoing REGISTER messages. However, you can modify the Expires value using the following parameters: `SBCUserRegistrationTime`, `SBCProxyRegistrationTime`, `SBCRandomizeExpires`, and `SBCSurvivabilityRegistrationTime`. You can also modify the Expires value of REGISTER requests received from users located behind NAT, using the IP Profile parameters `IpProfile_SBCUserBehindUdpNATRegistrationTime` and `IpProfile_SBCUserBehindTcpNATRegistrationTime`.
- By default, the Contact header in outgoing REGISTER message is different than the Contact header in the incoming REGISTER. The user part of the Contact is populated with a unique contact generated by the device and associated with the specific registration. The IP address in the host part is changed to the address of the device. Alternatively, the original user can be retained in the Contact header and used in the outgoing REGISTER request (using the `SBCKeepContactUserinRegister` parameter).

20.4.4 Registration Refreshes

Registration refreshes are incoming REGISTER requests from users that are registered in the device's registration database. The device sends these refreshes to the serving proxy only if the serving proxy's Expires time is about to expire; otherwise, the device responds with a 200 OK to the user, 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 its registration database and whose registration timeout has expired. When a user's registration timer expires, the device removes the user's 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).

You can also apply a graceful period to unregistered requests, using the 'User Registration Grace Time' parameter (`SBCUserRegistrationGraceTime`):

- You can configure the device 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.
- The graceful period is also used before removing a user from the registration database when the device receives a successful unregister response (200 OK) from the registrar/proxy server. This is useful in scenarios, for example, in which users (SIP user agents) such as IP Phones erroneously send unregister requests. Instead of immediately removing the user from the registration database upon receipt of a successful unregister response, the device waits until it receives a successful unregister response from the registrar server, waits the user-defined graceful time and if no register refresh request is received from the user agent, removes the contact (or AOR) from the database.

The device keeps registered users in its' registration database even if connectivity with the proxy 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).

20.4.5 Registration Restriction Control

The device provides flexibility in controlling user registrations:

- **Limiting Number of Registrations:** You can limit the number of users that can register with the device per IP Group, SIP Interface, and/or SRD, in the IP Group, SIP Interface and SRD tables respectively. By default, no limitation exists.
- **Blocking Incoming Calls from Unregistered Users:** You can block incoming calls (INVITE requests) from unregistered users belonging to User-type IP Groups. By default, calls from unregistered users are not blocked. This is configured per SIP Interface or SRD. When the call is rejected, the device sends a SIP 500 (Server Internal Error) response to the remote end.

20.4.6 Deleting Registered Users

You can remove registered users from the device's registration database through CLI:

- To delete a specific registered user:

```
# clear voip register db sbc user <AOR of user - user part or user@host>
```

For example:

```
# clear voip register db sbc user John@10.33.2.22  
# clear voip register db sbc user John
```
- To delete all registered users belonging to a specific IP Group:

```
# clear voip register db sbc ip-group <ID or name>
```

20.5 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 multiple media sessions of different types (e.g. audio and fax). In a SIP dialog, multiple offer-answer transactions may occur and each may change the media session 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 (e.g., audio, secure audio, video, fax, and 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

Typically, the device does not change the negotiated media capabilities (mainly performed by the remote user agents). However, it does examine and may take an active role in the SDP offer-answer mechanism. This is done mainly to anchor the media to the device (default) and also to change the negotiated media type, if configured. Some of the media handling features, which are described later in this section, include the following:

- Media anchoring (default)
- Direct media
- Audio coders restrictions
- Audio coders transcoding
- RTP-SRTP transcoding
- DTMF translations
- Fax translations and detection
- Early media and ringback tone handling
- Call hold translations and held tone generation
- NAT traversal
- RTP broken connections
- Media firewall
 - 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.

20.5.1 Media Anchoring

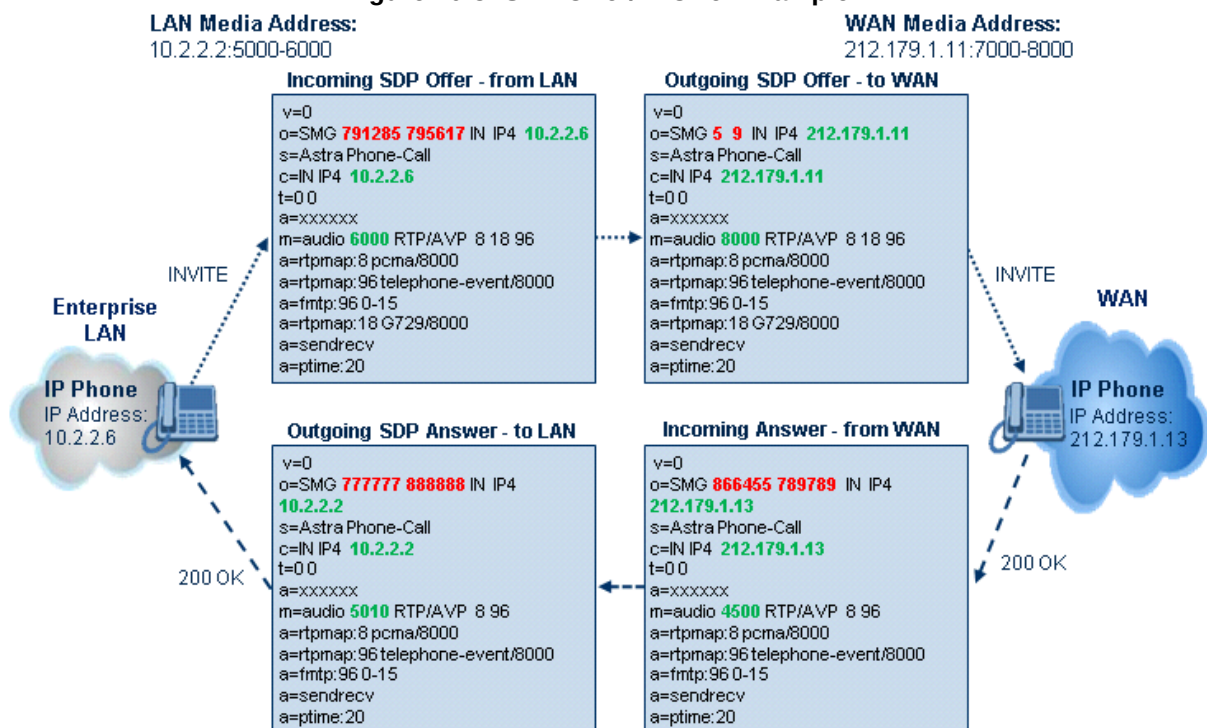
By default, the device anchors the media (RTP) traffic. In other words, the media between SIP endpoints traverses the device. You can change this default mode by enabling direct media between SIP endpoints. Media anchoring may be required, for example, to resolve NAT problems, enforce media security policies, perform media transcoding, and media monitoring.

To enforce RTP traffic to flow through the device, the device modifies all IP address fields in the SDP:

- 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

The device uses different local ports (e.g., for RTP, RTCP and fax) for each leg (inbound and outbound). The local ports are allocated from the Media Realm associated with each leg. The Media Realm assigned to the leg's IP Group (in the IP Group table) is used. If not assigned to the IP Group, the Media Realm assigned to the leg's SIP Interface (in the SIP Interface table) is used. The following figure provides 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 20-3: SDP Offer/Answer Example



20.5.2 Direct Media

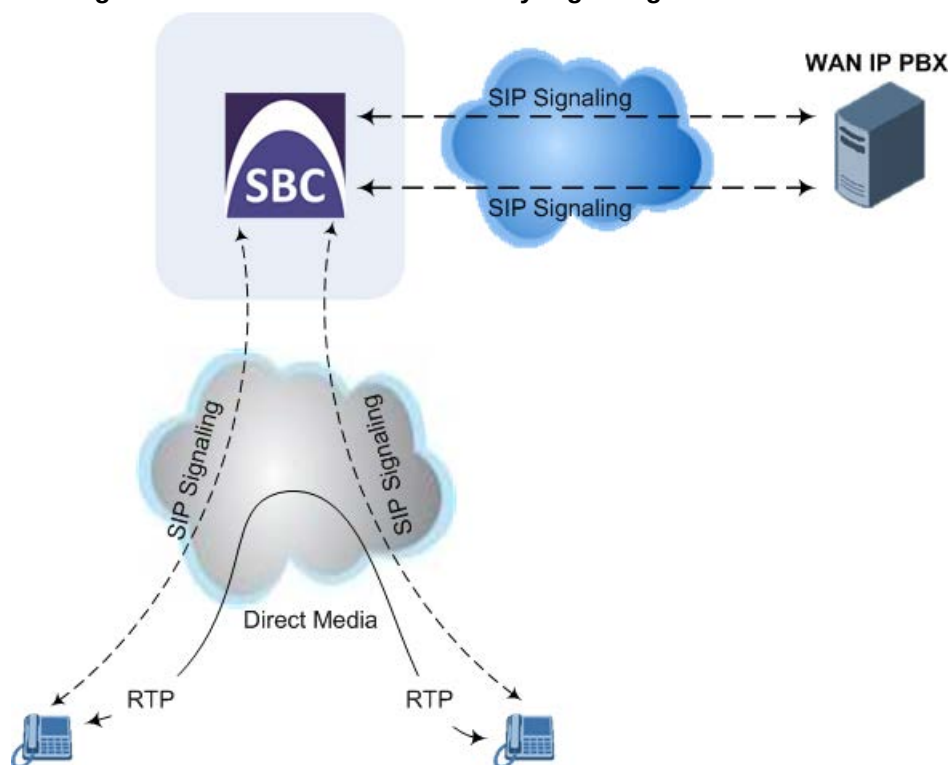
You can configure the device to allow the media (RTP/SRTP) session to flow directly between the SIP endpoints, without traversing the device. This is referred to as No Media Anchoring (also known as Anti-Tromboning or Direct Media). SIP signaling continues to traverse the device, with minimal intermediation and involvement, to enable certain SBC capabilities such as routing. By default, the device employs media anchoring, whereby the media session traverses the device, as described in "Media Anchoring" on page 435.

Direct media offers the following benefits:

- Saves network bandwidth
- Reduces the device's CPU usage (as there is no media handling)
- Avoids interference in SDP negotiation and header manipulation on RTP/SRTP

Direct media is typically implemented for calls between users located in the same LAN or domain, and where NAT traversal is not required and other media handling features such as media transcoding is not required. The following figure provides an example of direct media between LAN IP phones, while SIP signaling continues to traverse the device between LAN IP phones and the hosted WAN IP-PBX.

Figure 20-4: Direct Media where only Signaling Traverses Device



➤ To enable direct media:

- **For all calls:** Use the global parameter, `SBCDirectMedia` (overrides all other direct media configuration).
- **For specific calls:**
 - **SIP Interface:** You can enable direct media per SIP Interface (in the SIP Interface table), whereby calls (source and destination) associated with **this same** SIP Interface are handled as direct media calls. The SIP Interface can also enable direct media for users located behind the same NAT. For more information, see "Configuring SIP Interfaces" on page 342.

- **Direct Media Tag:** You can enable direct media between users that are configured with the same Direct Media tag value. The tag is configured using the IP Profile table's IPProfile_SBCDirectMediaTag parameter (see "Configuring IP Profiles" on page 391).

The device employs direct media between endpoints under the following configuration conditions (listed in chronological order):

1. Direct media is enabled by the global parameter (SBCDirectMedia).
2. IP Groups of the endpoints are associated with IP Profiles whose 'Direct Media Tag' parameter has the same value (non-empty value).
3. IP Groups of the endpoints have the 'SBC Operation Mode' parameter set to Microsoft Server (direct media is required in the Lync environment). For more information, see "Configuring IP Groups" on page 348.
4. IP Groups of the endpoints use the same SIP Interface and the SIP Interface's 'SBC Direct Media' parameter is set to **Enable** (SIPInterface_SBCDirectMedia = 1).
5. IP Groups of the endpoints use the same SIP Interface and the SIP Interface's 'SBC Direct Media' parameter is set to **Enable When Single NAT** (SIPInterface_SBCDirectMedia = 2), and the endpoints are located behind the same NAT.

Notes:

- If you enable direct media by the SBCDirectMedia parameter, direct media is applied to all calls even if direct media is disabled per SIP Interface.
- If you configure direct media for all calls (using the SBCDirectMedia parameter), the device does not open voice channels nor allocate media ports for the calls, as the media always bypasses the device. In contrast, if you configure direct media for specific calls, the device allocates ports for these calls. The reason is that the ports may be required for mid-call services (e.g., early media, call forwarding, call transfer, and playing on-hold tones) handled by the server (IP PBX), which traverse the device. Therefore, make sure that you have allocated sufficient media ports (Media Realm) for such calls.
- Direct media cannot operate with the following features:
 - ✓ Manipulation of SDP data (offer-answer transaction) such as ports, IP address, coders
 - ✓ Force transcoding
 - ✓ Extension 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 direct media. Restricted coders are removed from the SDP offer message.
- For two users belonging to the same SIP Interface that is enabled for direct media and one of the users is defined as a foreign user (example, "follow me service") located in the WAN while the other is located in the LAN: calls between these two users cannot be established until direct media is disabled for the SIP Interface. The reason for this is that 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).



20.5.3 Restricting Audio Coders

You can configure a list of permitted (allowed) voice coders that can be used for a specific SIP entity (leg). In other words, you can enforce the use of specific coders. If the SDP offer in the incoming SIP message does not contain any coder that is configured as an allowed coder, the device rejects the calls (unless transcoding is implemented whereby Extension coders are added to the SDP, as described in [Coder Transcoding](#) on page 438). If the SDP offer contains some coders that are configured as allowed coders, the device manipulates the SDP offer by removing the coders that are not configured as allowed coders, before routing the SIP message to its destination. The device also re-orders (prioritizes) the coder list in the SDP according to the listed order of configured allowed coders.

For example, assume the following:

- The SDP offer in the incoming SIP message contains the G.729, G.711, and G.723 coders.
- The allowed coders configured for the SIP entity include G.711 and G.729.

The device removes the G.723 coder from the SDP offer, re-orders the coder list so that G.711 is listed first, and sends the SIP message containing only the G.711 and G.729 coders in the SDP.

The allowed coders are configured in the Allowed Audio Coders Group table. For more information, see ["Configuring Allowed Audio Coder Groups"](#) on page 467.



Note: If you assign the SIP entity an Allowed Coders Group for coder restriction and a Coders Group for extension coders (i.e., voice transcoding), the allowed coders take precedence over the extension coders. In other words, if an extension coder is not listed as an allowed coder, the device does not add the extension coder to the SDP offer.

20.5.4 Coder Transcoding

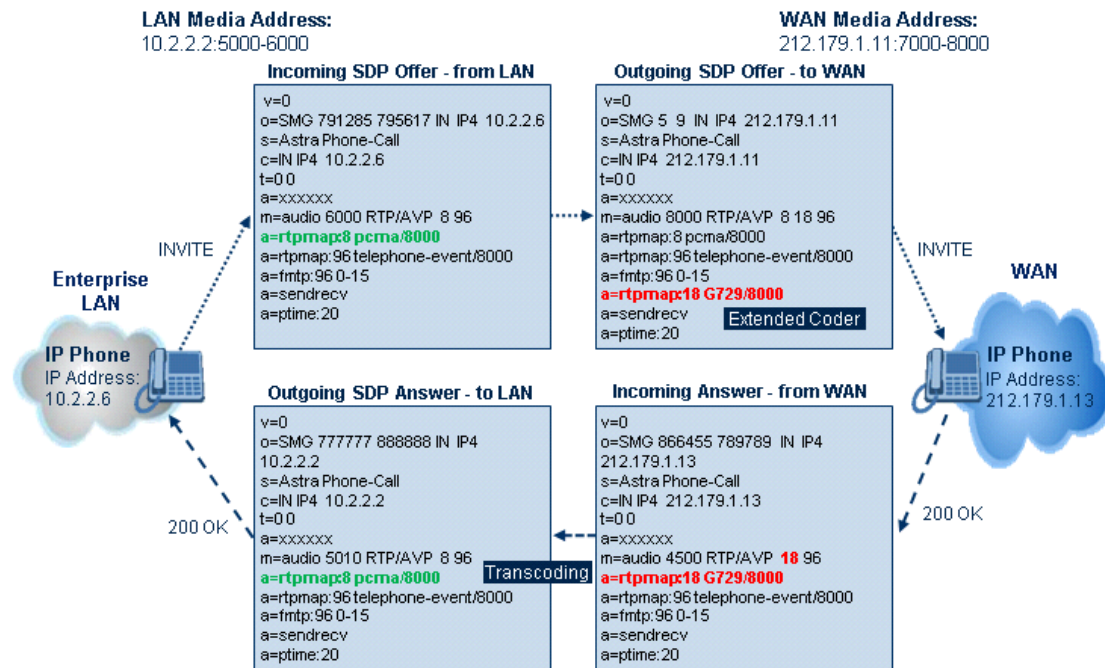
By default, the device forwards media packets transparently (i.e., no media negotiation) between SIP endpoints. However, when there are no common coders between two SIP entities that need to establish voice communication (i.e., the SDP answer from one SIP entity doesn't include any coder included in the SDP offer previously sent by the other), you can configure the device to perform audio coder transcoding between the inbound and outbound legs in order to enable media flow between them.

Transcoding may also be performed in scenarios where the same coder has been chosen between the legs, but where coder transrating is required. For example, the coders may use different coder settings such as rate and packetization time (G.729 at 20 ms to G.729 at 30 ms).

The coders that the device adds to the SDP offer on the outbound leg is referred to as *extension coders*. The extension coders are configured using Coder Groups (see ["Configuring Coder Groups"](#) on page 389), which you need to then assign to the IP Profile associated with the SIP entity.

The figure below illustrates transcoding between two SIP entities (IP Groups) where one uses G.711 (LAN IP phone) and the other G.729 (WAN IP phone). The initial SDP offer received on the inbound leg from the LAN IP phone includes coder G.711 as the supported coder. In the outgoing SDP offer on the outbound leg to the WAN IP phone, the device adds extension coder G.729 to the SDP, which is supported by the WAN IP phone. The subsequent incoming SDP answer from the WAN IP phone includes the G.729 coder as the chosen coder. Since this coder was not included in the original incoming SDP offer from the LAN IP phone, the device performs G.729-G.711 transcoding between the inbound and outbound legs.

Figure 20-5: Transcoding using Extended Coders (Example)

**Notes:**

- If you assign a SIP entity an Allowed Coders Group for coder restriction (allowed coders) and a Coders Group for extension coders, the allowed coders take precedence over the extension coders. In other words, if an extension coder is not listed as an allowed coder, the device does not add the extension coder to the SDP offer.
- If none of the coders in the incoming SDP offer on the inbound leg appear in the associated Allowed Audio Coders Group for coder restriction, the device rejects the call (sends a SIP 488 to the SIP entity that initiated the SDP offer).
- If none of the coders (including extension coders) in the outgoing SDP offer on the outbound leg appear in the associated Allowed Audio Coders Group for coder restriction, the device rejects the call (sends a SIP 488 to the SIP entity that initiated the SDP offer).
- For coder transcoding, the following prerequisites must be met (otherwise, the extension coders are not added to the SDP offer):
 - ✓ The device must support at least one of the coders listed in the incoming SDP offer.
 - ✓ The device must have available DSPs for both legs (inbound and outbound).
 - ✓ The incoming SDP offer must have at least one media line that is audio ('m=audio').
- The device adds the extension coders below the coder list received in the original SDP offer. This increases the chance of media flow without requiring transcoding.
- The device does not add extension coders that also appear in the original SDP offer.



As an example for using allowed and extension coders, assume the following:

■ **Inbound leg:**

- Incoming SDP offer includes the G.729, G.711, and G.723 coders.

```
m=audio 6050 RTP/AVP 18 0 8 4 96
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

The SDP "m=audio 6010 RTP/AVP 18 0 8 4 96" line shows the coder priority, where "18" (G.729) is highest and "4" (G.723) is lowest.

- Allowed Audio Coders Group for coder restriction includes the G.711 and G.729 coders (listed in order of appearance).

■ **Outbound leg:**

- Allowed Audio Coders Group for coder restriction includes the G.723, G.726, and G.729 coders (listed in order of appearance).
- Allowed Audio Coders Group for coder extension (transcoding) includes the G.726 coder.

1. On the inbound leg for the incoming SDP offer: The device allows and keeps the coders in the SDP that also appear in the Allowed Audio Coders Group for coder restriction (i.e., G.711 and G.729). It changes the order of listed coders in the SDP so that G.711 is listed first. The device removes the coders (i.e., G.723) from the SDP that do not appear in the Allowed Audio Coders Group for coder restriction.

```
m=audio 6050 RTP/AVP 0 8 18 96
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

2. On the outbound leg for the outgoing SDP offer: The SDP offer now includes only the G.711 and G.729 coders due to the coder restriction process on the incoming SDP offer (see Step 1).

- a. The device adds the extension coder to the SDP offer and therefore, the SDP offer now includes the G.711, G.729 and G.726 coders.

```
m=audio 6050 RTP/AVP 0 8 18 96 96
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:18 G729/8000
a=rtpmap:96 G726-32/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

- b. The device applies coder restriction to the SDP offer. As the Allowed Audio Coders Group for coder restriction includes the G.723, G.726, and G.729 coders, the device allows and keeps the G.729 and G.726, but removes the G.711 coder as it does not appear in the Allowed Audio Coders Group for coder restriction.

```
m=audio 6050 RTP/AVP 18 96 96
a=rtpmap:18 G729/8000
a=rtpmap:96 G726-32/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

- 3. The device includes only the G.729 and G.726 coders in the SDP offer that it sends from the outgoing leg to the outbound SIP entity. The G.729 is listed first as the Allowed Audio Coders Group for coder restriction takes precedence over the extension coder.

➤ **To configure coder transcoding:**

- 1. In the Coders Group Settings table, configure a Coders Group for extension coders. For more information, see "Configuring Coder Groups" on page 389.
- 2. In the IP Profile table, configure the IP Profile associated with the SIP entity:
 - a. Assign the Coders Group to the IP Profile, using the 'Extension Coders' parameter (SBCExtensionCodersGroupID).
 - b. Enable extension coders by configuring the 'Allowed Coders Mode' parameter to **Restriction** or **Restriction and Preference**.



Notes:

- To implement transcoding, you must configure the number of required DSP channels for transcoding (for example, MediaChannels = 120). Each transcoding session uses two DSP resources.
- The transcoding mode can be configured globally, using the TranscodingMode parameter or for specific calls, using the IP Profile table.

20.5.5 Transcoding Mode

By default, the device performs transcoding only if required. This refers to all types of transcoding (interworking) that require DSPs such as voice coder transcoding, DTMF negotiations, and fax negotiations. Transcoding is required, for example, when two SIP entities use different coders. In such a scenario, you would need to configure transcoding (i.e., extension coders), the device performs coder transcoding between the legs (inbound and outbound). If the SIP entities use the same coder, the device does not perform transcoding.

Alternatively, you can configure the device to always perform transcoding, regardless whether it is required or not. This is referred to as *forced* transcoding. For example, if the SIP entities use the same coder, the device performs transcoding of the same coder (e.g., G.711 and G.711) between the two legs.

The transcoding mode can be configured globally (TranscodingMode parameter) or per SIP entity using IP Profiles (IpProfile_TranscodingMode parameter).



Note: If the transcoding mode is configured to **Force** (i.e., always performs transcoding) for an IP Profile associated with a specific SIP entity, the device also applies forced transcoding for the SIP entity communicating with this SIP entity, regardless of its IP Profile settings.

20.5.6 Prioritizing Coder List in SDP Offer

In addition to restricting the use of coders using Allowed Coders Groups (see "Configuring Allowed Audio Coder Groups" on page 467), you can also prioritize the coders listed in the SDP offer. This feature is referred to as *Coder Preference* and applies to both SBC legs:

- **Incoming SDP offer:** The device arranges the coder list in the incoming SDP offer according to the order of appearance of the Allowed Coders Group that is associated with the incoming dialog. The coders listed higher up in the group take preference over ones listed lower down. To configure this, configure the 'Allowed Coders Mode' parameter (IpProfile_SBCAllowedCodersMode) in the associated IP Profile to **Preference** or **Restriction and Preference**. If you configure the parameter to **Preference**, the coders in the SDP offer that also appear in the Allowed Coders Group are listed first in the SDP offer, and the coders in the SDP offer that do not appear in the Allowed Coders Group are listed after the Allowed coders in the SDP offer. Therefore, this setting does not restrict coder use to Allowed coders, but uses (prefers) the Allowed coders whenever possible.
- **Outgoing SDP offer:** If only Allowed coders are used, the device arranges the coders in the SDP offer as described above. However, if Extension coders are also used, the coder list is arranged according to the SBCPreferencesMode parameter. Depending on the parameter's settings, the Extension coders are added after the Allowed coders according to their order in the Allowed Coders Group, or the Allowed and Extension coders are arranged according to their position in the Allowed Coders Group.

20.5.7 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, 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.

DSP resources are not required for RTP-SRTP transcoding.

20.5.8 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 supports 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 (e.g., does not support the coder), it relays the SBC dialog transparently.

20.5.9 Interworking Miscellaneous Media Handling

This section describes various interworking features relating to media handling.

20.5.9.1 Interworking DTMF Methods

The device supports interworking between various DTMF methods such as RFC 2833, In-Band DTMF's, and SIP INFO (Cisco\Nortel\Korea). By default, the device allows the remote user agents to negotiate (in case of RFC 2833) and passes DTMF without intervention. However, if two user agents (UA) support different DTMF methods, the device can interwork these different DTMF methods at each leg.

This DTMF interworking feature is enabled using IP Profiles (*ini* file parameter IPProfile):

- SBCRFC2833Behavior - affects the RFC 2833 SDP offer-answer negotiation:
 - [0] (default): the device does not intervene in the RFC 2833 negotiation.
 - [1]: each outgoing offer-answer includes RFC 2833 in the offered SDP (the device adds RFC 2833 only if the incoming offer does not include RFC 2833).
 - [2]: the device removes RFC 2833 from the incoming offer.
- SBCAlternativeDTMFMethod – the device's first priority for DTMF method at each leg is RFC 2833. Therefore, if a specific leg negotiates RFC 2833 successfully, then the chosen DTMF method for this leg is RFC 2833. For legs where RFC 2833 is not negotiated successfully, the device uses the parameter to determine the DTMF method for the leg.

The chosen DTMF method determines (for each leg) which DTMF method is used for sending DTMF's. If the device interworks between different DTMF methods and one of the methods is In-band\RFC 2833, detection and generation of DTMF methods requires DSP allocation.

20.5.9.2 Interworking RTP Redundancy

The device supports interworking of RTP redundancy (according to RFC 2198) between SIP entities. Employing IP Profiles, you can configure RTP redundancy handling per SIP entity:

- Generate RFC 2198 redundant packets (IpProfile_RTPRedundancyDepth parameter).
- Determine RTP redundancy support in the RTP redundancy negotiation in SDP offer/answer (IpProfile_SBCRTPRedundancyBehavior parameter). If not supported, the device discards RTP redundancy packets (if present) received from or sent to the SIP entity.

For more information, see the above parameters in "Configuring IP Profiles" on page 391.

20.5.9.3 Interworking RTP-RTCP Multiplexing

The device supports interworking of RTP-RTCP multiplexing onto a single, local UDP port (according to RFC 5761) between SIP entities. Employing IP Profiles, you can configure RTP multiplexing per SIP entity, using the `IPProfile_SBCRTCPMux` parameter (see "Configuring IP Profiles" on page 391).

20.5.9.4 Interworking RTCP Attribute in SDP

The device supports interworking the RTCP attribute 'a=rtcp' in the SDP between SIP entities. Employing IP Profiles, you can configure RTCP attribute handling (add, remove or transparent) per SIP entity, using the `IpProfile_SBCSDPHandleRTCPAttribute` parameter (see "Configuring IP Profiles" on page 391).

20.5.9.5 Interworking Crypto Lifetime Field

The device supports interworking the lifetime field in the 'a=crypto' attribute of the SDP, between SIP entities. Employing IP Profiles, you can configure the lifetime field handling (remove or retain) per SIP entity, using the `IpProfile_SBCRemoveCryptoLifetimeInSDP` parameter (see "Configuring IP Profiles" on page 391).

20.5.9.6 Interworking Media Security Protocols

The device supports interworking media security protocols for SRTP, between SIP entities. Employing IP Profiles, you can configure the security protocol (SDS or DTLS) per SIP entity, using the `IPProfile_SBCMediaSecurityMethod` parameter (see "Configuring IP Profiles" on page 391). For more information on SDS and DTLS, see "Configuring Media (SRTP) Security" on page 211.

20.5.9.7 Interworking ICE Lite for NAT Traversal

The device supports interworking ICE for NAT traversal, between SIP entities. Employing IP Profiles, you can enable ICE Lite per SIP entity, using the `IPProfile_SBCIceMode` parameter (see "Configuring IP Profiles" on page 391).

20.6 Fax Negotiation and Transcoding

The device can allow fax transmissions to traverse transparently without transcoding or it can handle the fax as follows:

- Allow interoperability between different fax machines, supporting fax transcoding if required.
- Restrict usage of specific fax coders to save bandwidth, enhance performance, or comply with supported coders. These coders include G.711 (A-Law or Mu-Law), VBD (G.711 A-Law or G.711 Mu-Law), and T38.

Fax configuration is done in the Coder Group Settings table and IP Profile table. The Coder Group Settings table defines the supported coders, which is assigned to the IP Profile associated with the SIP entity. The IP Profile table also defines the negotiation method used between the incoming and outgoing fax legs, using the following fax-related parameters:

- SBCFaxBehavior: defines the offer negotiation method - pass fax transparently, negotiate fax according to fax settings in IP Profile, or enforce remote UA to first establish a voice channel before fax negotiation.
- SBCFaxCodersGroupID: defines the supported fax coders (from the Coders Group Settings table).
- SBCFaxOfferMode: determines the fax coders sent in the outgoing SDP offer.
- SBCFaxAnswerMode: determines the fax coders sent in the outgoing SDP answer.
- IPProfile_SBCRemoteRenegotiateOnFaxDetection: You can also configure the device to detect for faxes (CNG tone) immediately after the establishment of a voice channel (i.e., after 200 OK) and within a user-defined interval. If detected, it can then handle the subsequent fax renegotiation by sending re-INVITE messages to both SIP entities (originating and terminating faxes). For more information, see the parameter in "Configuring IP Profiles" on page 391.



Note: The voice-related coder configuration (Allowed and Extension coders) is independent of the fax-related coder configuration, with the exception of the G.711 coder. If the G.711 coder is restricted by the Allowed Coders Group table, it is not used for fax processing even if it is listed in the Coders Group Settings table for faxes. However, support for G.711 coders for voice is not dependent upon which fax coders are listed in the Coders Group Settings table.

20.7 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.

20.8 SBC Authentication

The device can authenticate SIP servers and SBC users (clients). The different authentication methods are described in the subsequent subsections.

20.8.1 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 591). 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 348).

20.8.2 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.

20.9 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.

20.9.1 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 Mode'.

20.9.1.1 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:

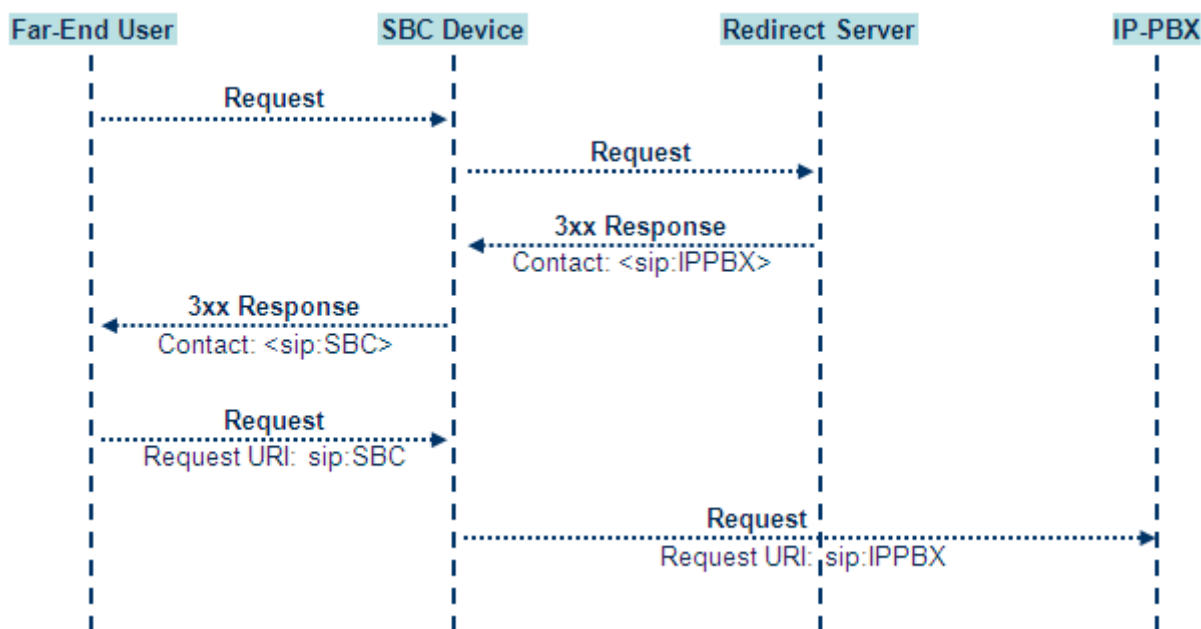
- Changes the host part to the device's IP address – this change causes the remote user agent to send the INVITE to the device.
- Adds a special prefix ("T~&R_") to the Contact user part – to identify the new INVITE as a 3xx resultant INVITE.

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 20-6: SIP 3xx Response Handling



The process of this feature is described using an example:

1. The device receives the Redirect server's SIP 3xx response (e.g., Contact: <sip:User@IPPBX:5060;transport=tcp;param=a>;q=0.5).
2. The device replaces the Contact header value with the special prefix and database key value as user part, and with the device's URL as host part (e.g., Contact: <sip:Prefix_Key_User@SBC:5070;transport=udp>;q=0.5).
3. The device sends this manipulated SIP 3xx response to the Far-End User (FEU).
4. The FEU sends a new request with the Request-URI set to the value of the received 3xx response's Contact header (e.g., RequestURI: sip:Prefix_Key_User@SBC:5070;transport=udp).
5. Upon receipt of the new request from the FEU, the device replaces the Request-URI with the new destination address (e.g., RequestURI: sip:Prefix_User@IPPBX:5070;transport=tcp;param=a).
6. The device removes the user prefix from the Request-URI, and then sends this Request-URI to the new destination (e.g., RequestURI: sip:User@IPPBX:5070;transport=tcp;param=a).

20.9.1.2 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).

20.9.2 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 20-1: 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.

Parameter Value	SIP Header Present in Received SIP Message		
HistoryInfoMode = Remove DiversionMode = Remove	Diversion removed.	History-Info removed.	Both removed.

20.9.3 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, 'Remote REFER Mode'.

- 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.

20.9.4 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.

20.9.5 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'.

20.9.6 Interworking SIP Early Media

The device supports early media. Early media is when the media flow starts before the SIP call is established (i.e., before the 200 OK response). This occurs when the first SDP offer-answer transaction completes. The offer-answer options can be included in the following SIP messages:

- Offer in first INVITE, answer on 180, and no or same answer in the 200 OK
- Offer in first INVITE, answer on 180, and a different answer in the 200 OK (not standard)
- INVITE without SDP, offer in 180, and answer in PRACK
- PRACK and UPDATE transactions can also be used for initiating subsequent offer-answer transactions before the INVITE 200 OK response.
- In a SIP dialog life time, media characteristics after originally determined by the first offer-answer transaction can be changed by using subsequent offer-answer transactions. These transactions may be carried either in UPDATE or re-INVITE transactions. The media handling is similar to the original offer-answer handling. If the offer is rejected by the remote party, no media changes occur (e.g., INVITE without SDP, then 200 OK and ACK, offer-answer within an offer-answer, and Hold re-INVITE with IP address of 0.0.0.0 - IP address is unchanged).

The device supports various interworking modes for 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 the 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, 'Remote Early Media RTP Detection Mode', 'SBC Remote Supports RFC 3960', and 'SBC Remote Can Play Ringback'. See the flowcharts below for the device's handling of such scenarios:

Figure 20-7: SBC Early Media RTP 18x without SDP

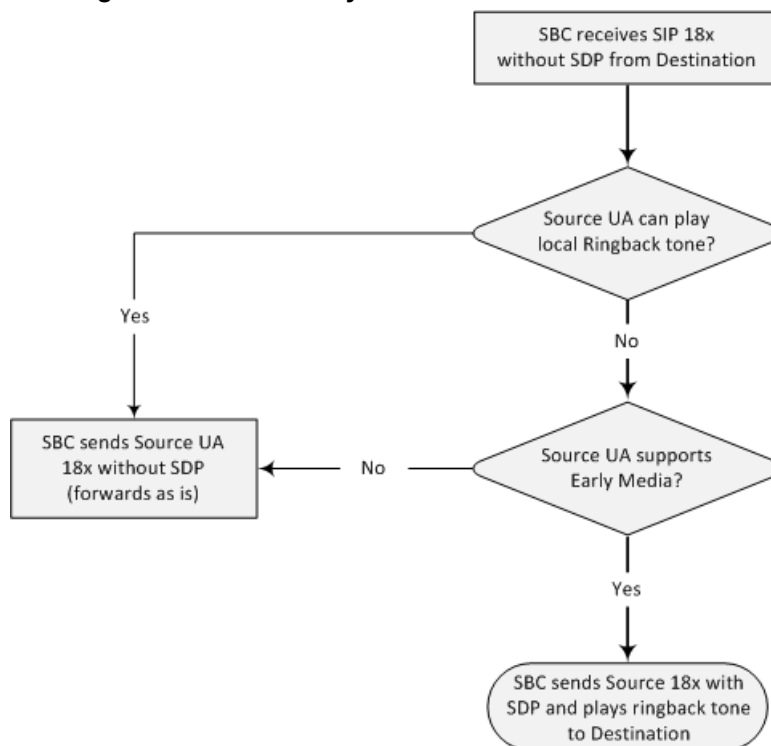
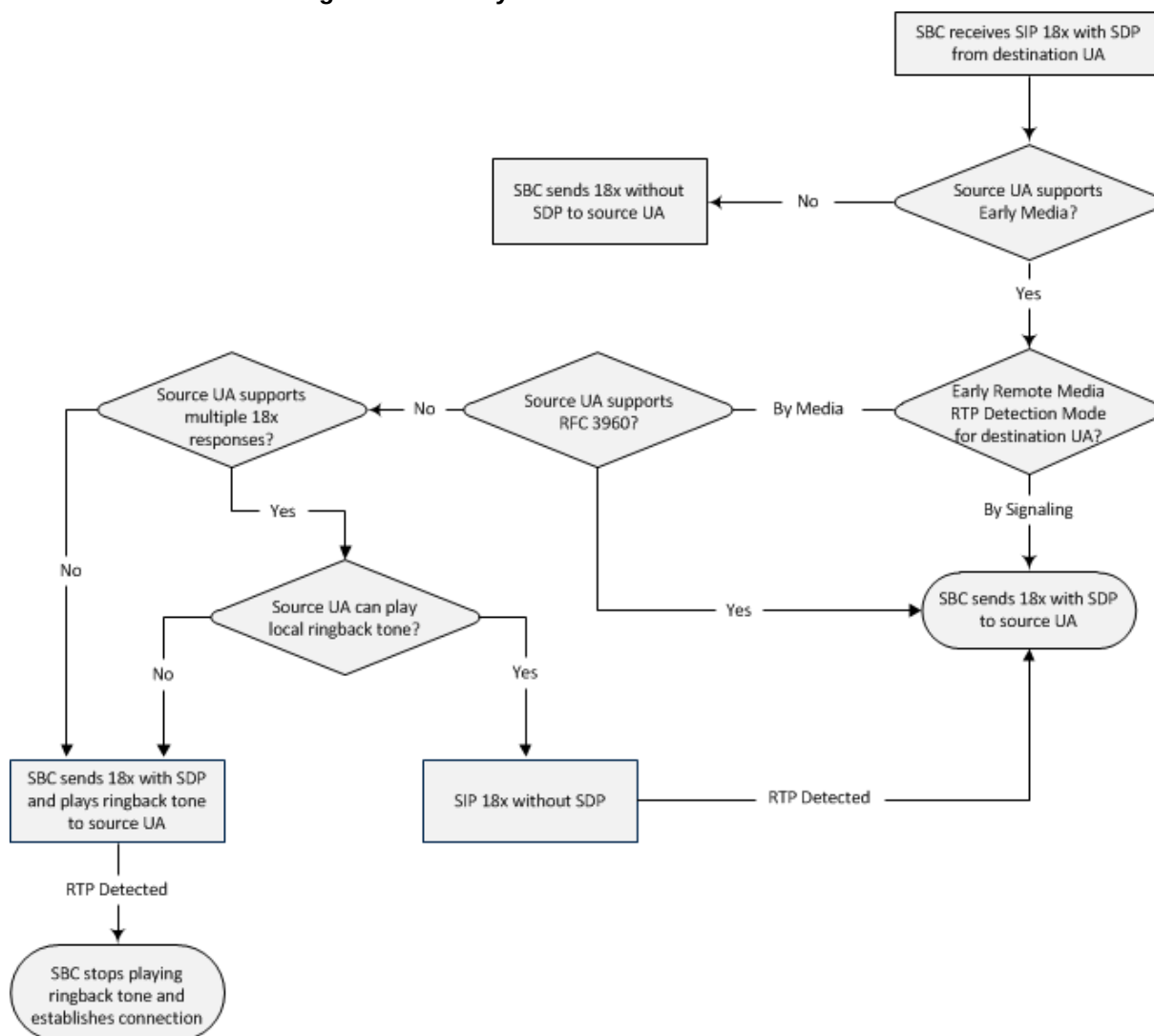


Figure 20-8: Early Media RTP - SIP 18x with SDP



20.9.7 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'.

20.9.8 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'.

20.9.9 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.

20.9.10 Interworking Delayed Offer

The device supports interworking of INVITE messages with and without SDP between SIP entities. 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. This intervention in the SDP offer-answer process may require transcoding. Delayed offer is also supported when early media is present.

Employing IP Profiles, you can configure this interworking feature per SIP entity, using the 'SBC Remote Delayed Offer Support' parameter (see "Configuring IP Profiles" on page 391).



Note: For SIP entities that do not support delayed offer, you must assign extension coders to its IP Profile (using the 'Extension Coders' parameter).

20.9.11 Interworking Call Hold

The device supports the interworking of call hold / retrieve requests between SIP 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 391.

20.9.12 Interworking SIP Via Headers

The device supports the interworking of SIP Via headers between SIP entities. For the outgoing message sent to a SIP entity, the device can remove or retain all the Via headers received in the incoming SIP request from the other side. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the `IpProfile_SBCKeepVIAHeaders` parameter (see "Configuring IP Profiles" on page 391).

20.9.13 Interworking SIP User-Agent Headers

The device supports the interworking of SIP User-Agent headers between SIP entities. For the outgoing message sent to a SIP entity, the device can remove or retain all the User-Agent headers received in the incoming SIP request/response from the other side. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the `IpProfile_SBCKeepUserAgentHeader` parameter (see "Configuring IP Profiles" on page 391).

20.9.14 Interworking SIP Record-Route Headers

The device supports the interworking of SIP Record-Route headers between IP entities. For the outgoing message sent to a SIP entity, the device can remove or retain all the Record-Route headers received in the incoming SIP request/response from the other side. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the `IpProfile_SBCKeepRoutingHeaders` parameter (see "Configuring IP Profiles" on page 391).

20.9.15 Interworking SIP To-Header Tags in Multiple SDP Answers

The device supports the interworking of SIP To-header tags in call forking responses (i.e., multiple SDP answers) between IP entities. The device can either use the same To-header tag value for all SDP answers sent to the SIP entity, or send each SDP answer with its original tag. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the `IpProfile_SBCRemoteMultipleEarlyDialogs` parameter (see "Configuring IP Profiles" on page 391).

20.9.16 Interworking In-dialog SIP Contact and Record-Route Headers

The device supports the interworking of in-dialog, SIP Contact and Record-Route headers between SIP entities. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the `IpProfile_SBCRemoteRepresentationMode` parameter (see "Configuring IP Profiles" on page 391).

This page is intentionally left blank.

21 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 596.

➤ **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.

This page is intentionally left blank.

22 Configuring General SBC Settings

The General Settings page allows you to configure general SBC parameters. For a description of these parameters, see "SBC Parameters" on page 804.

➤ **To configure general parameters:**

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

Figure 22-1: 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

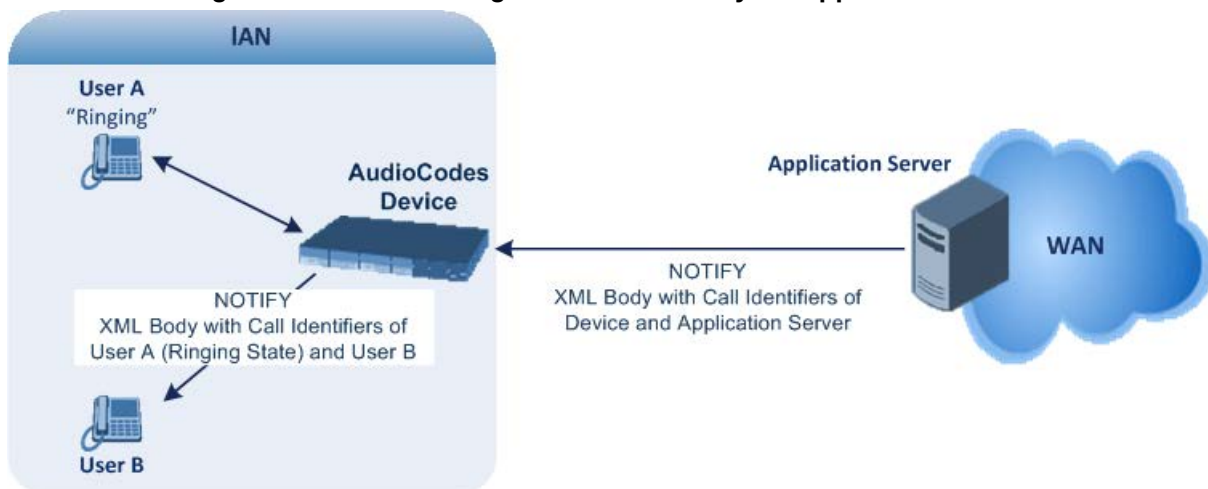
2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 578.

22.1 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 22-2: 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>
```

This page is intentionally left blank.

23 Configuring Admission Control

The Admission Control table lets you configure up to 625 Call Admission Control rules (CAC). CAC rules define the maximum number of concurrent calls (SIP dialogs) permitted per IP Group, SIP Interface or SRD, and per user (identified by its registered contact). 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 an SRD and each of its associated IP Groups, by configuring multiple CAC rules. 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 SRD'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 through the Web interface. You can also configure it through ini file (SBCAdmissionControl) or CLI (configure voip > sbc sbc-admission-control).

➤ **To configure a CAC rule:**

1. Open the Admission Control table (**Configuration** tab > **VoIP** menu > **SBC** > **Admission Control**).
2. Click **Add**; the following dialog box appears:

Figure 23-1: Admission Control Table - Add Row Dialog Box

3. Configure an Admission Control rule according to the parameters described in the table below.
4. Click **Add**.

Table 23-1: Admission Control Table Parameter Description

Parameter	Description
Index [SBCAdmissionControl_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no value is

Parameter	Description
admission-name [SBCAdmissionControl_ AdmissionControlName]	defined.
Limit Type limit-type [SBCAdmissionControl_ LimitType]	Defines the entity to which the rule applies. ▪ [0] IP Group (default) ▪ [1] SRD ▪ [2] SIP Interface
SRD srd-name [SBCAdmissionControl_ SRDName]	Assigns an SRD to the rule. For all SRDs, configure the parameter to Any . By default, no value is defined (None). Note: The parameter is applicable only if 'Limit Type' is configured to SRD .
IP Group ip-group-name [SBCAdmissionControl_ IPGroupName]	Assigns an IP Group to the rule. For all IP Groups, configure the parameter to Any . By default, no value is defined (None). Note: The parameter is applicable only if 'Limit Type' is configured to IP Group .
SIP Interface sip-interface-name [SBCAdmissionControl_ SIPInterfaceName]	Assigns a SIP Interface to the rule. For all SIP Interfaces, configure the parameter to Any . By default, no value is defined (None). Note: The parameter is applicable only if 'Limit Type' is configured to SIP Interface .
Request Type request-type [SBCAdmissionControl_ RequestType]	Defines the SIP dialog-initiating request type to which you want to apply the rule (not the subsequent requests that can be of different type and direction). ▪ [0] All = (Default) Includes the total number of all dialogs. ▪ [1] INVITE ▪ [2] SUBSCRIBE ▪ [3] Other
Request Direction 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.
Reserved Capacity reservation [SBCAdmissionControl_ Reservation]	Defines the guaranteed (minimum) call capacity. The default is 0 (i.e., no reserved capacity). Notes: ▪ Reserved call capacity is applicable only to IP Groups and SRDs (i.e., the 'Limit Type' parameter must be configured to IP Group or SRD). If you configure the 'Limit Type' parameter to SIP Interface, leave the 'Reserved Capacity' parameter at its' default (i.e., 0). ▪ 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.

Parameter	Description
Limit limit [SBCAdmissionControl_Limit]	Defines the maximum number of concurrent SIP dialogs per IP Group, SIP Interface or SRD. You can also use the following special values: ▪ [0] 0 = Block all these dialogs. ▪ [-1] -1 = (Default) Unlimited.
Limit Per User limit-per-user [SBCAdmissionControl_LimitPerUser]	Defines the maximum number of concurrent SIP dialogs per user belonging to the specified IP Group, SIP Interface or SRD. You can also use the following special values: ▪ [0] 0 = Block all these dialogs. ▪ [-1] -1 = (Default) Unlimited.
Rate rate [SBCAdmissionControl_Rate]	Defines the maximum number of SIP dialogs per IP Group, SIP Interface or SRD that can be handled per second. 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 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, SIP Interface, SRD, SIP request type, and SIP request direction.

24 Configuring Coder Groups

24.1 Configuring Allowed Audio Coder Groups

The Allowed Audio Coders Group table lets you configure up to 20 Allowed Audio Coders Groups. For each Allowed Audio Coders Group, you can configure up to 10 audio coders, which can include default coders and user-defined (string) coders for non-standard or unknown coders.

Allowed Audio Coders Groups restrict coders used for SIP entities. Only coders listed in the Allowed Audio Coders Group (i.e., allowed coders) that is associated with the SIP entity can be used. If the coders in the SDP offer ('a=rtpmap' field) of the incoming SIP message are not listed in the Allowed Audio Coders Group, the device rejects the calls, unless transcoding is configured, whereby extension coders are added to the SDP, as described in Coder Transcoding on page 438. If the SDP offer contains some coders that are listed in the Allowed Audio Coders Group, the device manipulates the SDP offer by removing the coders that are not listed in the Allowed Audio Coders Group, before routing the SIP message to its destination. Thus, only coders that are common between the coders in the SDP offer and the coders in the Allowed Audio Coders Group are used. For more information on coder restriction, see "Restricting Audio Coders" on page 438.

For example, assume the following:

- The SDP offer in the incoming SIP message contains the G.729, G.711, and G.723 coders.
- The allowed coders configured for the SIP entity include G.711 and G.729.

The device removes the G.723 coder from the SDP offer, re-orders the coder list so that G.711 is listed first, and sends the SIP message containing only the G.711 and G.729 coders in the SDP.

To apply an Allowed Audio Coders Group for restricting coders to a SIP entity:

1. Configure an Allowed Audio Coders Group in the Allowed Audio Coders Group table (see description below).
2. In the IP Profile associated with the SIP entity (see "Configuring IP Profiles" on page 391):
 - Assign the Allowed Audio Coders Group (using the `IpProfile_SBCAllowedCodersGroupID` parameter).
 - Enable the use of Allowed Audio Coder Groups (by configuring the `IpProfile_SBCAllowedCodersMode` parameter to **Restriction** or **Restriction and Preference**).

The device also re-orders (prioritizes) the coder list in the SDP according to the order of appearance of the coders listed in the Allowed Audio Coders Group. The first listed coder has the highest priority and the last coder has the lowest priority. For more information, see "Prioritizing Coder List in SDP Offer" on page 442.



Notes:

- For configuring coders (extension coders) to add to the SDP offer for audio transcoding, you need to use the Coder Group Settings table (see Configuring Coder Groups on page 389).
- The Allowed Audio Coders Group for coder restriction takes precedence over the Coder Group for extension coders. In other words, if an extension coder is not listed as an allowed coder, the device does not add the extension coder to the SDP offer.

The following procedure describes how to configure Allowed Audio Coder Groups through the Web interface. You can also configure it through ini file (AllowedCodersGroup) or CLI (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 24-1: Allowed Audio Coders Group Page

2. Configure an Allowed Audio Coders Group according to the parameters described in the table below.
3. Click **Submit**.

Table 24-1: Allowed Audio Coders Group Table Parameter Descriptions

Parameter	Description
Allowed Audio Coders Group ID [AllowedCodersGroupX]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Coder Name 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.

24.2 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 391). 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 Audio Coders" on page 438.

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 442.

Currently, the Allowed Video Coder Groups table can only be configured through ini file (AllowedVideoCodersGroup) or CLI (configure voip > sbc allowed-video-coders-group group-0). The table below describes the parameter.

Table 24-2: Allowed Video Coders Group Table Parameter Descriptions

Parameter	Description
Allowed Coders Group ID [AllowedVideoCodersGroupX]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Coder Name 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.

This page is intentionally left blank.

25 Routing SBC

This section describes the configuration of the call routing entities for the SBC application.

25.1 Configuring Classification Rules

The Classification table lets you configure up to 625 Classification rules. A Classification rule classifies incoming SIP dialog-initiating requests (e.g., INVITE messages) to a "source" IP Group. The source IP Group is the SIP entity that sent the SIP dialog request. Once classified, the device uses the IP Group to process the call (manipulation and routing).

You can also use the Classification table for employing SIP-level access control for successfully classified calls, by configuring Classification rules with whitelist and blacklist settings. If a Classification rule is configured as a whitelist ("Allow"), the device accepts the SIP dialog and processes the call. If the Classification rule is configured as a blacklist ("Deny"), the device rejects the SIP dialog.

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). Classification is primarily based on the SIP Interface (as the matching characteristics) on which the incoming dialog is received. As Classification rules must first be assigned with an SRD, the SIP Interface is one that belongs to the SRD. Therefore, Classification rules are configured per SRD, where multiple SIP Interfaces can be used as matching characteristics. However, as multiple SRDs are relevant only for multi-tenant deployments, for most deployments only a single SRD is required. As the device provides a default SRD ("Default_SRD"), when only one SRD is required, the device automatically assigns it to the Classification rule.
- **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. If it doesn't find a matching rule (i.e., classification fails), the device either rejects or allows the call depending on the setting of the 'Unclassified Calls' parameter (see Configuring General SBC Settings on page 459). If the parameter is set to Allow, the incoming SIP dialog is assigned to an IP Group as follows:

1. The device determines on which SIP listening port (e.g., 5061) the incoming SIP dialog request was received and the SIP Interface configured with this port (in the SIP Interface table).
2. The device determines the SRD associated with this SIP Interface (in the SIP Interface table) and then classifies the SIP dialog to the first IP Group in the IP Group table that is associated with the SRD. For example, if IP Groups 3 and 4 belong to the same SRD, the device classifies the call to IP Group 3.



Note: If classification of 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 342).

The Classification table is used to classify incoming SIP dialog requests **only** if the following classification stages fail:

1. **Classification Stage 1 - Based on User Registration Database:** The device searches its users registration database to check whether the incoming SIP dialog arrived from a registered user. The device searches the database for a user that matches the address-of-record (AOR) and Contact of the incoming SIP message:

- Compares the SIP Contact header to the contact value of the user in the database.
- Compares the URL in the SIP P-Asserted-Identity/From header to the registered address-of-record (AOR) in the database.

If the device finds a matching registered user, it classifies the user to the IP Group associated with the user in the database. If this classification 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 348). 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.

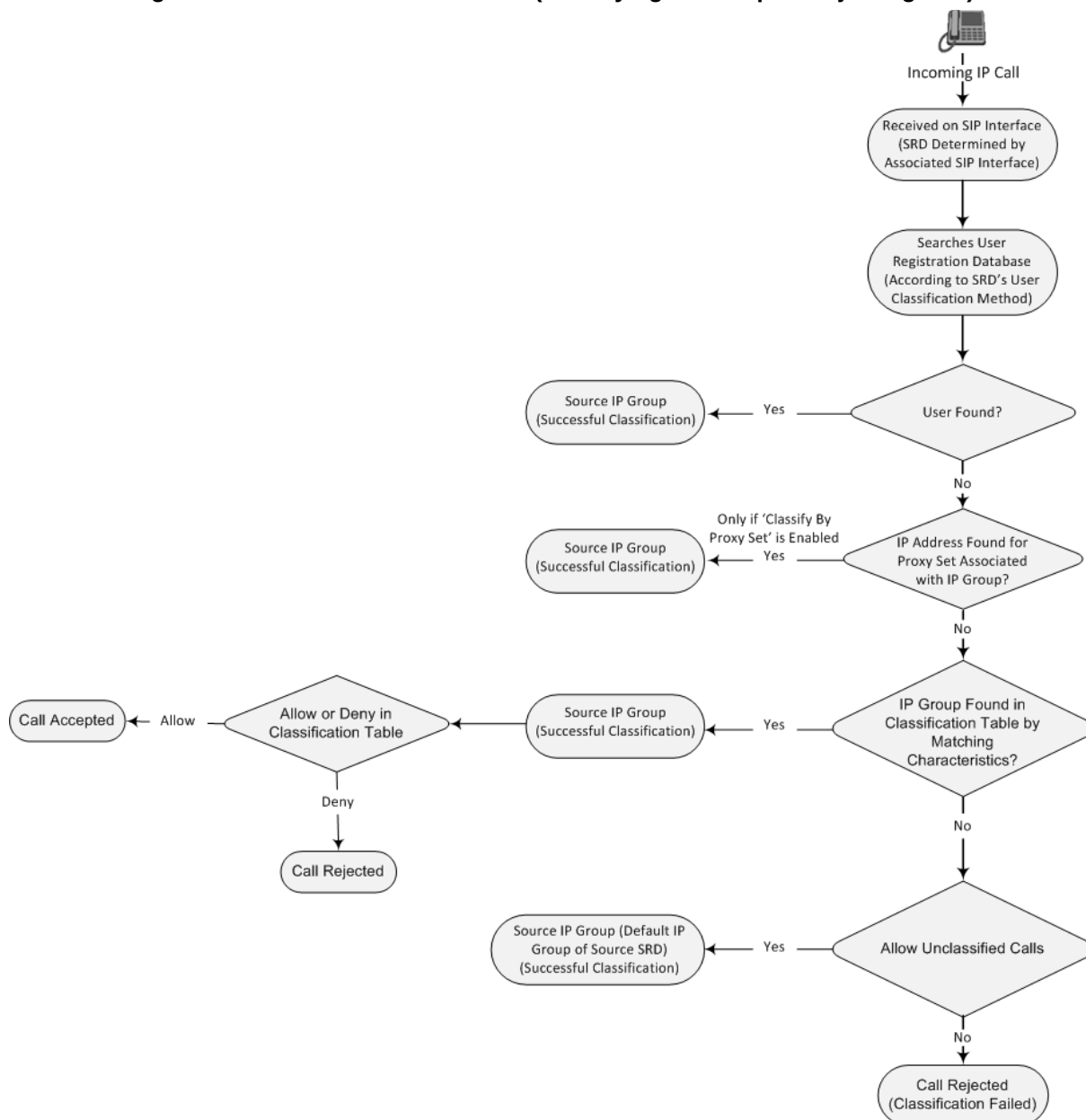


Note:

- 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).
- 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 flowchart below illustrates the classification process:

Figure 25-1: Classification Process (Identifying IP Group or Rejecting Call)



The following procedure describes how to configure Classification rules through the Web interface. You can also configure it through ini file (Classification) or CLI (configure voip > sbc routing classification).

➤ To configure a Classification rule:

1. Open the Classification table (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Classification Table**).
2. Click **Add**; the following dialog box appears:

Figure 25-2: Classification Table - Add Row Dialog Box

3. Configure the Classification rule according to the parameters described in the table below.
4. Click **Add**.

Table 25-1: Classification Table Parameter Descriptions

Parameter	Description
Index [Classification_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name classification-name [Classification_ClassificationName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no name is defined. Note: Each row must be configured with a unique name.

Parameter	Description
Rule (Matching Characteristics)	
SRD srd-name [Classification_SRDName]	Assigns an SRD to the rule as a matching characteristic for the incoming SIP dialog. If only one SRD is configured in the SRD table, the SRD is assigned to the rule, by default. If multiple SRDs are configured in the SRD table, no value is assigned. For configuring SRDs, see "Configuring SRDs" on page 332. Note: The parameter is mandatory.
Source SIP Interface src-sip-interface-name [Classification_SrcSIPInterfaceName]	Assigns a SIP Interface to the rule as a matching characteristic for the incoming SIP dialog. The default is Any (i.e., all SIP Interfaces belonging to the SRD assigned to the rule). Note: The SIP Interface must belong to the SRD assigned to the rule (see the 'SRD' parameter in the table).
Source IP Address src-ip-address [Classification_SrcAddress]	Defines a source IP address as a matching characteristic for the incoming SIP dialog. The valid value is an IP address in dotted-decimal notation. In addition, the following wildcards can be used: ▪ "x" wildcard: represents single digits. For example, 10.8.8.xx represents all addresses between 10.8.8.10 and 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. By default, no value is defined (i.e., 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 Transport Type src-transport-type [Classification_SrcTransportType]	Defines the source transport type as a matching characteristic for the incoming SIP dialog. ▪ [-1] Any = (Default) All transport types ▪ [0] UDP ▪ [1] TCP ▪ [2] TLS
Source Port src-port [Classification_SrcPort]	Defines the source port number as a matching characteristic for the incoming SIP dialog.
Source Username Prefix src-user-name-prefix [Classification_SrcUsernamePrefix]	Defines the prefix of the source URI user part as a matching characteristic for 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, in the IP Group table ('Source URI Input' parameter). For more information on how the device obtains the URI, see "SIP Dialog Initiation Process" on page 428.

Parameter	Description
	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 715. Note: For REGISTER requests, the source URI is obtained from the To header.
Source Host src-host [Classification_SrcHost]	Defines the prefix of the source URI host name as a matching characteristic for 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, in the IP Group table ('Source URI Input' parameter). For more information on how the device obtains this URI, see "Call Processing of SIP Dialog Requests" on page 428. The default is the asterisk (*) symbol, which represents any source host prefix. Note: For REGISTER requests, the source URI is obtained from the To header.
Destination Username Prefix dst-user-name-prefix [Classification_DestUserNamePrefix]	Defines the prefix of the destination Request-URI user part as a matching characteristic for 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 715.
Destination Host dst-host [Classification_DestHost]	Defines the prefix of the destination Request-URI host name as a matching characteristic for the incoming SIP dialog. The default is the asterisk (*) symbol, which represents any destination host prefix.
Message Condition message-condition-name [Classification_MessageConditionName]	Assigns a Message Condition rule to the Classification rule as a matching characteristic for the incoming SIP dialog. By default, no value is defined (None). To configure Message Condition rules, see "Configuring Message Condition Rules" on page 478.
Action	
Action Type action-type [Classification_ActionType]	Defines a whitelist or blacklist for the matched incoming SIP dialog. [0] Deny = Blocks incoming SIP dialogs that match the characteristics of the rule (blacklist). [1] Allow = (Default) Allows incoming SIP dialogs that match the characteristics of the rule (whitelist) and assigns it to the associated IP Group.
Destination Routing Policy dest-routing-policy [Classification_DestRoutingPolicy]	Assigns an SBC Routing Policy to the matched incoming SIP dialog. The assigned SBC Routing Policy overrides the SBC Routing Policy assigned to the SRD (in the SRD table). The option to assign Routing Policies to Classification rules is useful in deployments requiring different routing and manipulation rules for specific calls pertaining to the same SRD. In such scenarios, you need to configure multiple Classification rules for the same SRD, where for some rules no Routing Policy is assigned (i.e., the SRD's assigned Routing Policy is used) while for others a different Routing Policy is specified to override the SRD's assigned Routing Policy.

Parameter	Description
	By default, no value is defined (None). For configuring SBC Routing Policies, see "Configuring SBC Routing Policy Rules" on page 493.
Source IP Group src-ip-group-name [Classification_SrcIPGroup upName]	Assigns an IP Group to the matched incoming SIP dialog. By default, no value is defined (None). For configuring IP Groups, see "Configuring IP Groups" on page 348. Note: The IP Group must be associated with the assigned SRD (see the 'SRD' parameter in the table).
IP Profile ip-profile-id [Classification_IpProfileName]	Assigns an IP Profile to the matched incoming SIP dialog. The assigned IP Profile overrides the IP Profile assigned to the IP Group (in the IP Group table) to which the SIP dialog is classified. Therefore, assigning an IP Profile during classification allows you to assign different IP Profiles to specific users (calls) that belong to the same IP Group (User or Server type). For example, you can configure two Classification rules to classify incoming calls to the same IP Group. However, one Classification rule is a regular rule that doesn't specify any IP Profile (IP Profile assigned to IP Group is used), while the second rule is configured with an additional matching characteristic for the source hostname prefix (e.g., "abcd.com") and with an additional action that assigns a different IP Profile. By default, no value is defined (None). Note: For User-type IP Groups, if a user is already registered with the device (from a previous, initial classification process), the device classifies subsequent INVITE requests from the user according to the device's users database instead of the Classification table. In such a scenario, the same IP Profile that was previously assigned to the user by the Classification table is also used (in other words, the device's users database stores the associated IP Profile).

25.1.1 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=z9hG4bKVEBTDHAHSUYRTEXEDEGJY
From: <sip:100@10.33.4.226>;tag=YSQQKXXREVDPTYPTNFMWG
To: <sip:8000@10.33.4.226>
Call-ID: FKPN0YRNKROIMEGBSSKS@10.33.4.226
CSeq: 1 INVITE
Contact: <sip:100@10.33.4.226>
Route: <sip:2000@10.10.10.10>,<sip:300@10.10.10.30>
Supported: em,100rel,timer,replaces
P-Called-Party-ID: <sip:1111@10.33.38.1>
User-Agent: Sip Message Generator V1.0.0.5
Content-Length: 0
```

1. In the Classification table, add the following classification rules:

Index	Source Username Prefix	Destination Username Prefix	Destination Host	Source IP Group
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 2.

25.2 Configuring Message Condition Rules

The Message Condition table lets you configure up to 500 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 471)
- IP-to-IP routing rules in the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 480)
- IP-to-IP outbound manipulation rules in the IP to IP Outbound Manipulation table (see "Configuring IP-to-IP Outbound Manipulations" on page 504)

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 377). 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 through the Web interface. You can also configure it through ini file (ConditionTable) or CLI (configure voip > sbc routing condition-table).

➤ **To configure a Message Condition rule:**

1. Open the Message Condition table (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Message Condition Table**).
2. Click **Add**; the following dialog box appears:

Figure 25-3: Message Condition Table - Add Row Dialog Box

The dialog box titled 'Add Row' contains three input fields: 'Index' with the value '0', 'Name', and 'Condition'. At the bottom right, there are two buttons: 'Add' and 'Cancel'.

3. Configure a Message Condition rule according to the parameters described in the table below.
4. Click **Add**.

An example of configured Message Condition rules is shown in the figure below:

Figure 25-4: Example of Configured SIP Message Conditions

Index	Name	Condition
0	IP Group user	param.ipg.src.type==user
1	Contains SIP Via header	header.via.exists
2	101 user part in From header	header.from.url.user=='101'

- **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 25-2: Message Condition Table Parameter Descriptions

Parameter	Description
Index [ConditionTable_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [ConditionTable_Name]	Defines a brief description of the Condition rule. The valid value is a string of up to 59 characters.
Condition 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.

25.3 Configuring SBC IP-to-IP Routing

The IP-to-IP Routing table lets you configure up to 3,750 SBC IP-to-IP routing rules.

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

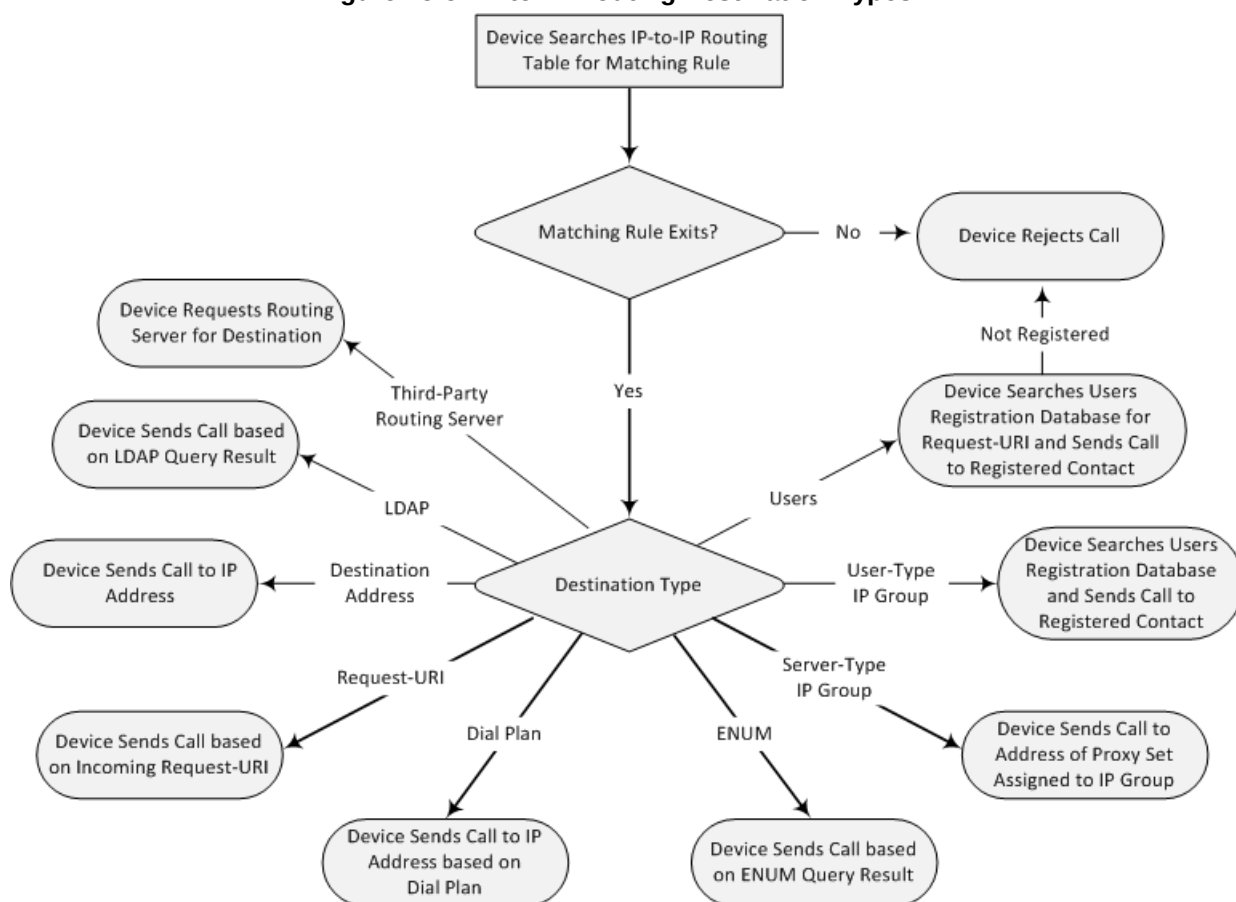
- **Rule:** Defines the characteristics of the incoming SIP dialog message (e.g., IP Group from which the message is received).
- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule (i.e., routes the call to the specified destination).

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.

An IP-to-IP routing rule routes received SIP dialog messages (e.g., INVITE) to any of the following configurable IP destinations:

- According to registered user Contact listed in the device's registration database (only for User-type IP Groups).
- IP Group - the destination is the address configured for the Proxy Set associated with the IP Group.
- 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.
- Any registered user in the registration database. If the Request-URI of the incoming INVITE exists in the database, the call is sent to the corresponding contact address specified in the database.
- According to result of an ENUM query.
- Hunt Group - used for call survivability of call centers (see "Call Survivability for Call Centers" on page 536).
- According to result of LDAP query (for more information on LDAP-based routing, see "Routing Based on LDAP Active Directory Queries" on page 241).
- Third-party routing server, which determines the destination (next hop) of the call (IP Group). The IP Group represents the next device in the routing path to the final destination. For more information, see "Centralized Third-Party Routing Server" on page 280.

Figure 25-5: IP-to-IP Routing Destination Types



To configure and apply an IP-to-IP Routing rule, the rule must be associated with a Routing Policy. The Routing Policy associates the routing rule with an SRD(s). Therefore, the Routing Policy lets you configure routing rules for calls belonging to specific SRD(s). However, as multiple Routing Policies are relevant only for multi-tenant deployments (if needed), for most deployments, only a single Routing Policy is required. As the device provides a default Routing Policy ("Default_SBCRoutingPolicy"), when only one Routing Policy is required, the device automatically assigns the default Routing Policy to the routing rule. If you are implementing LDAP-based routing (with or without Call Setup Rules) and/or Least Cost Routing (LCR), you need to configure these settings for the Routing Policy (regardless of the number of Routing Policies employed). For more information on Routing Policies, see "Configuring SBC Routing Policy Rules" on page 493.

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 491).
 - ◆ 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 360) 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 269. 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 that are assigned Cost Groups, according to the default LCR settings configured for the assigned Routing Policy (see "Configuring SBC Routing Policy Rules" on page 493).
- **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 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 tags to represent these numbers. Thus, instead of configuring many routing rules, you need to configure only one routing rule using the tag as the source and destination number matching characteristics, and a destination for the calls. For more information on prefix tags, see Configuring Dial Plans on page 511.



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 through the Web interface. You can also configure it through ini file (IP2IPRouting) or CLI (configure voip > sbc routing ip2ip-routing).

- **To configure an IP-to-IP routing rule:**
- 1. Open the IP-to-IP Routing table (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **IP-to-IP Routing Table**).

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

Figure 25-6: IP-to-IP Routing Table - Add Row Dialog Box

3. Configure an IP-to-IP routing rule according to the parameters described in the table below.
4. Click **Add**.

Table 25-3: IP-to-IP Routing Table Parameter Descriptions

Parameter	Description
Index [IP2IPRouting_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Routing Policy sbc-routing-policy-name [IP2IPRouting_RoutingPolicyName]	Assigns an SBC Routing Policy to the rule. The SBC Routing Policy associates the rule with an SRD(s). The SBC Routing Policy also defines default LCR settings as well as the LDAP servers used if the routing rule is based on LDAP routing (and Call Setup Rules). If only one SBC Routing Policy is configured in the SBC Routing Policy table, the SBC Routing Policy is automatically assigned. If multiple SBC Routing Policies are configured, no value is assigned. For configuring SBC Routing Policies, see "Configuring SBC Routing Policy Rules" on page 493. Note: The parameter is mandatory.
Name route-name [IP2IPRouting_RouteName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no value is defined.
Rule (Matching Characteristics)	
Alternative Route Options alt-route-options [IP2IPRouting_AltRouteOptions]	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

Parameter	Description
	characteristics matches this rule. ▪ [1] Alternative 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] Alternative 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 491). 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).
Source IP Group src-ip-group-name [IP2IPRouting_SrcIPGroup pName]	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 471). The default is Any (i.e., any IP Group). Note: The selectable IP Group for the parameter depends on the assigned SBC Routing Policy (in the 'Routing Policy' parameter in this table). For more information, see "Configuring SBC Routing Policy Rules" on page 493.

Parameter	Description
Request Type request-type [IP2IPRouting_RequestType]	Defines the SIP dialog request type (SIP Method) of the incoming SIP dialog. ▪ [0] All (default) ▪ [1] INVITE ▪ [2] REGISTER ▪ [3] SUBSCRIBE ▪ [4] INVITE and REGISTER ▪ [5] INVITE and SUBSCRIBE ▪ [6] OPTIONS
Source Username Prefix src-user-name-prefix [IP2IPRouting_SrcUsernamePrefix]	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 715. The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty. Note: If you need to route calls of many different source URI user names to the same destination, you can use tags (see 'Source Tags' parameter below) instead of this parameter.
Source Host src-host [IP2IPRouting_SrcHost]	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.
Source Tags src-tags [IP2IPRouting_SrcTags]	Assigns a tag to denote source URI user names corresponding to the tag configured in the associated Dial Plan. The valid value is a string of up to 20 characters. The tag is case insensitive. To configure tags, see Configuring Dial Plans on page 516. Note: ▪ Make sure that you assign the Dial Plan in which you have configured the tag, to the related IP Group or SRD. ▪ Instead of using tags and configuring the parameter, you can use the 'Source Username Prefix' parameter to specify a specific URI source user or all source users.
Destination Username Prefix dst-user-name-prefix [IP2IPRouting_DestUsernamePrefix]	Defines the prefix of the incoming SIP dialog's destination URI (usually the Request URI) user part. You can use special notations for denoting the prefix. To denote calls without a user part in the URI, use the \$ sign. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 715. The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty. Note: If you need to route calls of many different destination URI user names to the same destination, you can use tags (see 'Destination Tags' parameter below) instead of this parameter.
Destination Host dst-host [IP2IPRouting_DestHost]	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.

Parameter	Description
Destination Tags dst-tags [IP2IPRouting_DestTags]	Assigns a tag to denote destination URI user names corresponding to the tag configured in the associated Dial Plan. The valid value is a string of up to 20 characters. The tag is case insensitive. To configure tags, see Configuring Dial Plans on page 516. Note: - Make sure that you assign the Dial Plan in which you have configured the tag, to the related IP Group or SRD. - Instead of using tags and configuring the parameter, you can use the 'Destination Username Prefix' parameter to specify a specific URI destination user or all destination users.
Message Condition message-condition-name [IP2IPRouting_MessageConditionName]	Assigns a SIP Message Condition rule to the IP-to-IP Routing rule. For configuring Message Condition rules, see "Configuring Message Condition Rules" on page 478.
Call Trigger trigger [IP2IPRouting_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 configuration setup ensures that the device uses this alternative routing rule only when RTP broken connection is detected.

Parameter	Description
ReRoute IP Group re-route-ip-group-id [IP2IPRouting_ReRouteIPGroupName]	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. For more information, see "Interworking SIP 3xx Redirect Responses" on page 447 and "Interworking SIP REFER Messages" on page 450, respectively. The parameter functions together with the 'Call Trigger' parameter (in the table). The default is Any (i.e., any IP Group). Note: The selectable IP Group for the parameter depends on the assigned SBC Routing Policy (in the 'Routing Policy' parameter in this table). For more information, see "Configuring SBC Routing Policy Rules" on page 493.
Action	
Destination Type dst-type [IP2IPRouting_DestType]	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 as defined in the 'Destination IP Group' (IP2IPRouting_DestIPGroupName) parameter. For more information on the actual address, see the 'Destination IP Group' parameter. ▪ [1] Dest Address = The SIP dialog is sent to the address configured in the following parameters: '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 536. ▪ [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. Make sure that the Routing Policy assigned to the routing rule is configured with the LDAP Server Group for defining the LDAP server(s) to query. ▪ [9] Routing Server = Device sends a request to a third-party routing

Parameter	Description
	server for an appropriate destination (next hop) for the matching call. [10] All Users = Device checks whether the Request-URI (i.e., destination user) in the incoming INVITE is registered in its' users' database, and if yes, it sends the INVITE to the address of the corresponding contact specified in the database. If the Request-URI is not registered, the call is rejected.
Destination IP Group dst-ip-group-name [IP2IPRouting_DestIPGroupUpName]	Defines the IP Group to where you want to route the call. The actual destination of the SIP dialog message depends on the IP Group type (as defined in the 'Type' parameter): - Server-type IP Group: The SIP dialog is sent to the IP address configured for the Proxy Set that is associated with the IP Group. - User-type IP Group: The device checks if the SIP dialog is from a registered user, by searching for a match between the Request-URI of the received SIP dialog and an AOR registration record in the device's database. If found, the device sends the SIP dialog to the IP address specified in the database for the registered contact. By default, no value is defined (None). Notes: - The parameter is only relevant if the parameter 'Destination Type' is set to IP Group. - The selectable IP Group for the parameter depends on the assigned SBC Routing Policy (in the 'Routing Policy' parameter in this table). For more information, see "Configuring SBC Routing Policy Rules" on page 493.
Destination SIP Interface dst-srd-id [IP2IPRouting_DestSIPInterfaceName]	Defines the destination SIP Interface to where the call is sent. By default, no value is defined (None). For configuring SIP Interfaces, see "Configuring SIP Interfaces" on page 342. Notes: - The parameter is applicable only if the 'Destination Type' parameter is configured to any value other than IP Group. If the 'Destination Type' parameter is configured to IP Group, the following SIP Interface is used: ✓ Server-type IP Groups: SIP Interface that is assigned to the Proxy Set associated with the IP Group. ✓ User-type IP Groups: SIP Interface is determined during user registration with the device. - For multi-tenancy, if the assigned SBC Routing Policy is not shared (i.e., the Routing Policy is associated with an Isolated SRD), the SIP Interface must be one that is associated with the Routing Policy or with a shared Routing Policy (i.e., the Routing Policy is associated with one or more Shared SRDs). If the Routing Policy is shared, the SIP Interface can be one that is associated with any SRD or Routing Policy (but it's recommended that it belong to the same SRD/Routing Policy or to shared SRD/Routing Policy to avoid "bleeding").
Destination Address dst-address [IP2IPRouting_DestAddress]	Defines the destination address to where the call is sent. The address 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) the 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

Parameter	Description
	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 (IP address or FQDN). By default, no value is defined. Notes: - The 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 153). - To terminate SIP OPTIONS messages at the device (i.e., to handle them locally), set the parameter to "internal".
Destination Port dst-port [IP2IPRouting_DestPort]	Defines the destination port to where the call is sent.
Destination Transport Type dst-transport-type [IP2IPRouting_DestTransportType]	Defines the transport layer type for sending the call: [-1] = (Default) Not configured - the transport type is determined by the SIPTransportType global parameter. [0] UDP [1] TCP [2] TLS
Call Setup Rules Set ID 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 292.
Group Policy 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 cost-group [IP2IPRouting_CostGroup]	Assigns a Cost Group to the routing rule for determining the cost of the call. By default, no value is defined (None). For configuring Cost Groups, see "Configuring Cost Groups" on page 271. Notes: - To implement LCR and its Cost Groups, you must enable LCR for the Routing Policy assigned to the routing rule (see "Configuring SBC Routing Policy Rules" on page 493). If LCR is disabled, the device ignores the parameter. - The Routing Policy also determines whether matched routing rules that are not assigned Cost Groups are considered as a higher or lower cost route compared to matching routing rules that are assigned Cost Groups. For example, if the 'Default Call Cost' parameter in the Routing Policy is configured to Lowest Cost, even if the device locates matching routing rules that are assigned Cost Groups, the first-matched routing rule without an assigned Cost Group is considered as the lowest cost route and thus, chosen as the preferred route.

25.4 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 360), 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 480).

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, that are issued by the device itself:

- **805 IP Profile Call Limit:** The device generates this response code when Call Admission Control (CAC) limits (such as maximum concurrent calls) are exceeded for an IP Group (or SRD). The CAC rules are configured in the Admission Control table (see "Configuring Admission Control" on page 463). 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 QoS 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 through the Web interface. You can also configure it through ini file (SBCAlternativeRoutingReasons) or CLI (configure voip > sbc routing sbc-alt-routing-reasons).

➤ **To configure SIP reason codes for alternative IP routing:**

1. Open the SBC Alternative Routing Reasons table (**Configuration** tab > **VoIP** menu > **SBC > Routing SBC > Alternative Routing Reasons**).
2. Click **Add**; the following dialog box appears:

Figure 25-7: SBC Alternative Routing Reasons Table - Add Row Dialog Box

3. Configure a SIP response code for alternative routing according to the parameters described in the table below.
4. Click **Add**.

Table 25-4: SBC Alternative Routing Reasons Table Parameter Descriptions

Parameter	Description
Index [SBCAlternativeRoutingReasons_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Release Cause rel-cause [SBCAlternativeRoutingReasons_ReleaseCause]	Defines a SIP response code for triggering the device's alternative routing mechanism.

25.5 Configuring SBC Routing Policy Rules

The SBC Routing Policy table lets you configure up to 250 SBC Routing Policy rules. A Routing Policy determines the routing and manipulation (inbound and outbound) rules per SRD in a multiple SRD configuration topology. The Routing Policy also configures the following:

- Enables Least Cost Routing (LCR), and configures default call cost (highest or lowest) and average call duration for routing rules that are not assigned LCR Cost Groups. The default call cost determines whether matched routing rules that are not assigned Cost Groups are considered as a higher or lower cost route compared to other matching routing rules that are assigned Cost Groups. If you disable LCR, the device ignores the Cost Groups assigned to the routing rules in the IP-to-IP Routing table.
- Assigns LDAP servers (LDAP Server Group) for LDAP-based routing. IP-to-IP routing rules configured for LDAP or CSR (Call Setup Rules) queries use the LDAP server(s) that is assigned to the routing rule's associated Routing Policy. You can configure a Routing Policy per SRD or alternatively, configure a single Routing Policy that is shared between all SRDs.

The implementation of Routing Policies is intended for the following deployments **only**:

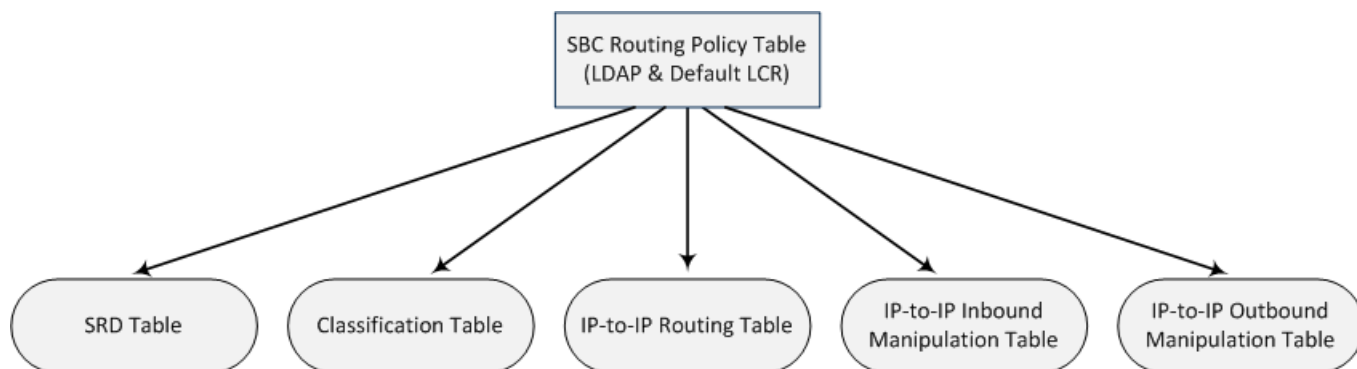
- Deployments requiring LCR and/or LDAP-based routing.
- Multi-tenant deployments that require multiple, logical routing tables where each tenant has its own dedicated ("separated") routing (and manipulation) table. In such scenarios, each SRD (tenant) is configured as an Isolated SRD and assigned its own unique Routing Policy, implementing an almost isolated, non-bleeding routing configuration topology.

For all other deployment scenarios, the Routing Policy is irrelevant and the handling of the configuration entity is not required as a default Routing Policy ("Default_SBCRoutingPolicy" at Index 0) is provided. When only one Routing Policy is required, the device automatically associates the default Routing Policy with newly added configuration entities that can be associated with the Routing Policy (as mentioned later in this section, except for Classification rules). This facilitates configuration, eliminating the need to handle the Routing Policy configuration entity (except if you need to enable LCR and/or assign an LDAP server to the Routing Policy). In such a setup, where only one Routing Policy is used, single routing and manipulation tables are employed for all SRDs.



Note: If possible, it is recommended to use only **one** Routing Policy for all SRDs (tenants), unless deployment requires otherwise (i.e., a dedicated Routing Policy per SRD).

Once configured, you need to associate the Routing Policy with an SRD(s) in the SRD table. To determine the routing and manipulation rules for the SRD, you need to assign the Routing Policy to routing and manipulation rules. The figure below shows the configuration entities to which Routing Policies can be assigned:



Typically, assigning a Routing Policy to a Classification rule is not required, as when an incoming call is classified it uses the Routing Policy associated with the SRD to which it belongs. However, if a Routing Policy is assigned to a Classification rule, it overrides the Routing Policy assigned to the SRD. The option to assign Routing Policies to Classification rules is useful in deployments requiring different routing and manipulation rules for specific calls pertaining to the **same** SRD. In such scenarios, you need to configure multiple Classification rules for the same SRD, where for some rules no Routing Policy is assigned (i.e., the SRD's assigned Routing Policy is used) while for others a different Routing Policy is specified to override the SRD's assigned Routing Policy.

In multi-tenant environments employing multiple SRDs and Routing Policies, the IP Groups that can be used in routing rules (in the IP-to-IP Routing table) are as follows:

- If the Routing Policy is assigned to only one SRD and the SRD is an Isolated SRD, the routing rules of the Routing Policy can be configured with IP Groups belonging to the Isolated SRD and IP Groups belonging to all Shared SRDs.
- If the Routing Policy is assigned to a Shared SRD, the routing rules of the Routing Policy can be configured with any IP Group (i.e., belonging to Shared and Isolated SRDs). In effect, the Routing Policy can include routing rules for call routing between Isolated SRDs.
- If the Routing Policy is assigned to multiple SRDs (Shared and/or Isolated), the routing rules of the Routing Policy can be configured with IP Groups belonging to all Shared SRDs as well as IP Groups belonging to Isolated SRDs that are assigned the Routing Policy.

To facilitate the configuration of routing rules in the IP-to-IP Routing table through the Web interface, only the permitted IP Groups (according to the above) are displayed as optional values.

The general flow for processing the call for multi-tenant deployments and Routing Policies is as follows:

1. Using the Classification table, the device classifies the incoming call to an IP Group, based on the SIP Interface on which the call is received. Based on the SIP Interface, the device associates the call to the SRD that is assigned to the SIP Interface.
2. Once the call has been successfully classified to an IP Group, the Routing Policy assigned to the associated SRD is used. However, if a Routing Policy is configured in the Classification table, it overrides the Routing Policy assigned to the SRD.
3. The regular manipulation (inbound and outbound) and routing processes are done according to the associated Routing Policy.

**Notes:**

- The Classification table is used only if classification by registered user in the device's users registration database or by Proxy Set fails.
- If the device receives incoming calls (e.g., INVITE) from users that have already been classified and registered in the device's registration database, the device ignores the Classification table and uses the Routing Policy that was determined for the user during the initial classification process.

The following procedure describes how to configure SBC Routing Policy rules through the Web interface. You can also configure it through ini file (SBCRoutingPolicy) or CLI (configure voip > sbc routing sbc-routing-policy).

➤ **To configure an SBC Routing Policy rule:**

1. Open the SBC Routing Policy table (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **SBC Routing Policy**).
2. Click **Add**; the following dialog box appears:

Figure 25-8: SBC Routing Policy Table - Add Row Dialog Box

3. Configure the SBC Routing Policy rule according to the parameters described in the table below.
4. Click **Add**.

Table 25-5: SBC Routing Policy Table Parameter Descriptions

Parameter	Description
Index	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [SBCRoutingPolicy_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no name is defined. If you don't configure a name, the device automatically assigns a name in the following format: "SBCRoutingPolicy_<Index>", for example, "SBCRoutingPolicy_2". Note: Each row must be configured with a unique name.
LDAP Servers Group Name	Assigns an LDAP Server Group to the Routing Policy. Routing rules in the IP-to-IP Routing table that are associated with the Routing Policy and that are configured with LDAP and/or Call Setup Rules, use the LDAP

Parameter	Description
ldap-srv-group-name [SBCRoutingPolicy_LdapServersGroupName]	server(s) configured for this LDAP Server Group. By default, no value is defined (None). For more information on LDAP Server Groups, see "Configuring LDAP Server Groups" on page 243. Note: The default SBC Routing Policy is assigned the default LDAP Server Group ("DefaultCTRLServersGroup").
LCR Feature lcr-enable [SBCRoutingPolicy_LCR Enable]	Enables the Least Cost Routing (LCR) feature for the Routing Policy. ▪ [0] Disable (default) ▪ [1] Enable For more information on LCR, see "Least Cost Routing" on page 269.
Default Call Cost lcr-default-cost [SBCRoutingPolicy_LCR DefaultCost]	Defines whether routing rules in the IP-to-IP Routing table that are not assigned a 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 = (Default) The device considers a matched routing rule (belonging to the Routing Policy) that is not assigned a Cost Group as the lowest cost route. Therefore, it uses the routing rule. ▪ [1] Highest Cost = The device considers a matched routing rule (belonging to the Routing Policy) that is not assigned a Cost Group as the highest cost route. Therefore, it is only used if the other matched routing rules that are assigned Cost Groups are unavailable. Note: If multiple matched routing rules without an assigned Cost Group exist, the device selects the first matched rule in the table.
LCR Call Duration lcr-call-length [SBCRoutingPolicy_LCR AverageCallLength]	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 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, "Weekend B" carries the lower cost.

26 SBC Manipulations

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

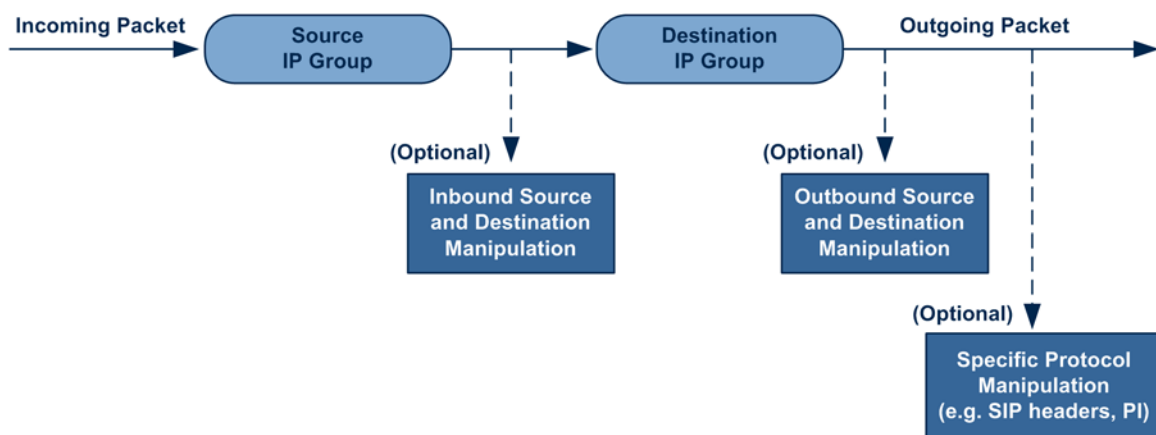


Note: For additional manipulation features, see the following:

- "Configuring SIP Message Policy Rules".
- "Configuring SIP Message Manipulation" on page 377.

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 26-1: 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=0llan;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=0Wan;paramer1=abe
To: <sip: 9721000@ ITSP;user=phone>
Call-ID: USEVWWAN@212.179.1.12
CSeq: 38 INVITE
Contact: <sip:7000@212.179.1.12>
Supported: em,100rel,timer,replaces
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER
User-Agent: Sip Message Generator V1.0.0.5
Content-Type: application/sdp
Content-Length: 155
v=0
o=SMG 5 9 IN IP4 212.179.1.11
s=Phone-Call
c=IN IP4 212.179.1.11
t=0 0
m=audio 8000 RTP/AVP 8
a=rtpmap:8 pcma/8000
a=sendrecv
a=ptime:20
```

The SIP message manipulations in the example above (contributing to typical topology hiding) are as follows:

■ Inbound source SIP URI user name from "7000" to "97000":

```
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=0LLAN;paramer1=abe
```

to

```
From:
<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=0Wan;paramer1=abe
```

■ Source IP Group name (i.e., SIP URI host name) from "10.2.2.6" to "IP_PBX":

```
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=0LLAN;paramer1=abe
```

to

```
From:
<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=0Wan;paramer1=abe
```

- Inbound destination SIP URI user name from "1000" to 9721000":

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
To: <sip:1000@10.2.2.3;user=phone>
to
INVITE sip:9721000@ITSP;user=phone;x=y;z=a SIP/2.0
To: <sip:9721000@ITSP;user=phone>
```
- Destination IP Group name (SIP URI host name) from "10.2.2.3" to "ITSP":

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
To: <sip:1000@10.2.2.3;user=phone>
to
INVITE sip:9721000@ITSP;user=phone;x=y;z=a SIP/2.0
To: <sip:9721000@ITSP;user=phone>
```

26.1 Configuring IP-to-IP Inbound Manipulations

The IP to IP Inbound Manipulation table lets you configure up to 1,250 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)

The configuration of an IP-to-IP Inbound Manipulation rule includes two areas:

- **Rule:** Defines the matching characteristics of an incoming SIP dialog (e.g., source host name).
- **Action:** Defines the operation that must be done if the incoming call matches the characteristics of the rule. In other words, the device manipulates the source or destination SIP URI user part of the SIP dialog (e.g., removes a user-defined number of characters from the left of the SIP URI user part).

To configure and apply an IP-to-IP Inbound Manipulation rule, the rule must be associated with a Routing Policy. The Routing Policy associates the rule with an SRD(s). Therefore, the Routing Policy lets you configure manipulation rules for calls belonging to specific SRD(s). However, as multiple Routing Policies are relevant only for multi-tenant deployments (if needed), for most deployments, only a single Routing Policy is required. As the device provides a default Routing Policy ("Default_SBCRoutingPolicy"), when only one Routing Policy is required, the device automatically assigns the default Routing Policy to the routing rule. If you are implementing LDAP-based routing (with or without Call Setup Rules) and/or Least Cost Routing (LCR), you need to configure these settings for the Routing Policy (regardless of the number of Routing Policies employed). For more information on Routing Policies, see "Configuring SBC Routing Policy Rules" on page 493.



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 348).

The following procedure describes how to configure IP-to-IP Inbound Manipulation rules through the Web interface. You can also configure it through ini file (IPInboundManipulation) or CLI (configure voip > sbc manipulations ip-inbound-manipulation).

➤ **To configure an IP-to-IP Inbound Manipulation rule:**

1. Open the IP to IP Inbound Manipulation table (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP-to-IP Inbound**).
2. Click **Add**; the following dialog box appears:

Figure 26-2: IP to IP Inbound Manipulation Table - Add Row Dialog Box

3. Configure the IP-to-IP inbound manipulation rule according to the parameters described in the table below.
4. Click **Add**.

Table 26-1: IP to IP Inbound Manipulation Table Parameter Descriptions

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

Parameter	Description
Routing Policy routing-policy-name [IPInboundManipulation_RoutingPolicyName]	Assigns an SBC Routing Policy to the rule. The SBC Routing Policy associates the rule with an SRD(s). The SBC Routing Policy also defines default LCR settings as well as the LDAP servers if the routing rule is based on LDAP routing (and Call Setup Rules). If only one SBC Routing Policy is configured in the SBC Routing Policy table, the SBC Routing Policy is automatically assigned. If multiple SBC Routing Policies are configured, no value is assigned. For configuring SBC Routing Policies, see "Configuring SBC Routing Policy Rules" on page 493. Note: The parameter is mandatory.
Manipulation Name manipulation-name [IPInboundManipulation_ManipulationName]	Defines an arbitrary name to easily identify the manipulation rule. The valid value is a string of up to 40 characters. By default, no value is defined.
Matching Characteristics - Rule	
Additional Manipulation CLI: is-additional-manipulation [IPInboundManipulation_IsAdditionalManipulation]	Determines whether additional SIP URI user part manipulation is done for the table entry rule listed directly above it. ▪ [0] No = (Default) Regular manipulation rule (not done in addition to the rule above it). ▪ [1] Yes = If the above row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule. Note: Additional manipulation can only be done on a different SIP URI, source or destination, to the rule configured in the row above as configured by the 'Manipulated URI' parameter (see below).
Manipulation Purpose CLI: purpose [IPInboundManipulation_ManipulationPurpose]	Defines the purpose of the manipulation: ▪ [0] Normal = (Default) Inbound manipulations affect the routing input and source and/or destination number. ▪ [1] Routing input only = Inbound manipulations affect the routing input only, retaining the original source and destination number. ▪ [2] Shared Line = Used for the Shared-Line Appearance feature. This manipulation is for registration requests to change the destination number of the secondary extension numbers to the primary extension. For more information, see "BroadSoft's Shared Phone Line Call Appearance for SBC Survivability" on page 535.
Source IP Group CLI: src-ip-group-name [IPInboundManipulation_SrcIpGroupName]	Defines the IP Group from where the incoming INVITE is received. The default is Any (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 715.

Parameter	Description
Source Host CLI: src-host [IPInboundManipulation_SrcHost]	Defines the source SIP URI host name - full name (usually in the From header). The default is the asterisk (*) symbol (i.e., any host name).
Destination Username Prefix CLI: dst-user-name-prefix [IPInboundManipulation_DestUsernamePrefix]	Defines the prefix of the destination SIP URI user name, 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 715.
Destination Host CLI: dst-host [IPInboundManipulation_DestHost]	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 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".

Parameter	Description
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".

26.2 Configuring IP-to-IP Outbound Manipulations

The IP to IP Outbound Manipulation table lets you configure up to 1,250 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).

The configuration of an IP-to-IP Outbound Manipulation rule includes two areas:

- **Rule:** Defines the matching characteristics of an incoming SIP dialog (e.g., source host name). As the device performs outbound manipulations only after the routing process, destination IP Groups can also be used as matching characteristics.
- **Action:** Defines the operation that must be done if the incoming call matches the characteristics of the rule. In other words, the device manipulates the source or destination SIP URI user part or calling name of the SIP dialog (e.g., removes a 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 through the Web interface. You can also configure it through ini file (IPOutboundManipulation) or CLI (configure voip > sbc manipulations ip-outbound-manipulation).

➤ **To configure IP-to-IP outbound manipulation rules:**

1. Open the IP to IP Outbound Manipulation table (**Configuration** tab > **VoIP** menu > **SBC > Manipulations SBC > IP-to-IP Outbound**).

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

Figure 26-3: IP to IP Outbound Manipulation Table- Add Row Dialog Box

- Configure an IP-to-IP outbound manipulation rule according to the parameters described in the table below.
- Click **Add**.

Table 26-2: IP to IP Outbound Manipulation Table Parameter Description

Parameter	Description
Index [IPOutboundManipulation_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Routing Policy routing-policy-name [IPOutboundManipulation_RoutingPolicyName]	Assigns an SBC Routing Policy to the rule. The SBC Routing Policy associates the rule with an SRD(s). The SBC Routing Policy also defines default LCR settings as well as the LDAP servers if the routing rule is based on LDAP routing (and Call Setup Rules). If only one SBC Routing Policy is configured in the SBC Routing Policy table, the SBC Routing Policy is automatically assigned. If multiple SBC Routing Policies are configured, no value is assigned. For configuring SBC Routing Policies, see "Configuring SBC Routing Policy Rules" on page 493. Note: The parameter is mandatory.
Manipulation Name manipulation-name [IPOutboundManipulation_ManipulationName]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 40 characters. By default, no value is defined.

Parameter	Description
Rule (Matching Characteristics)	
Additional Manipulation 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 src-ip-group-name [IPOutboundManipulation_SrcIPGroupName]	Defines the IP Group from where the INVITE is received. The default value is Any (i.e., any IP Group).
Destination IP Group dst-ip-group-name [IPOutboundManipulation_DestIPGroupName]	Defines the IP Group to where the INVITE is to be sent. The default value is Any (i.e., any IP Group).
Source Username Prefix 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 715. Note: If you need to manipulate calls of many different source URI user names, you can use tags (see 'Source Tags' parameter below) instead of this parameter.
Source Host 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).
Source Tags src-tags [IPOutboundManipulation_SrcTags]	Assigns a prefix tag to denote source URI user names corresponding to the tag configured in the associated Dial Plan. The valid value is a string of up to 20 characters. The tag is case insensitive. To configure prefix tags, see Configuring Dial Plans on page 511. Note: - Make sure that you assign the Dial Plan in which you have configured the prefix tag, to the related IP Group or SRD. - Instead of using tags and configuring the parameter, you can use the 'Source Username Prefix' parameter to specify a specific URI source user or all source users.
Destination Username Prefix dst-user-name-prefix [IPOutboundManipulation]	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

Parameter	Description
_DestUsernamePrefix]	Manipulation" on page 715. Note: If you need to manipulate calls of many different destination URI user names, you can use tags (see 'Destination Tags' parameter below) instead of this parameter.
Destination Host 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).
Destination Tags dest - tags [IPOutboundManipulation _DestTags]	Assigns a prefix tag to denote destination URI user names corresponding to the tag configured in the associated Dial Plan. The valid value is a string of up to 20 characters. The tag is case insensitive. To configure prefix tags, see Configuring Dial Plans on page 511. Note: - Make sure that you assign the Dial Plan in which you have configured the prefix tag, to the related IP Group or SRD. - Instead of using tags and configuring the parameter, you can use the 'Destination Username Prefix' parameter to specify a specific URI destination user or all destinations users.
Calling Name Prefix 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 message-condition - name [IPOutboundManipulation _MessageConditionName]	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 478.
Request Type request - type [IPOutboundManipulation _RequestType]	Defines the SIP request type to which the manipulation rule is applied. [0] All = (Default) all SIP messages. [1] INVITE = All SIP messages except REGISTER and SUBSCRIBE. [2] REGISTER = Only SIP REGISTER messages. [3] SUBSCRIBE = Only SIP SUBSCRIBE messages. [4] INVITE and REGISTER = All SIP messages except SUBSCRIBE. [5] INVITE and SUBSCRIBE = All SIP messages except REGISTER.
ReRoute IP Group re-route-ip-group - name [IPOutboundManipulation _ReRouteIPGroupName]	Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. The 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 Any (i.e., any IP Group). Notes: - The 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

Parameter	Description
	page 447 and "Interworking SIP REFER Messages" on page 450, respectively.
Call Trigger 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.
Action	
Manipulated Item 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 remove-from-left [IPOutboundManipulation_RemoveFromLeft]	Defines the number of digits to remove from the left of the manipulated item prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".
Remove From Right 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 leave-from-right [IPOutboundManipulation_LeaveFromRight]	Defines the number of digits to keep from the right of the manipulated item.
Prefix to Add 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 the parameter to a string of up to 49 characters. If you set the 'Manipulated Item' parameter to Calling Name , you can configure the parameter to a string of up to 36 characters.
Suffix to Add 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 the parameter to a string of up to 49 characters. If you set the 'Manipulated Item' parameter to Calling Name , you can configure the parameter to a string of up to 36 characters.
Privacy Restriction Mode privacy-restriction-mode	Defines user privacy handling (i.e., restricting source user identity in outgoing SIP dialogs). ▪ [0] Transparent = (Default) No intervention in SIP privacy.

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

This page is intentionally left blank.

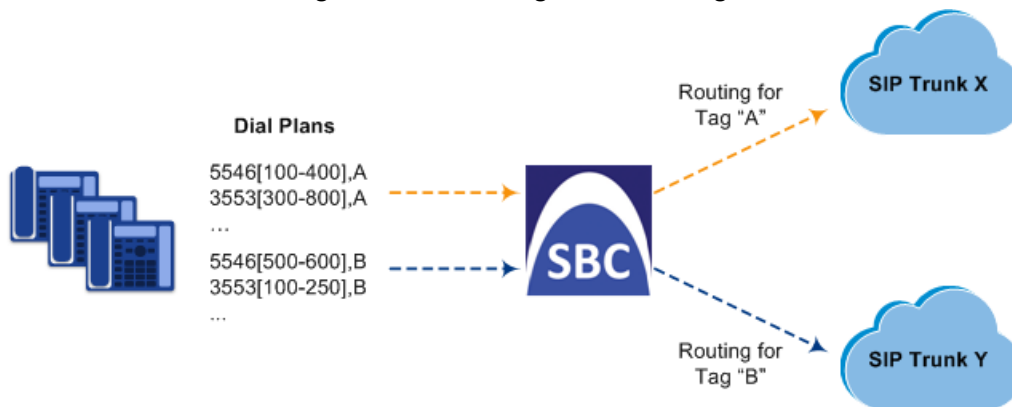
27 Configuring Dial Plans

Dial Plans let you categorize users (source and/or destination) according to source and/or destination numbers of the incoming SIP dialog-initiating requests. The device categorizes users by searching in the Dial Plan for rules that match these numbers according to prefix, suffix, and/or whole number. The categorization result in the Dial Plan is a *tag* corresponding to the matched rules. You can then use the tags to represent these users (source and/or destination users) as matching characteristics (source and/or destination tags) for the following:

- IP-to-IP Routing rules (see 'Using Dial Plan Tags for IP-to-IP Routing' on page 590)
- IP-to-IP Outbound Manipulation rules ('Using Dial Plan Tags for Outbound Manipulation' on page 520)

The figure below shows a conceptual example of routing based on tags, where users categorized as tag "A" are routed to SIP Trunk "X" and those categorized as tag "B" are routed to SIP Trunk "Y":

Figure 27-1: Routing based on Tags



Note:

- User categorization by Dial Plan is done only after the device's Classification and Inbound Manipulation processes, and before the routing process.
- Once the device successfully categorizes an incoming call by Dial Plan, it not only uses the resultant tag in the immediate routing or manipulation process, but also in subsequent routing and manipulation processes that may occur, for example, due to alternative routing or local handling of call transfer and call forwarding (SIP 3xx\REFER).
- For manipulation, tags are applicable only to outbound manipulation.

You can assign a Dial Plan to an IP Group or SRD. After Classification and Inbound Manipulation, the device checks if a Dial Plan is associated with the incoming call. It first checks the source IP Group and if no Dial Plan is assigned, it checks the SRD. If a Dial Plan is assigned to the IP Group or SRD, the device first searches the Dial Plan for a dial plan rule that matches the source number and then it searches the Dial Plan for a rule that matches the destination number. If matching dial plan rules are found, the tags configured for these rules are used in the routing and/or manipulation processes as source and/or destination tags.

The Dial Plan itself is a set of dial plan rules having the following attributes:

- **Prefix:** The prefix is matched against the source and/or destination number of the incoming SIP dialog-initiating request.

- **Tag:** The tag corresponds to the matched prefix of the source and/or destination number and is the categorization result.

You can use various syntax notations for configuring the prefix numbers in dial plan rules. You can configure the prefix as a complete number (all digits) or as a partial number using some digits and various syntax notations (patterns) to allow the device to match a dial plan rule for similar source and/or destination numbers. For more information, see the description of the 'Prefix' parameter (DialPlanRule_Prefix) described later in this section.

The device employs a "best-match" method instead of a "first-match" method to match the source/destination numbers to prefixes configured in the dial plan. The matching order is done digit-by-digit and from left to right. The numbers are first matched to the rule configured with the most constrained (specific) character set. Most constrained implies that the dial plan pattern that has the fewest possible matches for a digit is matched first. For example, if one rule contains the "x" wildcard character, which has ten possible matches (i.e., 0-9) and another rule a specific digit (e.g., 4), the rule with the specific digit is selected as the matching rule.

The best match priority is listed below in chronological order:

- Specific character (prefix)
- Number range
- "x" wildcard, which denotes any digit (0-9)
- Suffix, where the longest digits is first matched. For example, ([001-999]) takes precedence over ([01-99]) which takes precedence over ([1-9]).
- . (dot), which denotes any character

The following examples show how the best-matching method is done. Each example has two dial plan rules which are shown listed in chronological order as they would be configured in the table.

- For incoming calls with prefix number "5234", the rule with tag B is chosen (more specific for digit "4"):

```
523x, A
5234, B
```

- For incoming calls with prefix number "5234", the rule with tag B is chosen (more specific for digit "4"):

```
523x, A
523[1-9], B
```

- For incoming calls with prefix number "53211111", the rule with tag B is chosen (more specific for fourth digit):

```
532[1-9]1111, A
5321, B
```

- For incoming calls with prefix number "53124", the rule with tag B is chosen (more specific for digit "1"):

```
53([2-4]), A
531(4), B
```

- For incoming calls with prefix number "321444", the rule with tag A is chosen and for incoming calls with prefix number "32144", the rule with tag B is chosen:

```
321xxx, A
321, B
```

- For incoming calls with prefix number "5324", the rule with tag B is chosen (prefix is more specific for digit "4"):

```
532[1-9], A
532[2-4], B
```

- For incoming calls with prefix number "53124", the rule with tag C is chosen (longest suffix - C has three digits, B two digits and A one digit):

```
53([2-4]), A
53([01-99]), B
53([001-999]), C
```

- For incoming calls with prefix number "53124", the rule with tag B is chosen (suffix is more specific for digit "4"):

```
53([2-4]), A
53(4), B
```

Dial Plans are configured using two tables with parent-child type relationship:

- Parent table: Dial Plan table, which defines the name of the Dial Plan. You can configure up to five Dial Plans.
- Child table: Dial Plan Rule table, which defines the actual dial plans (rules) per Dial Plan. You can configure up to 2,000 dial plan rules in total (where all can be configured for one Dial Plan or configured between different Dial Plans).

The following procedure describes how to configure Dial Plans through the Web interface. You can also configure it through other management platforms:

- **Dial Plan table:** *ini* file (DialPlans) or CLI (configure voip > sbc dial-plan)
- **Dial Plan Rule table:** *ini* file (DialPlanRule) or CLI (configure voip > sbc dial-plan-rule)

➤ **To configure Dial Plans:**

1. Open the Dial Plan table (**Configuration** tab > **VoIP** menu > **SBC** > **Dial Plan**).
2. Click **New**; the following dialog box appears:

Figure 27-2: Dial Plan Table - Add Row Dialog Box

3. Configure a Dial Plan name according to the parameters described in the table below.
4. Click **Apply**.

Dial Plan Table Parameter Descriptions

Parameter	Description
Index [DialPlans_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [DialPlans_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 15 characters. Note: Each row must be configured with a unique name.

5. In the Dial Plan table, select the row for which you want to configure dial plan rules, and then click the **Dial Plan Rule** link located below the table; the Dial Plan Rule table appears.

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

Figure 27-3: Dial Plan Rule Table

7. Configure a dial plan rule according to the parameters described in the table below.
8. Click **New**, and then save ("burn") your settings to flash memory.

Dial Plan Rule Table Parameter Descriptions

Parameter	Description
Index index [DialPlanRule_RuleIndex]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Name name [DialPlanRule_Name]	Defines an arbitrary name to easily identify the row. The valid value is a string of up to 15 characters.
Prefix prefix [DialPlanRule_Prefix]	Defines the prefix number of the source or destination number. The valid value is up to 50 characters. The following syntax can be used: ▪ 0-9: Specific digit. ▪ x: Wildcard denoting any digit from 0 through 9. ▪ z: Denotes a number from 1 through 9. ▪ n: Denotes a number from 2 through 9. ▪ a-z: Lower-case letter. ▪ A-Z: Upper-case letter. ▪ *: (Asterisk symbol) If it is the only character in the rule, it denotes any number. To denote the asterisk "*" symbol itself, precede it with the escape "\\" character (see below). ▪ \: (Backslash escape character) When it prefixes a wildcard character (*, z, n, and x), the character itself is used and not the meta-meaning. For example, "\\x" denotes the character "x", while "x" is the wildcard denoting any digits from 0-9. ▪ #: (Pound or hash symbol) When used at the end of the prefix it denotes the end of the number. For example, "54324#" represents a 5-digit number that starts with the digits 54324. ▪ .: (Period) Denotes any letter or digit. ▪ [n-m], (n-m), or ([n1-m1,n2-m2,a,b,c,n3-m3]): Represents a mixed notation of single numbers and multiple ranges. To represent the prefix, the notation is enclosed by square brackets [...]; to represent the suffix, the notation is enclosed by square brackets which are enclosed by parenthesis ([...]). For example, to denote numbers 123 through 130, 455, 766,

Parameter	Description
	and 780 through 790: ✓ Prefix: [123-130,455,766,780-790] ✓ Suffix: ([123-130,455,766,780-790]) Note: The ranges and the single numbers in the syntax must have the same amount of digits. For example, each number range and single number in the example above consists of three digits.
Tag tag [DialPlanRule_Tag]	Defines a tag. The valid value is up to 16 characters. The tag is case insensitive.

27.1 Importing and Exporting Dial Plans

You can import/export Dial Plans from/to a remote server in comma-separated value (CSV) file format through the CLI:

■ Export:

- To export a specific Dial Plan from the device to a file:

```
(config-voip)# sbc dial-plan-rule export-csv-to <Dial Plan name or index> <URL to CSV file>
```

Example:

```
# sbc dial-plan-rule export-csv-to 0
http://10.8.8.20/upload/index_0_Dial_Plans.csv
```
- To export all Dial Plans from the device to a file:

```
(config-voip)# sbc dial-plan-rule export-csv-to all <URL to CSV file>
```

■ Import:

- To import dial plan rules from a file to a specific Dial Plan on the device:

```
(config-voip)# sbc dial-plan-rule import-csv-from <Dial Plan name or index> <URL path to CSV file>
```

The rules of the imported file replace all existing rules of the corresponding Dial Plan on the device. The Dial Plan name (or index) must exist on the device; otherwise, the Dial Plan is not imported.

Example:

```
# sbc dial-plan-rule import-csv-from 0
http://10.8.8.20/upload/Dial_Plan_1_Rules.csv
```
- To overwrite all Dial Plans on the device by importing all dial plan rules from a file:

```
(config-voip)# sbc dial-plan-rule import-csv-from all <URL to CSV file>
```

The rules of the Dial Plans in the imported file replace all existing rules of the corresponding Dial Plans on the device. For Dial Plans on the device that are not listed in the imported file, the device deletes all of their rules. For example, if the imported file contains Dial Plan 1 and the device is currently configured with Dial Plans 1 and 2, the rules of Dial Plan 1 in the imported file replace the rules of Dial Plan 1 on the device, and the rules of Dial Plan 2 on the device are deleted (the Dial Plan itself remains). The Dial Plan names in the imported file must be identical to the existing Dial Plan names on the device; otherwise, the specific Dial Plan is not imported.

For creating Dial Plans in a CSV file for import, see 'Creating Dial Plan Files for Import' on page 516.

27.2 Creating Dial Plan Files

You can configure Dial Plans in an external file (*.csv) and then import them into the device, as described in 'Importing and Exporting Dial Plans' on page 515. You can create the file using any text-based editor such as Notepad or Microsoft Excel. The file must be saved with the *.csv file name extension.

To configure Dial Plans in a file, use the following syntax:

```
<Dial Plan>,<Rule>,<Prefix>,<Tag>
```

Where:

- *Dial Plan* is the name of the Dial Plan.
- *Rule* is the name of the dial plan rule.
- *Prefix* is the source or destination number prefix
- *Tag* is the result of the user categorization and can be used as matching characteristics for routing and outbound manipulation

For example:

```
DialPlanName, Name, Prefix, Tag
PLAN1, rule_100, 5511361xx, A
PLAN1, rule_101, 551136184[4000-9999]#, B
MyDialPlan, My_rule_200, 5511361840000#, itsp_1
MyDialPlan, My_rule_201, 66666#, itsp_2
```

27.3 Using Dial Plan Tags for IP-to-IP Routing

For deployments requiring hundreds of routing rules (which may exceed the maximum number of rules that can be configured in the IP-to-IP Routing table), you can employ tags to represent the many different calling (source URI user names) and called (destination URI user names) prefix numbers in your routing rules. Tags are typically implemented when you have users of many different called and/or calling numbers that need to be routed to the same destination (e.g., IP Group or IP address). In such a scenario, instead of configuring many routing rules to match all the required prefix numbers, you need only to configure a single routing rule using the tag to represent all the possible prefix numbers.

An example scenario where employing tags could be useful is in deployments where the device needs to service calls in a geographical area that consists of hundreds of local area codes, where each area code is serviced by one of two SIP Trunks in the network. In such a deployment, instead of configuring hundreds of routing rules to represent each local area code, you can simply configure two routing rules where each is assigned a unique tag representing a group of local area codes and the destination IP Group associated with the SIP Trunk servicing them.

**Note:**

- Source and destination tags can be used in the same routing rule.
- The same tag can be used for source and destination tags in the same routing rule.

The following procedure describes how to configure IP-to-IP routing based on tags.

➤ **To configure IP-to-IP routing based on tags:**

1. In the Dial Plan table configure a Dial Plan (see 'Configuring Dial Plans' on page 511).
2. In the IP Group or SRD associated with the calls for which you want to use tag-based routing, assign the Dial Plan that you configured in Step 1.
 - IP Group table: 'Dial Plan' parameter (IPGroup_SBCDialPlanName) - see Configuring IP Groups
 - SRD table: 'Dial Plan' parameter (SRD_SBCDialPlanName) - see Configuring SRDs
3. In the IP-to-IP Routing table (see Configuring SBC IP-to-IP Routing Rules), configure a routing rule with the required destination and whose matching characteristics include the tag(s) that you configured in your Dial Plan in Step 1. The tags are assigned under the **Rule** tab using the following parameters:
 - 'Source Tags' parameter (IP2IPRouting_SrcTags): tag denoting the calling user
 - 'Destination Tags' parameter (IP2IPRouting_DestTags): tag denoting the called user

An example of a routing rule using a destination tag "LOC" is shown below:

Figure 27-4: Assigning Tag to Routing Rule

The screenshot shows the 'Add Row' dialog box for configuring a routing rule. The 'Rule' tab is selected. The 'Destination Tags' field is highlighted with an arrow and contains the value 'LOC'. Other fields include Index (1), Routing Policy (Default_SBCRoutingP), Name (Local Calls), Alternative Route Options (Route Row), Source IP Group (Any), Request Type (All), Source Username Prefix (*), Source Host (*), Destination Username Prefix (*), Destination Host (*), Message Condition (None), and Call Trigger (Any).

Field	Value
Index	1
Routing Policy	Default_SBCRoutingP
Name	Local Calls
Alternative Route Options	Route Row
Source IP Group	Any
Request Type	All
Source Username Prefix	*
Source Host	*
Source Tags	
Destination Username Prefix	*
Destination Host	*
Destination Tags	LOC
Message Condition	None
Call Trigger	Any

27.3.1 Dial Plan Backward Compatibility



Note: This section is for backward compatibility **only**. It is recommended to migrate your Dial Plan configuration to the latest Dial Plan feature (see 'Using Dial Plan Tags for IP-to-IP Routing' on page 590).

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.
- When using prefix tags, you need to configure manipulation rules to remove the tags before the device sends the calls to their destinations.

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 **New**.
 - b. Click the **Rule** tab, and then configure matching characteristics for the incoming call (e.g., set 'Source IP Group' 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 **New**.
 - 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 **New**.
 - b. Click the **Rule** tab, and then configure matching characteristics for the incoming call (e.g., set 'Source IP Group' 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.

27.4 Using Dial Plan Tags for Outbound Manipulation

You can use Dial Plan tags to denote source and/or destination URI user names in Outbound Manipulation rules in the IP-to-IP Outbound Manipulation table.

The following procedure describes how to configure Outbound Manipulation based on tags.

➤ **To configure Outbound Manipulation based on tags:**

1. In the Dial Plan table configure a Dial Plan (see 'Configuring Dial Plans' on page 511).
2. In the IP Group or SRD associated with the calls for which you want to use tag-based routing, assign the Dial Plan that you configured in Step 1.
 - IP Group table: 'Dial Plan' parameter (IPGroup_SBCDialPlanName) - see Configuring IP Groups
 - SRD table: 'Dial Plan' parameter (SRD_SBCDialPlanName) - see Configuring SRDs
3. In the Outbound Manipulations table (see Configuring IP-to-IP Outbound Manipulations), configure a rule with the required manipulation and whose matching characteristics include the tag(s) that you configured in your Dial Plan in Step 1. The tags are assigned using the following parameters:
 - 'Source Tags' parameter (IPOutboundManipulation_SrcTags): tag denoting the calling users
 - 'Destination Tags' parameter (IPOutboundManipulation_DestTags): tag denoting the called users

28 Advanced SBC Features

28.1 Configuring Call Preemption for SBC Emergency Calls

The device supports emergency call preemption for SBC calls, by prioritizing emergency calls over regular calls. If the device receives an incoming emergency call when there are unavailable resources to process the call, the device preempts one of the active regular calls to free up resources for sending the emergency call to its' destination (i.e., emergency service provider), and not reject it. The device may preempt more than one active call in order to provide sufficient resources for processing the emergency call. Available resources depends on the number of INVITE messages currently being processed by the device.

If the device preempts a call, it disconnects the call as follows:

- If the call is being setup (not yet established), it sends a SIP 488 response to the incoming leg and a SIP CANCEL message to the outgoing leg.
- If the call is already established, it sends a SIP BYE message to each leg. The device includes in the SIP BYE message, the Reason header describing the cause as "preemption".

Once the device terminates the regular call, it immediately sends the INVITE message of the emergency call to its' destination without waiting for any response from the remote sides (e.g., 200 OK after BYE). If the device is unable to preempt a call for the emergency call, it rejects the emergency call with a SIP 503 "Emergency Call Failed" (instead of "Service Unavailable") response.

For the device to identify incoming calls as emergency calls, you need to configure a Message Condition rule. Below are examples of Message Condition rules, configured in the Message Condition table, for identifying emergency calls:

- Indices 0 and 1: SIP Resource-Priority header contains a string indicating an emergency call.
- Indices 2 to 4: Destination user-part contains the emergency provider's address.

Figure 28-1: Examples of Message Condition Rules for Identifying SBC Emergency Calls

Index ↕	Name	Condition
0	Emergency1 - Resource-Priority header	header.resource-priority contains 'emergency'
1	Emergency2 - Resource-Priority header	header.resource-priority contains 'esnet'
2	Emergency1 - user part with providers address	param.call.dst.user == '911'
3	Emergency2 - user part with providers address	param.call.dst.user == '100' param.call.dst.user == '101' param.call.dst.user == '102'
4	Emergency3 - user part with providers address	header.request-uri contains 'urn:service:sos'

The device applies the Message Condition rule only after call classification (but, before inbound manipulation).

➤ To configure SBC emergency call preemption:

1. In the Message Condition table, configure a Message Condition rule to identify incoming emergency calls. See above for examples. For more information on Message Conditions, see "Configuring Message Condition Rules" on page 478.

- Open the SBC General Settings page (**Configuration** tab > **VoIP** > **SBC** > **SBC General Settings**), and then scroll down the page to the Call Priority and Preemption group:

Figure 28-2: Configuring Emergency SBC Call Preemption

Call Priority and Preemption	
SBC Preemption Mode	Enable
SBC Emergency Message Condition	-1
SBC Emergency RTP DiffServ	46
SBC Emergency Signaling DiffServ	40

- From the 'SBC Preemption Mode' drop-down list (SBCPreemptionMode), select **Enable** to enable the SBC call preemption feature.
- In the 'SBC Emergency Message Condition' field, enter the row index of the Message Condition rule that you configured in Step 1 for identifying incoming emergency calls.
- (Optional) Assign DiffServ levels (markings) to packets belonging to emergency calls:
 - In the 'SBC Emergency RTP DiffServ' field (SBCEmergencyRTPDiffServ), enter the QoS level for RTP packets.
 - In the 'SBC Emergency Signaling DiffServ' field (SBCEmergencySignalingDiffServ), enter the QoS level for SIP signaling packets.
- Click **Submit**.



- Note:** The device does not preempt already established emergency calls.

28.2 Emergency Call Routing using LDAP to Obtain ELIN

The device can route emergency calls (e.g., 911) for INVITE messages that are received without an ELIN number. This is in contrast to when the device is deployed in a Microsoft Lync environment, whereby INVITE messages received from Lync contain ELIN numbers. (For a detailed explanation on ELIN numbers and handling of emergency calls by emergency server providers, see "Enhanced 9-1-1 Support for Lync Server" on page 298.)

To obtain an ELIN number for emergency calls received without ELINs, you can configure the device to query an LDAP server for the 911 caller's ELIN number. The device adds the resultant ELIN number and a Content-Type header for the PIDF XML message body to the outgoing INVITE message, for example:

```
Content-Type: application/pidf+xml
<NAM>1234567890</NAM>
```

To enable this functionality, you need to configure a Call Setup rule in the Call Setup Rules table (see "Configuring Call Setup Rules" on page 292). The following example shows a Call Setup rule that queries an Active Directory (AD) server for the attribute "telephoneNumber" whose value is the E9-1-1 caller's number, and then retrieves the user's ELIN number from the attribute, "numberELIN":

Figure 28-3: Example of Call Setup Rule for LDAP Query of ELIN

Index	Rules Set ID	Attributes To Query	Attributes To Get	Row Role	Condition	Action Subject	Action Type	Action Value
0	1	telephoneNumber="+param.call.src.user	numberELIN	Use Current Condit	ldap.attr.numberELIN exists	body,application/pidf+xml	Add	<NAM>'+ldap.attr.numberELIN+'</NAM>'

The rest of the process is similar to emergency call routing in a Lync environment.

Configuration includes the following:

- Enabling E9-1-1 by configuring the 'PSAP Mode' parameter to **PSAP Server** in the IP Group table for the IP Group of the PSAP server (see "Enabling the E9-1-1 Feature" on page 308).
- Configuring routing rules in the IP-to-IP Routing table for routing between the

emergency callers' IP Group and the PSAP server's IP Group. The only special configuration is required on the rule for routing from emergency callers to the PSAP server:

- Configure the emergency number (e.g., 911) in the 'Destination Username Prefix' field.
- Assign the Call Setup rule, which you configured for obtaining the ELIN number from the AD, in the 'Call Setup Rules Set ID' field (see "Configuring SBC IP-to-IP Routing Rule for E9-1-1" on page 309).

28.3 WebRTC

The device supports interworking of Web Real-Time Communication (WebRTC) and SIP-based VoIP communication. The device interworks WebRTC calls made from a Web browser (WebRTC client) and the SIP destination. The device provides the media interface to WebRTC.

WebRTC is a browser-based real-time communication protocol. WebRTC is an open source, client-side API definition (based on JavaScript) drafted by the World Wide Web Consortium (W3C) that supports browser-to-browser applications for voice calling (video chat, and P2P file sharing) without plugins. Currently, WebRTC is supported only by Mozilla Firefox and Google Chrome Web browsers. Though the WebRTC standard has obvious implications for changing the nature of peer-to-peer communication, it is also an ideal solution for customer-care solutions to allow direct access to the contact center. An example of a WebRTC application is a click-to-call button on a consumer Web site (see following figure). After clicking the button, the customer can start a voice and/or video call with a customer service personnel directly from the browser without having to download any additional software plugins. The figure below displays an example of a click-to-call application from a customer Web page, where the client needs to enter credentials (username and password) before placing the call.

Figure 28-4: Example of WebRTC for Click-to-Call Application



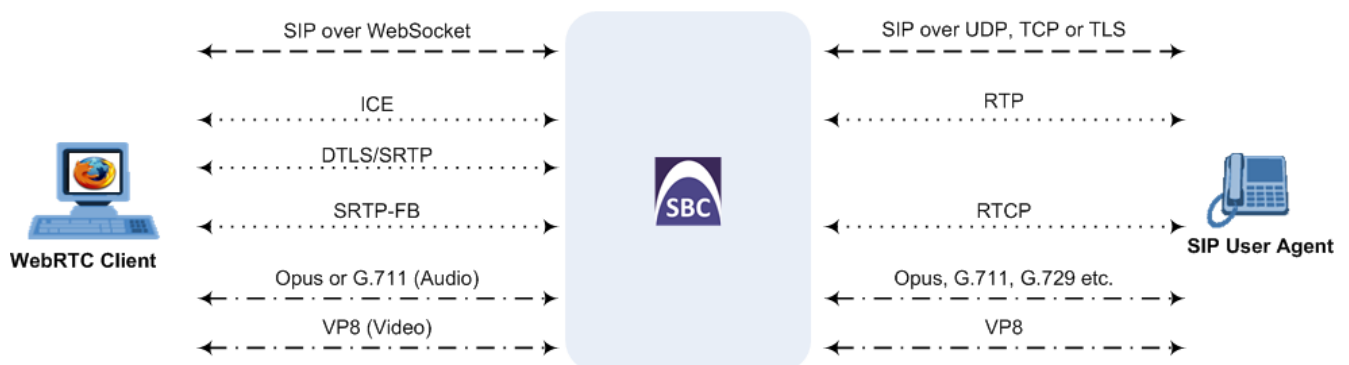


Note: The WebRTC feature is a license-dependent feature and is available only if it is included in the Software License Key that is installed on the device. For ordering the feature, please contact your AudioCodes sales representative.

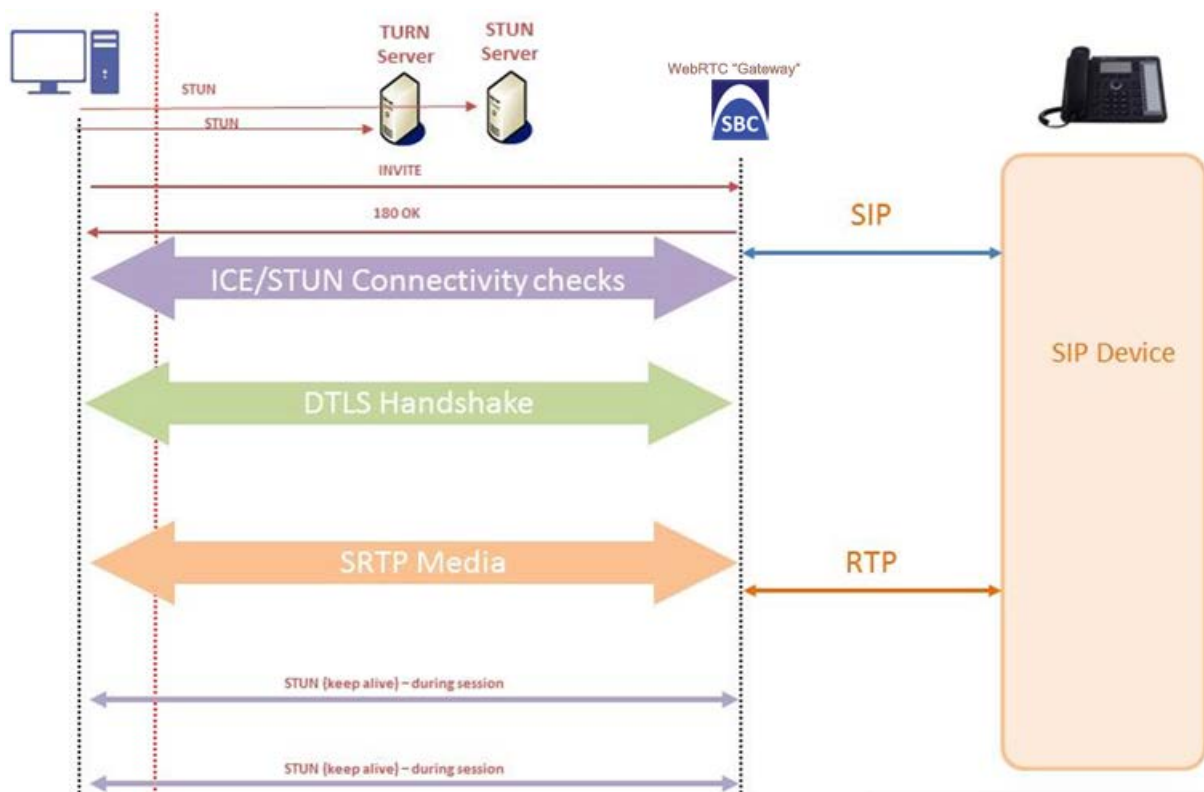
The WebRTC standard requires the following mandatory components, which are supported by the device:

- **Voice coders:** Narrowband G.711 and wideband Opus (Version 1.0.3, per RFC 6176).
- **Video coders:** VP8 video coder. The device transparently forwards the video stream, encoded with the VP8 coder, between the endpoints.
- **ICE (per RFCs 5389/5245):** Resolves NAT traversal problems, using STUN and TURN protocols to connect peers. For more information, see "ICE Lite".
- **DTLS-SRTP (RFCs 5763/5764):** Media channels must be encrypted (secured) through Datagram Transport Layer Security (DTLS) for SRTP key exchange. For more information, see "SRTP using DTLS Protocol" on page 213.
- **SRTP (RFC 3711):** Secures media channels by SRTP.
- **RTP Multiplexing (RFC 5761):** Multiplexing RTP data packets and RTCP control packets onto a single port for each RTP session. For more information, see "Interworking RTP-RTCP Multiplexing".
- **Secure RTCP with Feedback (i.e., RTP/SAVPF format in the SDP - RFC 5124):** Combines secured voice (SRTP) with immediate feedback (RTCP) to improve session quality. The SRTP profile is called SAVPF and must be in the SDP offer/answer (e.g., "m=audio 11050 RTP/SAVPF 103"). For more information, see the IP Profile parameter, IPProfile_SBCRTCPFeedback (see "Configuring IP Profiles" on page 391).
- **WebSocket:** WebSocket is a signaling (SIP messaging) transport protocol, providing full-duplex communication channels over a single TCP connection for Web browsers and clients. SIP messages are sent to the device over the WebSocket session. For more information, see "SIP over WebSocket" on page 526.

For more information on WebRTC, go to <http://www.webrtc.org/>. Below shows a summary of the WebRTC components and the device's interworking of these components between the WebRTC client and the SIP user agent:



The call flow process for interworking WebRTC with SIP endpoints by the device is illustrated below and subsequently described:



1. The WebRTC client uses a Web browser to visit the Web site page.
2. The Web page receives Web page elements and JavaScript code for WebRTC from the Web hosting server. The JavaScript code runs locally on the Web browser.
3. When the client clicks the Call button or call link, the browser runs the JavaScript code which sends the HTTP upgrade request for WebSocket in order to establish a WebSocket session with the device. The address of the device is typically included in the JavaScript code.
4. A WebSocket session is established between the WebRTC client and the device in order for the WebRTC client to register with the device. This is done using a SIP REGISTER message sent over the WebSocket session (SIP over WebSocket). Registration can be initiated when the client enters credentials (username and password) on the Web page or it can be done automatically when the client initially browses to the page. This depends on the design of the Web application (JavaScript).
5. Once registered with the device, the client can receive or make calls, depending on the Web application.
6. To make a call, the client clicks the call button or link on the Web page.
7. Negotiation of a workable IP address between the WebRTC client and the device is done through ICE.
8. Negotiation of SRTTP keys using DTLS is done between WebRTC and the client on the media.
9. Media flows between the WebRTC client and the SIP client located behind the device.

28.3.1 SIP over WebSocket

The device supports the transmission of SIP signaling over WebSocket. WebSocket is a protocol providing real-time, full-duplex (two-way) communication over a single TCP connection (socket) between a Web browser or page (client) and a remote host (server). This is used for browser-based applications such as click-to-call from a Web page. As WebSocket has been defined by the WebRTC standard as mandatory, its support by the device is important for deployments implementing WebRTC.

A WebSocket connection starts as an HTTP connection between the Web client and the server, guaranteeing full backward compatibility with the pre-WebSocket world. The protocol switch from HTTP to WebSocket is referred to as the WebSocket handshake, which is done over the same underlying TCP/IP connection. A WebSocket connection is established using a handshake between the Web browser (WebSocket client) and the server (i.e., the device). The browser sends a request to the server, indicating that it wants to switch protocols from HTTP to WebSocket. The client expresses its' desire through the Upgrade header (i.e., upgrade from HTTP to WebSocket protocol) in an HTTP GET request, for example:

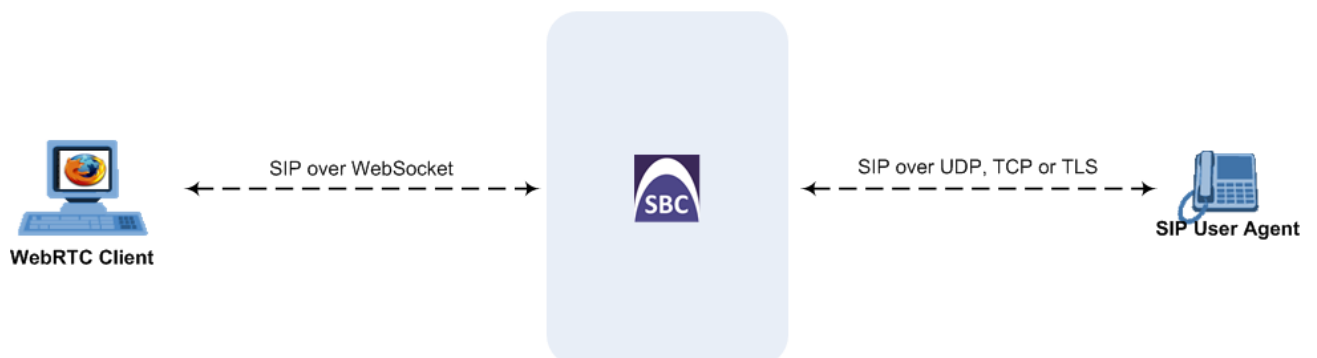
```
GET /chat HTTP/1.1
Upgrade: websocket
Connection: Upgrade
Host: <IP address:port of SBC device>
Sec-WebSocket-Protocol: SIP
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: <server that provided JavaScript code to browser, e.g.,
http://domain.com>
Sec-WebSocket-Version: 13
```

If the server understands the WebSocket protocol, it agrees to the protocol switch through the Upgrade header in an HTTP 101 response, for example:

```
HTTP/1.1 101 Switching Protocols
Upgrade: WebSocket
Connection: Upgrade
Sec-WebSocket-Accept: rLHCkw/SKs09GAH/ZSFhBATDKrU=
Sec-WebSocket-Protocol: SIP
Server: SBC
```

At this stage, the HTTP connection breaks down and is replaced by a WebSocket connection over the same underlying TCP/IP connection. By default, the WebSocket connection uses the same ports as HTTP (80) and HTTPS (443).

Once a WebSocket connection is established, the SIP messages are sent over the WebSocket session. The device, as a "WebSocket gateway" or server can interwork WebSocket browser originated traffic to SIP over UDP, TCP or TLS, as illustrated below:



The SIP messages over WebSocket are indicated by the "ws" value, as shown in the example below of a SIP REGISTER request received from a client:

```
REGISTER sip:10.132.10.144 SIP/2.0
Via: SIP/2.0/WS v6iqlt8lne5c.invalid;branch=z9hG4bK7785666
Max-Forwards: 69
To: <sip:101@10.132.10.144>
From: "joe" <sip:101@10.132.10.144>;tag=ub50pqjgpr
Call-ID: fhddgc3kc3hhu32h01fghl
CSeq: 81 REGISTER
Contact: <sip:0bfr9fd5@v6iqlt8lne5c.invalid;transport=ws>;reg-id=1;+sip.instance="urn:uuid:4405bbe2-cf06-4c27-9c59-6caf83af9b00>;expires=600
Allow: ACK,CANCEL,BYE,OPTIONS,INVITE,MESSAGE
Supported: path, outbound, gruu
User-Agent: JsSIP 0.3.7
Content-Length: 0
```

To keep a WebSocket session alive, it is sometimes necessary to send regular messages to indicate that the channel is still being used. Some servers, browsers or proxies may close an idle connection. The Ping-Pong WebSocket messages are designed to send non-application level traffic that prevents the channel from being prematurely closed. You can configure how often the device pings the WebSocket client, using the `WebSocketProtocolKeepAlivePeriod` parameter (see "Configuring WebRTC" on page 527). The device always replies to ping control messages with a pong message.



Note: When the device operates in High-Availability (HA) mode, if a WebSocket connection has been established and a switchover subsequently occurs, the WebSocket session is not copied to the redundant device. As Chrome does not renew the WebSocket connection with the device, WebRTC calls remain open indefinitely; the Chrome side will stop the call, but the device will keep all of the call's resources open and the other side will have an active call with no voice. To prevent this, for the IP Profile associated with the WebRTC clients, configure the 'Broken Connection Mode' parameter to **Disconnect**.

28.3.2 Configuring WebRTC

To support WebRTC, you need to perform special configuration settings for the device's SBC leg interfacing with the WebRTC client (i.e., Web browser). The following procedure describes the required configuration.

➤ **To configure WebRTC:**

1. Configure DTLS for communication between the device and the WebRTC client (see "SRTP using DTLS Protocol" on page 213)
2. Enable ICE: For the IP Profile associated with the WebRTC clients, configure the 'ICE Mode' parameter (IPProfile_SBCIceMode) to **Lite** (1).
3. Enable RTCP Feedback: For the IP Profile associated with the WebRTC clients, configure the 'RTCP Feedback' parameter (IPProfile_SBCRTCPFeedback) to **Enable** (1).
4. Enable RTCP Multiplexing: For the IP Profile associated with the WebRTC clients, configure the 'RTCP Mux' parameter (IPProfile_SBCRTCPMux) to **Supported** (1).
5. WebSocket:
 - a. On the SIP General Parameters page, configure the keep-alive interval with the WebSocket client, using the 'WebSocket Keep-Alive Period' parameter (`WebSocketProtocolKeepAlivePeriod`).

- b. Open the SIP Interface table ("Configuring SIP Interfaces" on page 342) and for the SIP Interface associated with the WebRTC clients, configure the 'Encapsulating Protocol' parameter (SIPInterface_EncapsulatingProtocol) to **WebSocket** (1). The setting identifies WebSocket traffic on the SIP Interface.

28.4 Configuring Dual Registration

Some SIP entities (e.g., IP Phones) are setup to register with two registrar/proxy servers (primary and secondary). The reason for this is to provide call redundancy for the SIP entity in case one of the proxy servers fail. When the SIP entity registers with the proxy servers, it sends two identical REGISTER messages - one to the primary proxy and one to the secondary proxy. When the device is located between the SIP entity and the two proxy servers, it needs to differentiate between these two REGISTER messages even though they are identical. This is crucial to ensure that the device forwards the two registrations to the proxy servers and that the device performs correct call routing between the SIP entity and the two proxy servers.

To differentiate between these REGISTER messages, a unique SIP Interface needs to be used for each REGISTER message. Each REGISTER message is registered in the registration database using a unique "ac-int=<value>" string identifying the SIP Interface for the Contact user. In addition, for SIP requests (e.g., INVITE) from the proxy servers, the device needs to search its registration database for the contact user so that it can forward it to the user. In normal registration, the host part of the Request-URI contains the IP address of the device and therefore, there is no way of knowing which registered user the INVITE is intended for. To overcome this issue, you can configure the device to use a special string with a unique value, "ac-feu=<value>" for each registration, allowing the device to differentiate between two registrations from the same user (identical REGISTER requests). Each REGISTER message is registered in the registration database using the unique "ac-feu=" string identifier for the Contact user.

A summary of how the device registers the two REGISTER messages in its registration database is as follows:

1. The addresses of the proxy servers that are configured on the SIP entity (IP Phone) must be the device's IP address with a different SIP local port for each one, for example:
 - Primary Proxy Server: 172.17.0.1:5060
 - Secondary Proxy Server: 172.17.0.1:5080
2. When the device receives two identical REGISTER messages from the SIP entity, it differentiates them by the SIP port on which they are received. The port allows the device to associate them with a SIP Interface (5060 for "Interface-1" and 5080 for "Interface-2").
3. The device performs SIP message manipulation (Pre-classification Manipulation) on the REGISTER messages to add a special parameter ("ac-int=<value>") to the Contact header to identify the SIP Interface on which each message is received. For example:
 - REGISTER for Primary Proxy received on SIP Interface "Interface-1":

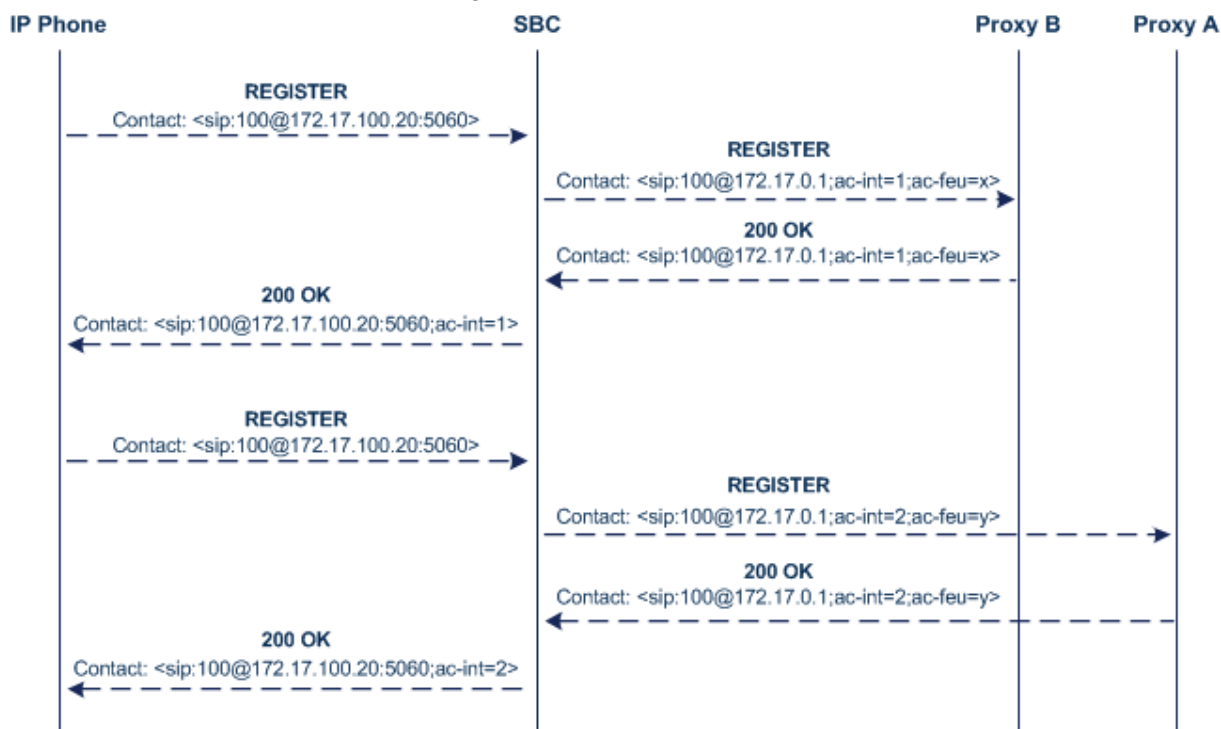
```
REGISTER
To: sip:100@audc.com
Contact: <sip:100@172.17.100.20;ac-int=1>
```
 - REGISTER for Secondary Proxy received on SIP Interface "Interface-2":

```
REGISTER
To: sip:100@audc.com
Contact: <sip:100@172.17.100.20;ac-int=2>
```
4. The device adds to the Contact header a special string with a unique value, "ac-feu=<value>" for each registration (e.g., "Contact: <sip:100@172.17.100.20;ac-

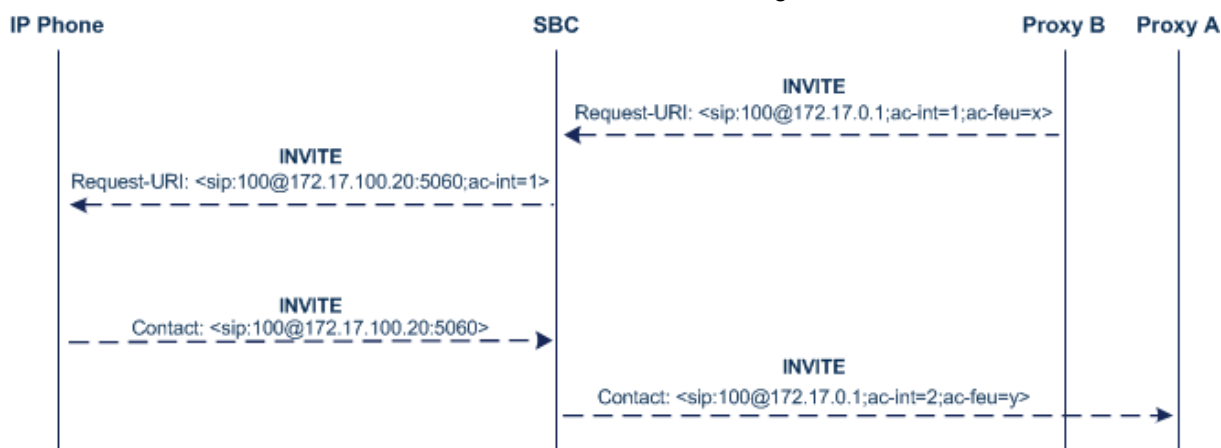
int=1;ac-feu=x>").

- The device saves the two contacts (100@172.17.100.20;ac-int=1;ac-feu=x and 100@172.17.113.32;ac-int=2;ac-feu=y) under the same AOR (100@audc.com) in its user registration database.

The SIP call flow for dual registration is shown below:



The basic SIP call flow for INVITEs to and from the registered user is shown below:



➤ **To configure support for dual registration:**

- On the SIP entity (IP Phone), configure the primary and secondary proxy server addresses as the IP address of the device and where each address has a different SIP port number.
- Open the SBC General Settings page (**Setup** menu > **Signaling & Media** tab > **SBC** folder > **SBC General Settings**), and then from the 'Keep Original User in Register' drop-down list, select **Keep user; add unique identifier as URI parameter**.
- In the Message Manipulations table, configure the following rules:
 - Index 0:

- ◆ Manipulation Set ID: **1**
- ◆ Action Subject: **header.contact.url.ac-int**
- ◆ Action Type: **Modify**
- ◆ Action Value: **'1'**
- Index 1:
 - ◆ Manipulation Set ID: **2**
 - ◆ Action Subject: **header.contact.url.ac-int**
 - ◆ Action Type: **Modify**
 - ◆ Action Value: **'2'**
- 4. In the SIP Interfaces table, configure the following SIP Interfaces:
 - Index 0 (SIP Interface for IP Phone A):
 - ◆ Name: **Interface-1**
 - ◆ UDP Port: **5060**
 - ◆ Pre-classification Manipulation Set ID: **1**
 - Index 1 (SIP Interface for IP Phone B):
 - ◆ Name: **Interface-2**
 - ◆ UDP Port: **5080**
 - ◆ Pre-classification Manipulation Set ID: **2**
- 5. In the Proxy Sets table, configure a Proxy Set for each proxy server (primary and secondary):
 - Index 0:
 - ◆ Proxy Name: **Primary**
 - ◆ SBC IPv4 SIP Interface: **Interface-1**
 - ◆ Proxy Address: **200.10.10.1**
 - Index 1:
 - ◆ Proxy Name: **Secondary**
 - ◆ SBC IPv4 SIP Interface: **Interface-2**
 - ◆ Proxy Address: **200.10.10.2**
- 6. In the IP Groups table, configure the following IP Groups:
 - Index 0:
 - ◆ Type: **Server**
 - ◆ Name: **Primary-Proxy**
 - ◆ Proxy Set: **Primary**
 - ◆ Classify By Proxy Set: **Enable**
 - Index 1:
 - ◆ Type: **Server**
 - ◆ Name: **Sec-Proxy**
 - ◆ Proxy Set: **Secondary**
 - ◆ Classify By Proxy Set: **Enable**
 - Index 2:
 - ◆ Type: **User**
 - ◆ Name: **IP-Phone-A**
 - Index 3:
 - ◆ Type: **User**
 - ◆ Name: **IP-Phone-B**
- 7. In the Classification table, configure rules to classify calls from the IP Phones based

on SIP Interface:

- Index 0:
 - ♦ Source SIP Interface: **Interface-1**
 - ♦ Source IP Group: **IP-Phone-A**
- Index 1:
 - ♦ Source SIP Interface: **Interface-2**
 - ♦ Source IP Group: **IP-Phone-B**

8. In the IP-to-IP Routing table, configure the routing rules:

- Index 0:
 - ♦ Source IP Group: **IP-Phone-A**
 - ♦ Destination IP Group: **Primary-Proxy**
- Index 1:
 - ♦ Source IP Group: **Primary-Proxy**
 - ♦ Destination IP Group: **IP-Phone-A**
- Index 2:
 - ♦ Source IP Group: **IP-Phone-B**
 - ♦ Destination IP Group: **Sec-Proxy**
- Index 3:
 - ♦ Source IP Group: **Sec-Proxy**
 - ♦ Destination IP Group: **IP-Phone-B**

28.5 Call Forking

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

28.5.1 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 348).

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.

28.5.2 SIP Forking Initiated by SIP Proxy Server

The device can handle the receipt of multiple SIP 18x responses as a result of SIP forking initiated by a proxy server. This occurs when the device sends an INVITE, received from a user agent (UA), to a proxy server and the proxy server then forks the INVITE request to multiple UAs. Several UAs may answer and the device may therefore, receive several replies (responses) for the single INVITE request. Each response has a different 'tag' value in the SIP To header.

During call setup, forked SIP responses may result in a single SDP offer with two or more SDP answers. The device "hides" all the forked responses from the INVITE-initiating UA, except the first received response ("active" UA) and it forwards only subsequent requests and responses from this active UA to the INVITE-initiating UA. All requests/responses from the other UAs are handled by the device; SDP offers from these UAs are answered with an "inactive" media.

The device supports two forking modes, configured by the `SBCForkingHandlingMode` parameter:

- **Latch On First:** The device forwards only the first received 18x response to the INVITE-initiating UA and disregards subsequently received 18x forking responses (with or without SDP).
- **Sequential:** The device forwards all 18x responses to the INVITE-initiating UA, sequentially (one after another). If 18x arrives with an offer only, only the first offer is forwarded to the INVITE-initiating UA.

The device 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, it is possible that the SDP answer that was forwarded to the INVITE-initiating UA is irrelevant and thus, 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 SDP offer to the INVITE-initiating UA. This causes the INVITE-initiating UA to send an offer which the device forwards to the UA that confirmed the call. Media synchronization is enabled by the `EnableSBCMediaSync` parameter.

28.5.3 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 480.

28.6 Call Survivability

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

28.6.1 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 28-5: 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>

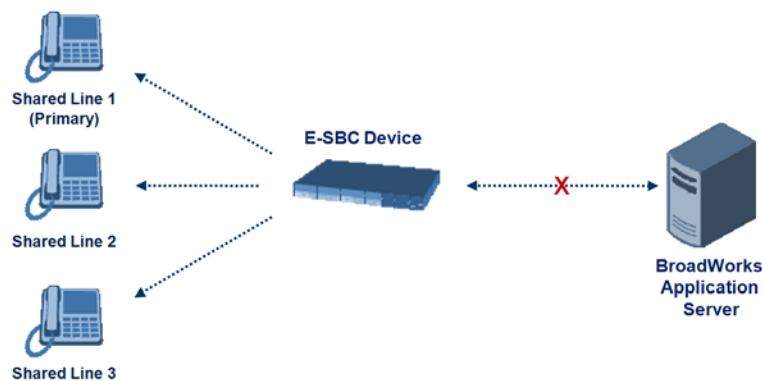
```

28.6.2 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 532. Note that incoming calls specific to extensions 601 or 602 ring only at these specific extensions.

Figure 28-6: 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 348), 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 480), 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 499), 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 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).

28.6.3 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 28-7: Normal Operation in Call Center Application

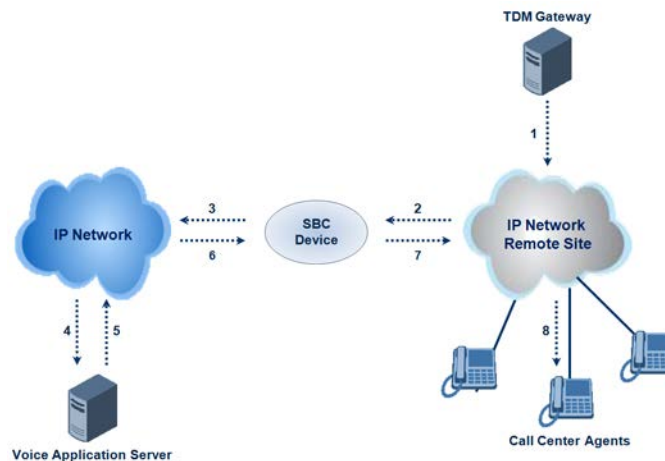
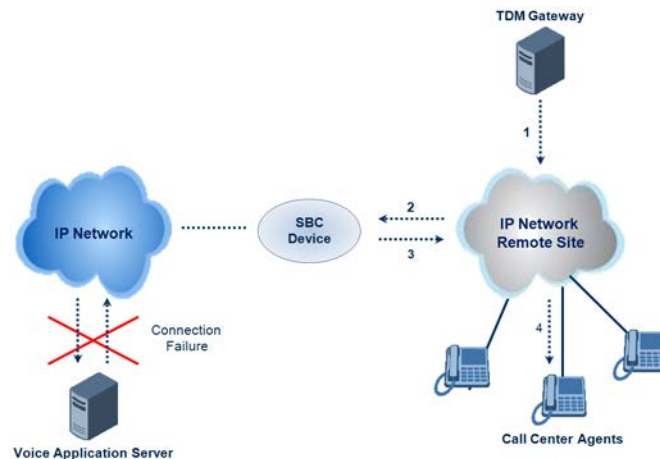


Figure 28-8: Call Survivability for Call Center



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

1. In the IP Group table (see "Configuring IP Groups" on page 348), 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 471), 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 480), 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' 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' 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 28-9: 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	

28.6.4 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>
```

28.7 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.

Part VI

Cloud Resilience Package

29 CRP Overview

The device's Cloud Resilience Package (CRP) application enhances cloud-based or hosted communications environments by ensuring survivability, high voice quality and security at enterprise branch offices and cloud service customer premises. CRP is designed to be deployed at customer sites and branches of:

- Cloud-based and hosted communications
- Cloud-based or hosted contact-center services
- Distributed PBX or unified communications deployments

The CRP application is based on the functionality of the SBC application, providing branch offices with call routing and survivability support similar to AudioCodes' Stand-Alone Survivability (SAS) application. CRP is implemented in a network topology where the device is located at the branch office, routing calls between the branch users, and/or between the branch users and other users located elsewhere (at headquarters or other branch offices), through a hosted server (IP PBX) located at the Enterprise's headquarters. The device maintains call continuity even if a failure occurs in communication with the hosted IP PBX.



Notes:

- The CRP application 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 596.
- For the maximum number of supported CRP sessions and CRP users than can be registered in the device's registration database, see "Technical Specifications" on page 837.
- The CRP application supersedes the SAS application and is the recommended application to use. However, SAS is still supported by the device. For a detailed description on SAS, refer to the *SAS Application Configuration Guide*.

For cloud providers, CRP ensures uninterrupted communications in the event of lost connection with the cloud providers' control systems. For distributed enterprises and contact centers, CRP is an essential solution for enterprises deploying geographically distributed communications solutions or distributed call centers with many branch offices. CRP ensures the delivery of internal and external calls even when the connection with the centralized control servers is lost.

Table 29-1: Key Features

Survivability	Quality of Experience/Service	Security
■ PSTN fallback ■ WAN redundancy ■ Local mode ■ High availability ■ Emergency calling (E911) ■ Basic call routing between registering users and device, or any other route to responding server ■ Short number dialog (short numbers are learned dynamically in the	■ QoE monitoring ■ Call Admission Control ■ SLA fulfillment ■ SIP mediation ■ Media transcoding ■ Test call agent	■ Layer 3 to 7 protection ■ Media encryption ■ Call control encryption ■ NAT traversal ■ Topology hiding

Survivability	Quality of Experience/Service	Security
registration process) - Survivability indication to IP phone - Call hold and retrieve - Call transfer (if IP phone initiates REFER) - Basic Shared Line Appearance (excluding correct busy line indications) - Call waiting (if supported by IP phone)		

One of the main advantages of CRP is that it enables quick-and-easy configuration. This is accomplished by its pre-configured routing entities, whereby only minimal configuration is required. For example, defining IP addresses to get the device up and running and deployed in the network.

30 CRP Configuration

This section describes configuration specific to the CRP application. As CRP has similar functionality to the SBC application, for configuration that is common to the SBC, which is not covered in this section, see the following SBC sections:

- "Configuring General SBC Settings" on page 459
- "Configuring Admission Control" on page 463
- "Configuring Allowed Audio Coder Groups" on page 467
- "Configuring Classification Rules" on page 471
- "Configuring Message Condition Rules" on page 478
- "Configuring SBC IP-to-IP Routing Rules" on page 480
- "Configuring SIP Response Codes for Alternative Routing Reasons" on page 491
- "Configuring IP-to-IP Inbound Manipulations" on page 499
- "Configuring IP-to-IP Outbound Manipulations" on page 504



Note: The main difference in the common configuration between the CRP and SBC applications is the navigation menu paths to opening these Web configuration pages. Wherever "SBC" appears in the menu path, for the CRP application it appears as "CRP".

30.1 Enabling the CRP Application

Before you can start configuring the CRP, you must first enable the CRP application. Once enabled, the Web interface displays the menus and parameter fields relevant to the CRP application.



Note: The CRP 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 596.

➤ **To enable the CRP application:**

1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** > **Applications Enabling**).
2. From the 'CRP 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.

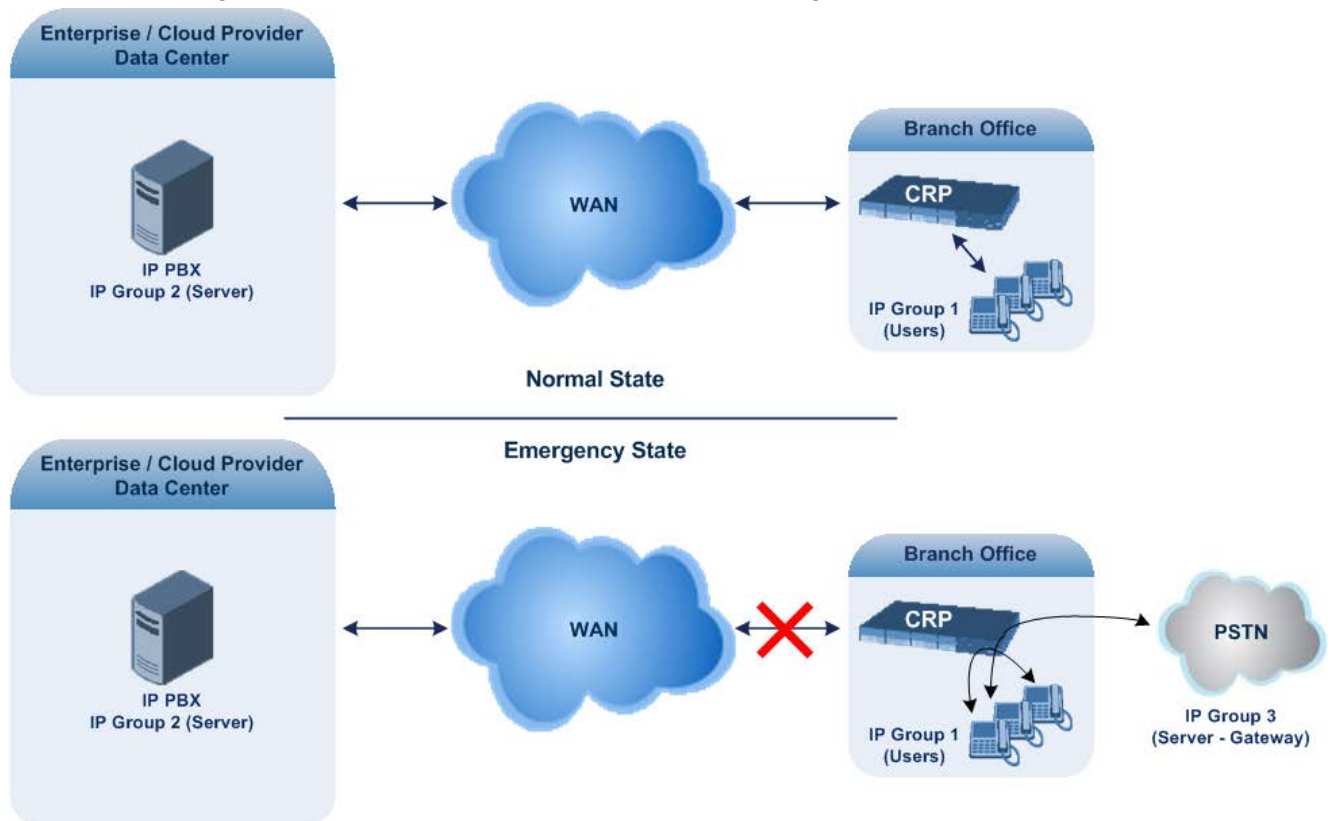
30.2 Configuring Call Survivability Mode

The CRP can be configured to operate in one of the following call survivability modes:

- **Normal (Default):** The CRP interworks between the branch users and the IP PBX located at headquarters. The CRP forwards all requests (such as for registration) from the branch users to the IP PBX, and routes the calls based on the IP-to-IP routing rules. If communication with the IP PBX fails (i.e., Emergency mode), it still allows calls between the branch users themselves. If this fails, it routes the calls to the PSTN

(if employed).

Figure 30-1: CRP in Normal & Auto Answer to Registrations Modes



- **Auto Answer to Registrations:** This mode is the same as the Normal mode, except that the CRP registers the branch users in its registration database instead of forwarding them to the IP PBX.



Note: SIP REGISTER and OPTIONS requests are terminated at the CRP.

- **Always Emergency:** The CRP routes the calls between the branch users themselves as if connectivity failure has occurred with the IP PBX. The CRP also registers the branch users in its registration database.

Figure 30-2: CRP in Always Emergency Mode



➤ **To configure the Call Survivability mode:**

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **CRP** > **General Settings**).
2. From the 'CRP Survivability Mode' drop-down list, select the required mode.
3. Click **Submit**.

30.3 Pre-Configured IP Groups

For CRP, the device is pre-configured with the following IP Groups in the IP Group table:

Table 30-1: Pre-configured IP Groups in the IP Group Table

Index	Name	Type	Description
1	"CRP Users"	User	LAN users (e.g., IP phones) at the branch office
2	"CRP Proxy"	Server	Server (e.g., hosted IP PBX at the Enterprise's headquarters)
3	"CRP Gateway"	Server	Device's interface with the PSTN

These IP Groups are used in the IP-to-IP routing rules to indicate the source and destination of the call (see "Pre-Configured IP-to-IP Routing Rules" on page 547).



Notes:

- These IP Groups cannot be deleted and additional IP Groups cannot be configured. The IP Groups can be edited, except for the fields listed above, which are read-only.
- For accessing the IP Group table and for a description of its parameters, see "Configuring IP Groups" on page 348.

30.4 Pre-Configured IP-to-IP Routing Rules

For the CRP application, the IP-to-IP Routing table is pre-configured with IP-to-IP routing rules. These rules depend on the configured Call Survivability mode, as described in "Configuring Call Survivability Mode" on page 545.



Notes:

- The IP-to-IP Routing table is read-only.
- For accessing the IP-to-IP Routing table and for a description of its parameters, see "Configuring SBC IP-to-IP Routing Rules" on page 480.

30.4.1 Normal Mode

The pre-configured IP-to-IP routing rules for the Normal CRP call survivability mode are shown in the table below:

Table 30-2: Pre-Configured IP-to-IP Routing Rules for CRP Normal Mode

Index	Source IP Group / Emergency	Request Type	Destination Type	Destination IP Group	Destination Address	Alternative Route Options
1	Any	OPTIONS	Dest Address	-	Internal	Route Row
3	#1 [CRP Users]	All	IP Group	#2 [CRP Proxy]	-	Route Row
4	#1 [CRP Users]	All	IP Group	#1 [CRP Users]	-	Alternative Route Ignore Inputs
5	#1 [CRP Users]	All	IP Group	#3 [CRP Gateway]	-	Alternative Route Ignore Inputs
6	#2 [CRP Proxy]	All	IP Group	#1 [CRP Users]	-	Route Row
7	#2 [CRP Proxy]	All	IP Group	#3 [CRP Gateway]	-	Route Row
8	#3 [CRP Gateway]	All	IP Group	#2 [CRP Proxy]	-	Route Row
9	#3 [CRP Gateway]	All	IP Group	#1 [CRP Users]	-	Alternative Route Ignore Inputs



Note: Index 7 appears only if the CRPGatewayFallback parameter is enabled (see 'Configuring PSTN Fallback' on page 551). This routing rule is used if the device can't find a matching destination user for IP Group 1 (User-type IP Group) in its registration database. If the CRPGatewayFallback parameter is disabled and no matching user is found, the device rejects the call.

30.4.2 Emergency Mode

The pre-configured IP-to-IP routing rules for the Emergency CRP call survivability mode are shown in the table below:

Table 30-3: Pre-Configured IP-to-IP Routing Rules for Emergency Mode

Mode	Index	Source IP Group / Emergency	Request Type	Destination Type	Destination IP Group ID	Destination Address	Alternative Route Options
Always Emergency	1	Any	OPTIONS	Dest Address	-	Internal	Route Row
	4	#1 [CRP Users]	All	IP Group	#1 [CRP Users]	-	Route Row
	5	#1 [CRP Users]	All	IP Group	#3 [CRP Gateway]	-	Route Row
	9	#3 [CRP Gateway]	All	IP Group	#1 [CRP Users]	-	Route Row



Note: The routing rule at Index 5 appears only if the CRPGatewayFallback parameter is enabled (1).

30.4.3 Auto Answer to Registrations

The pre-configured IP-to-IP routing rules for the Auto Answer to Registrations CRP call survivability mode are shown in the table below:

Table 30-4: Pre-Configured IP-to-IP Routing Rule for Auto Answer to Registrations Mode

Mode	Index	Source IP Group	Request Type	Destination Type	Destination IP Group	Destination Address	Alternative Route Options
Auto Answer to Registrations	1	Any	OPTIONS	Dest Address	-	Internal	Route Row
	2	Any	REGISTER	IP Group	Any	-	Route Row
	3	#1 [CRP Users]	All	IP Group	#2 [CRP Proxy]	-	Route Row
	4	#1 [CRP Users]	All	IP Group	#1 [CRP Users]	-	Alternative Route Ignore Inputs
	5	#1 [CRP Users]	All	IP Group	#3 [CRP Gateway]	-	Alternative Route Ignore Inputs
	6	#2 [CRP Proxy]	All	IP Group	#1 [CRP Users]	-	Route Row
	7	#2 [CRP Proxy]	All	IP Group	#3 [CRP Gateway]	-	Route Row
	8	#3 [CRP Gateway]	All	IP Group	#2 [CRP Proxy]	-	Route Row
	9	#3 [CRP Gateway]	All	IP Group	#1 [CRP Users]	-	Alternative Route Ignore Inputs



Note:

- The destination for the routing rule at Index 2 is the source IP Group (i.e., from where the REGISTER message is received).
- Routing rule at Index 7 appears only if the CRPGatewayFallback parameter is enabled (see Configuring PSTN Fallback on page 551).

30.5 Configuring PSTN Fallback

You can enable the CRP to route emergency calls (or PSTN-intended calls) such as "911" from the Proxy server (IP Group 2) to the PSTN (IP Group 3). In addition, for calls from the Proxy server to Users (IP Group 1), the device searches for a matching user in its Users Registration database and if not located, it sends the call to the PSTN (IP Group 3), as an alternative route.

To enable this feature, set the ini file parameter CRPGatewayFallback to 1. When enabled, the alternative routing rule appears immediately below the IP Group 2 to IP Group 1 rule in the IP-to-IP Routing table.

**Notes:**

- Enabling this feature (this routing rule) may expose the device to a security "hole", allowing calls from the WAN to be routed to the Gateway. Thus, configure this feature with caution and only if necessary.
- This PSTN routing rule is not an alternative routing rule. In other words, if a match for a user is located in the database, this PSTN rule will never be used regardless of the state of the user endpoint (e.g., busy).

This page is intentionally left blank.

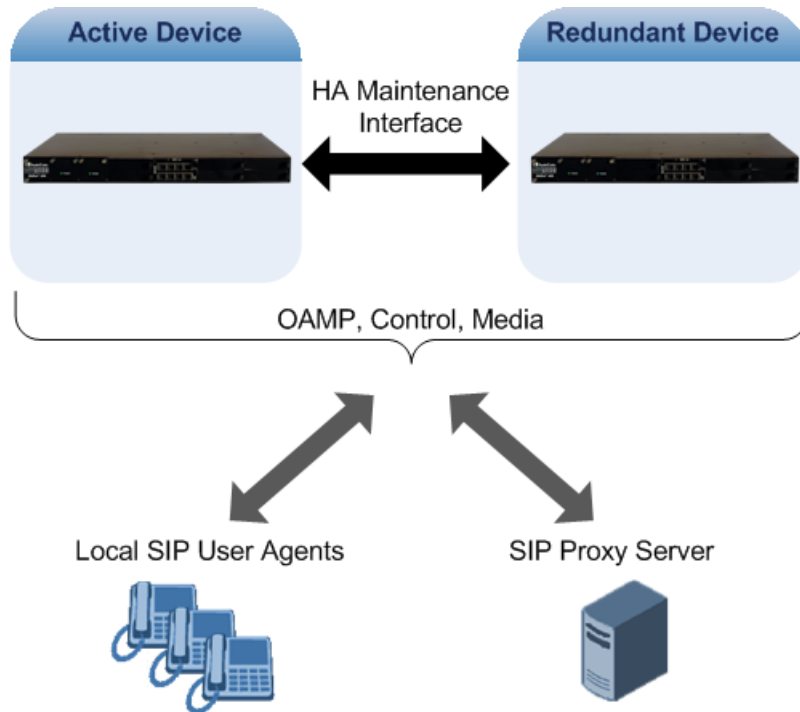
Part VII

High Availability System

31 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.



31.1 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 Group or share the same Ethernet 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. This ensures that the two units are synchronized regarding configuration and software.



Note: If the active unit runs an earlier version (e.g., 6.8) than the redundant unit (e.g., 7.0), the redundant unit is downgraded to the same version as the active unit (e.g., 6.8).

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.

31.2 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 different priority levels (1 to 10, where 1 is the lowest) to the two devices. Typically, you would configure the active device with a higher priority level (number) than the redundant device. The only factor that influences the configuration is which device has the greater number; the actual number is not important. For example, configuring the active with 5 and redundant with 4, or active with 9 and redundant with 2 both assign highest priority to the active device. 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, Preempt mode is disabled. Please see note below when using priority level 10.
- **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.

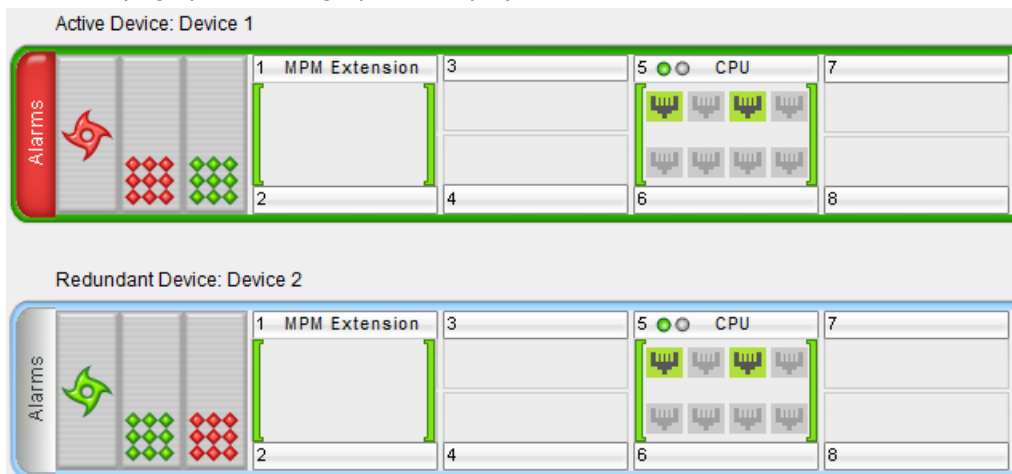


Notes:

- 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 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.

31.3 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.



- 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)

32 HA Configuration

This section describes the configuration of the HA system.

32.1 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.

32.1.1 Network Topology Types and Rx/Tx Ethernet Port Group Settings

Initial HA configuration depends on how you want to deploy your HA system in the network. The Maintenance interface, which is used for the HA link between active and redundant device can use a dedicated Ethernet Device and Ethernet Group (port), or share the same Ethernet Device and Ethernet Group with other IP network interface types (such as OAMP, Media and Control). However, it is recommended that you configure the Maintenance interface with a dedicated Ethernet Device and Ethernet Group to separate it from other IP network interfaces.

If you want to separate the Maintenance interface from other interfaces, the separation must also be done externally to the units, either by physical separation (i.e., different physical networks) or by logical separation (using VLANs). When using VLANs for this separation, make sure that you use a different Underlying Ethernet Device for each IP network interface (see [Configuring Underlying Ethernet Devices](#) on page 135 and [Configuring IP Network Interfaces](#) on page 138).



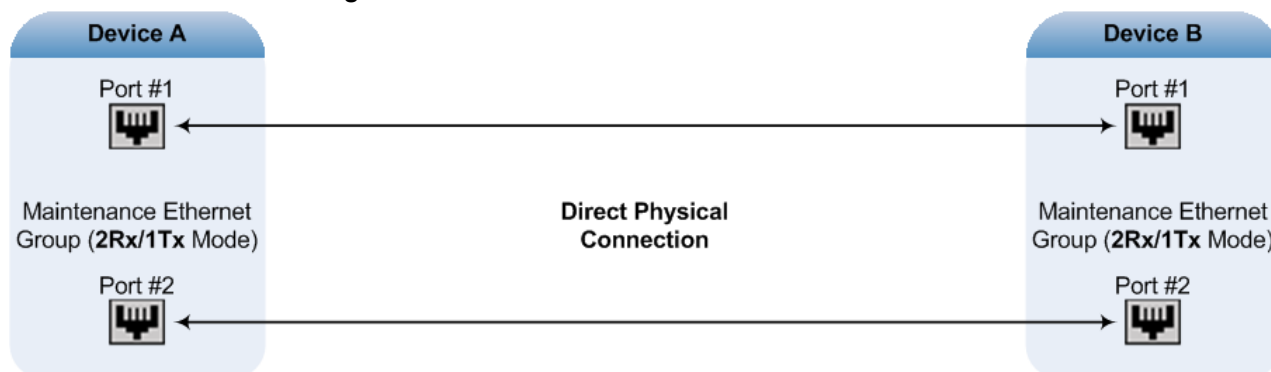
Note:

- The Maintenance interface is used for heartbeats and data transfer from active to standby device and therefore, any short interval interruption in communication may cause undesired switchovers.
- If you assign the same Underlying Ethernet Device to all the IP network interfaces, logical separation of traffic may not occur.

The Maintenance interface can employ Ethernet port redundancy (recommended), by using two ports. This is enabled by configuring the Ethernet Group associated with the Maintenance interface with two ports. The required receive (Rx) and transmit (TX) mode for the port pair in the Ethernet Group used by the Maintenance interface is as follows:

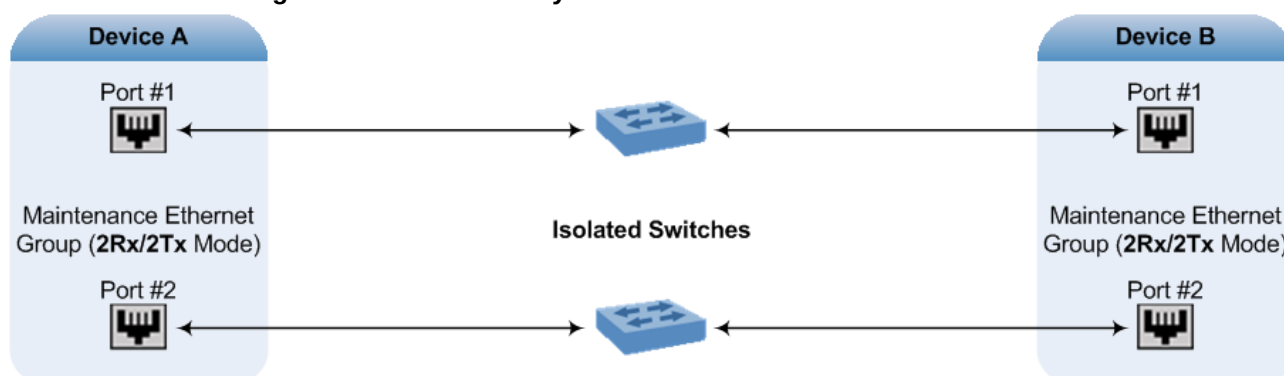
- If the Maintenance ports of both devices are connected directly to each other without intermediation of switches, configure the mode to **2RX/1TX**:

Figure 32-1: Rx/Tx Mode for Direct Connection



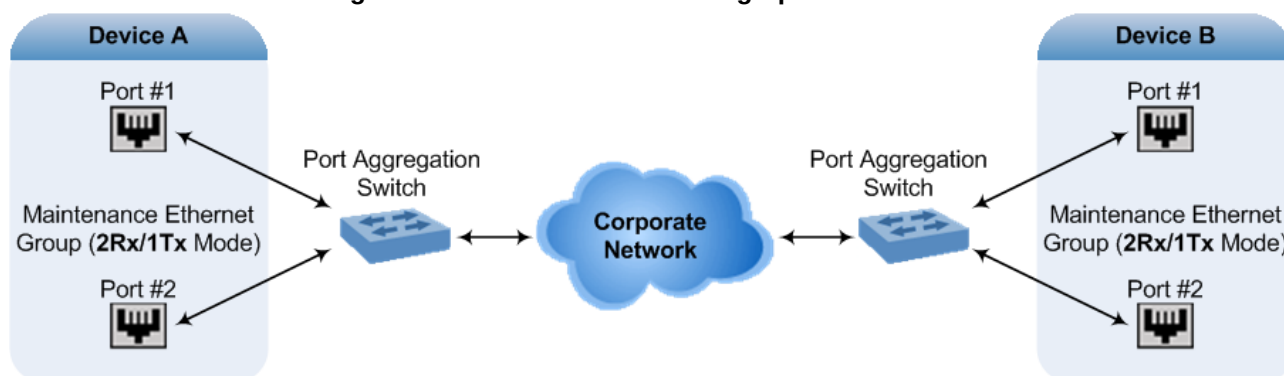
- If the two devices are connected through two (or more) isolated LAN switches (i.e., packets from one switch cannot traverse the second switch), configure the mode to **2RX/2TX**:

Figure 32-2: Redundancy Mode for Two Isolated Switches



- 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 32-3: Rx/Tx Mode for Geographical HA



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 "Configuring Ethernet Port Groups" on page 133

32.1.2 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 561
2. Configuring the second device for HA - see "Step 2: Configure the Second Device" on page 563
3. Activating HA on the devices - see "Step 3: Initialize HA on the Devices" on page 565



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 596.
- 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 Groups and the port numbers belonging to these Ethernet Groups. For example, if the first device uses Ethernet Group 1 (with ports 1 and 2), the second device must also use Ethernet 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 559.
- 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.

32.1.2.1 Step 1: Configure the First Device

The first stage is to configure the first device for HA, as described in the following procedure:



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. Connect your PC to the device using a local, direct physical cable connection and then access the Web interface using the default OAMP network address. For more information on initial access, see Assigning the OAMP IP Address on page 35.
 - b. Open the Interface table (**Configuration** tab > **VoIP** menu > **Network** > **IP Interfaces Table**).
 - c. Change the default OAMP network settings to suite your networking scheme.
 - 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 Device and Ethernet Group that is not used by any other network interface. The Ethernet Group is associated with the Ethernet Device assigned to the interface in the 'Underlying Device' field.

The Interface table below shows an example where the Maintenance interface is assigned to Ethernet Device "vlan 2" (which is associated with Ethernet Group "GROUP_2") in the 'Underlying Device' field, while the other interface is assigned to "vlan 1" (associated with "GROUP_1"):

Figure 32-4: 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

- If the connection is through a switch, the packets of both interfaces should generally be untagged. In such a scenario, set the Tagging parameter to **Untagged** for the Ethernet Device that is assigned to the Maintenance interface (see Configuring Underlying Ethernet Devices on page 133). The figure below shows an example (highlighted) where VLAN 2 is configured as the Native VLAN IDs of the Ethernet Group "GROUP_2", by setting the 'Tagging' parameter to **Untagged**:

Figure 32-5: Native VLAN for Ethernet Devices of Maintenance and Other Interfaces

Index	VLAN ID	Underlying Interface	Name	Tagging
0	1	GROUP_1	vlan 1	Untagged
1	2	GROUP_2	vlan 2	Untagged

- 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 133). 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 559.
- Configure the HA parameters in the HA Settings page (**Configuration** tab > **System** menu > **HA Settings**):

Figure 32-6: HA Settings Page

HA Settings

HA Remote Address

10.3.4.61

HA Revertive

Disable

HA Priority

5

Redundant HA Priority

5

- 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. Make sure that you configure different priority levels for the two devices. Typically, you would configure the active device with a higher priority level (number) than the redundant device. The only factor that influences the configuration is which device has the greater number; the actual number is not important. For example, configuring the active with 5 and redundant with 4, or active with 9 and redundant with 2 both assign highest priority to the active device. Configuring the level to 10 does not cause a switchover upon Ethernet connectivity loss. For more information on the feature, see Device Switchover upon Failure on page 556.
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 563 for configuring the second device.

32.1.2.2 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 following procedure 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 561).



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** 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.

32.1.2.3 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".
- Redundant device: The ↔ LED is lit yellow on the E-SBC module.

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".
 - Active device: The ↔ LED is lit green
 - Redundant device: The ↔ LED flashes yellow
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 566.

32.2 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).

32.3 Configuring Firewall Allowed Rules

If you add firewall rules in the Firewall Settings page (see "Configuring Firewall Settings" on page 169) 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 32-7: 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	65535	Any	Disable	None	0	0	0	Block	0

32.4 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 is referred to as *HA Network Reachability*.

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 HA Network Reachability 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).
- If you have configured the HA Network Reachability feature, but the feature is not operational (see note above), the device sends the SNMP trap event, acHANetworkWatchdogStatusAlarm to notify of the situation.
- If a switchover occurs due to no ping reply, the device sends the SNMP trap alarm, acHASystemFaultAlarm to notify of the switchover due to the HA Network Reachability feature.
- For a detailed description of the HA ping parameters, see "HA Parameters" on page 752.

➤ **To configure monitoring of IP entity using pings:**

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

Figure 32-8: 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.

33 HA Maintenance

This section describes HA maintenance procedures.

33.1 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).

33.2 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 561.

33.3 Forcing a Switchover

If required, you can force a switchover between active and redundant units. For more information, see "High Availability Maintenance" on page 579.

33.4 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).

For more information on upgrading the software, see "Software Upgrade Wizard" on page 601.

Part VIII

Maintenance

34 Basic Maintenance

The Maintenance Actions page allows you to perform the following:

- Reset the device - see "Resetting the Device" on page 575
- Lock and unlock the device - see "Locking and Unlocking the Device" on page 577
- Save configuration to the device's flash memory - see "Saving Configuration" on page 578

➤ 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 34-1: 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

34.1 Resetting the Device

The Maintenance Actions page allows you to remotely reset the device. Before resetting the device, you can also choose the following options:

- "Burn" (save) the device's current configuration to the device's flash memory (non-volatile).
- Graceful Shutdown, whereby the device resets only after a user-defined time (i.e., timeout) or after there is no traffic currently processed by the device (the earliest thereof).



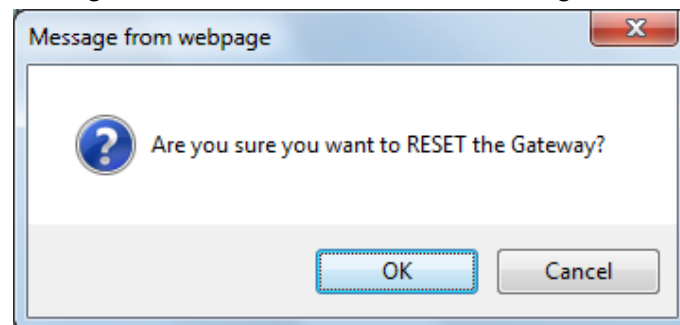
Notes:

- 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 46) 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 47).

➤ **To reset the device:**

1. Open the Maintenance Actions page (see "Basic Maintenance" on page 575).
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. Under the 'Reset Configuration' group, 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 34-2: 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.

34.2 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.

34.3 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 575).
2. Scroll down to the 'LOCK / UNLOCK' group:

Figure 34-3: 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 4) 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.

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 65).

34.4 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** or **Add** buttons 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 575).
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 577).
- 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 575).
- The Home page's General Information pane displays whether the device is currently "burning" the configuration (see "Viewing the Home Page" on page 65).

35 High Availability Maintenance

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

35.1 Initiating an HA Switchover

You can initiate a switchover from the Active unit to Redundant unit.



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 35-1: 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**.

35.2 Resetting the Redundant Unit

You can reset the Redundant unit, if necessary.



Note: When resetting the Redundant unit, the HA mode becomes temporarily unavailable.

➤ **To reset the Redundant unit:**

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 35-2: 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**.

36 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.

37 Software Upgrade

This chapter describes various software update procedures.

37.1 Auxiliary Files

You can install various Auxiliary files on the device. Auxiliary files provide the device with additional configuration settings. The table below lists the different types of Auxiliary files.

Table 37-1: 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 101.
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 586.
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 588.
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 589.
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 591.
AMD Sensitivity	Answer Machine Detector (AMD) Sensitivity file containing the AMD Sensitivity suites. For more information, see AMD Sensitivity File on page 595.

37.1.1 Loading Auxiliary Files

You can load Auxiliary files to the device using one of the following methods:

- Web interface - see "Loading Auxiliary Files through Web Interface" on page 584
- CLI - see Loading Auxiliary Files through CLI on page 585
- TFTP - see "Loading Auxiliary Files through ini File using TFTP" on page 585



Notes:

- You can schedule automatic loading of updated Auxiliary files using HTTP/HTTPS. For more information, see Automatic Update Mechanism.
- Saving Auxiliary files 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 577.
- To delete installed Auxiliary files, see "Viewing Device Information" on page 631.

37.1.1.1 Loading Auxiliary Files through Web Interface

The following procedure describes how to load Auxiliary files through the Web interface.

➤ **To load Auxiliary files through 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 Auxiliary file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name of the file appears 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. Reset the device with a burn-to-flash for your settings to take effect (if you have loaded a Call Progress Tones file).



Note: When loading an *ini* file using the Web interface, Auxiliary files that are already installed on the device are maintained if the ini file does not contain these Auxiliary files.

37.1.1.2 Loading Auxiliary Files through CLI

You can load Auxiliary files from user-defined URLs, using the following CLI commands:

■ **Single Auxiliary file:**

```
# copy <file> from <URL>
```

For example:

```
# copy call_progress_tones from
http://192.169.11.11:80/cpt_us.dat
```

■ **Multiple (batch) Auxiliary files:** The Auxiliary files must be contained in a TAR (Tape ARchive) file (.tar). The TAR file can contain any number and type of Auxiliary files (e.g., Dial Plan file and CPT file).

```
# copy aux-package from | to <URL with TAR file name>
```

For example:

```
# copy aux-package from http://192.169.11.11:80/aux_files.tar
```

For more information on CLI commands, refer to the *CLI Reference Guide*.

37.1.1.3 Loading Auxiliary Files through ini File using TFTP

You can load Auxiliary files to the device through the ini file, using a TFTP server. For more information on Auxiliary ini file parameters, see "Auxiliary and Configuration File Name Parameters" on page 729.

➤ **To load Auxiliary files through ini file:**

1. Create an ini file that includes the names of the Auxiliary files that you want loaded, for example:

```
CallProgressTonesFilename = 'usa_tones_13.dat'
DialPlanFileName = 'dial-plan-us.dat'
```

2. Save the ini file and the Auxiliary files in the same folder on your TFTP server.
3. Reset the device (you can power off and then power on the device); the device loads the ini file and then the Auxiliary files as defined in the ini file, through TFTP.

37.1.2 Deleting Auxiliary Files

You can delete loaded Auxiliary files through the Web interface, as described below.

➤ **To delete a loaded file:**

1. Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**); loaded files are listed under the Loaded Files group, as shown in the example below:

Figure 37-1: Loaded Files Listed on Device Information Page

Loaded Files		
Call Progress Tones File Name:	usa_tones_13.dat	Delete
Loaded Coder Table :	Default CODERTABLE	

2. Click the **Delete** button corresponding to the file that you want to delete; a confirmation message box appears.
3. Click **OK** to confirm deletion.
4. Reset the device with a burn-to-flash for your settings to take effect.

37.1.3 Call Progress Tones File

The Call Progress Tones (CPT) Auxiliary file includes the definitions of the CPT (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, the 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, the parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, the 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
```

37.1.4 Prerecorded Tones File

The Prerecorded Tone (PRT) is a .dat file containing a set of prerecorded tones that can be played by the device. For example, it can be used to play music on hold (MoH) to a call party that has been put on hold. Up to 40 tones (totaling approximately 10 minutes) can be stored in a single PRT file on the device's flash memory.

The PRT file overcomes the limitations of the CPT file such as limited number of predefined tones and limited number of frequency integrations in one tone. If a specific prerecorded tone exists in the PRT file, it overrides the same tone that exists in the CPT file, and is played instead.

You can define a PRT file with multiple tones of the same tone type but with different coders. If one of the tones is defined with the same coder as used in the current call, the device always selects it in order to eliminate the need for using DSP resources. If the coder of the tone is the same as that of the call, DSPs are not required; if they are different, DSPs are required.



Notes:

- The PRT file only generates (plays) tones; detection of tones is according to the CPT file.
- The device does not require DSPs for playing tones from a PRT file if the coder defined for the tone is the same as that used by the current call. If the coders are different, the device uses DSPs.
- The device requires DSPs for local generation of tones.
- For SBC calls, the PRT file supports only the ringback tone and hold tone.

The prerecorded tones can be created using standard third-party, recording utilities such as Adobe Audition, and then combined into a single file (PRT 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 (and other coders)
- Rate: 8 kHz
- Resolution: 8-bit
- Channels: mono

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.

Once created, you need to install the PRT file on the device. This can be done using the Web interface (see "Loading Auxiliary Files" on page 583).

37.1.5 Dial Plan File

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



Note: The Dial Plan described in this section is for backward compatibility purposes only. For the new method, see Configuring Dial Plans on page 511.

37.1.5.1 Creating a Dial Plan File



Note: The Dial Plan described in this section is for backward compatibility purposes only. For the new method, see Configuring Dial Plans on page 511.

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 583.
5. Select the Dial Plan index that you want to use. This depends on the feature and is described in the respective section.

37.1.5.2 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).



Note: The Dial Plan described in this section is for backward compatibility purposes only. For the new method, see Configuring Dial Plans on page 511.

➤ **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 589).
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.

37.1.5.3 Viewing Information of Installed Dial Plan File

You can view information about the Dial Plan file currently installed on the device, through the device's CLI:

- **Viewing Dial Plan file information:** You can view the file name of the installed Dial Plan file and the names of the Dial Plans defined in the Dial Plan file, by entering the following CLI command (in Enable mode):

```
# debug auxiliary-files dial-plan info
```

For example, the following shows the file name of the installed Dial Plan file and lists its Dial Plans:

```
# debug auxiliary-files dial-plan info
File Name: MyDialPlan.txt
Plans:
Plan #0 = PLAN1
Plan #1 = PLAN2
```

Note that the index number of the first Dial Plan is 0.

- **Searching a prefix number:** You can check whether a specific prefix number is defined in a specific Dial Plan (and view the corresponding tag if the Dial Plan implements tags), by entering the following CLI command (in Enable mode):

```
# debug auxiliary-files dial-plan match-number <Dial Plan
number> <prefix number>
```

For example, the following checks whether the called prefix number 2000 is defined in Dial Plan 1, which is used for obtaining the destination IP address (tag):

```
# debug auxiliary-files dial-plan match-number PLAN1 2000
Match found for 4 digits
Matched prefix: 2000
Tag: 10.33.45.92
```

37.1.6 User Information File

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

37.1.6.1 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 following procedure.

➤ **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 (EnableUserInfoUsage) to **Enable**.
3. Save this setting to the device with a reset for the setting to take effect.

37.1.6.2 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 through Web Interface" on page 592

- CLI - see Configuring SBC User Info Table through CLI on page 593
- Loadable User Info file - see "Configuring SBC User Info Table in Loadable Text File" on page 594

37.1.6.2.1 Configuring SBC User Info Table through Web Interface

The following procedure describes how to configure the SBC User Info table through the Web interface.



Note: If you load any User Info file 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 through the Web interface:

1. Open the SBC User Info table (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **User Information** > **SBC User Info Table**).
2. Click **Add**; the following dialog box appears:

Figure 37-2: SBC User Info Table - Add Row Dialog Box

3. Configure the SBC User Info table parameters according to the table below.
4. Click **Add**.
5. To save the changes to flash memory, see "Saving Configuration" on page 578.

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 37-2: SBC User Info Table Parameter Descriptions

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

Parameter	Description
Username [SBCUserInfoTable_Username]	Defines the username for registering the user when authentication is necessary. The valid value is a string of up to 40 characters.
Password [SBCUserInfoTable_Password]	Defines the password for registering the user when authentication is necessary. The valid value is a string of up to 20 characters.
IP Group [SBCUserInfoTable_IPGroupUpName]	Assigns an IP Group to the user and is used as the Request-URI source host part for the AOR in the database. For configuring IP Groups, see "Configuring IP Groups" on page 348.
Status [SBCUserInfoTable_Status]	(Read-only field) Displays the status of the user - "Registered" or "Not Registered".

37.1.6.2.2 Configuring SBC User Info Table through 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
```

37.1.6.2.3 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 583
- *ini* file, using the *UserInfoFileName* parameter - see "Auxiliary and Configuration File Name Parameters" on page 729
- 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
```

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:

- Make sure that there are no spaces between the values.
- 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.

Below is an example of a configured User Info file:

```
[ SBC ]
FORMAT LocalUser,UserName,Password,IPGroupID
john,john_user,john_pass,2
sue,sue_user,sue_pass,1
```

37.1.6.3 Viewing the Installed User Info File Name

You can view the name of the User Info file currently installed on the device, through the device's CLI (in Enable mode):

```
# debug auxiliary-files user-info info
```

For example:

```
# debug auxiliary-files user-info info
User Info File Name MyUsers.txt
```

37.1.7 AMD Sensitivity File

The device is shipped with a default, pre-installed *AMD Sensitivity* file for its Answering Machine Detection (AMD) feature. This file includes the detection algorithms for detecting whether a human or answering machine has answered the call, and is based on North American English. In most cases, the detection algorithms in this file suffice even when your deployment is in a region where a language other than English is spoken. However, if you wish to replace the default file with a different AMD Sensitivity file containing customized detection algorithms, please contact your AudioCodes sales representative for more information.

The AMD Sensitivity file is created in .xml format and then converted to a binary .dat file that can be installed on the device. The XML-to-binary format conversion can be done using AudioCodes DConvert utility. For more information on using this utility, refer to *DConvert Utility User's Guide*. Only one AMD Sensitivity file can be installed on the device. To install a new AMD Sensitivity file, use any of the following methods:

- Web interface: On the Load Auxiliary Files page - see "Loading Auxiliary Files" on page 583.
- TFTP during initialization: You need to configure the *ini* file parameter, *AMDSensitivityFileName*, and then copy the AMD Sensitivity file to the TFTP directory.
- Automatic Update feature: For more information, see Automatic Update Mechanism. For this method, the *AMDSensitivityFileUrl* parameter must be set through SNMP or *ini* file.

For more information on the AMD feature, see "Answering Machine Detection (AMD)" on page 206.

37.2 Software License Key

The License Key determines the device's supported features and call capacity, as ordered from your AudioCodes sales representative. You can upgrade or change your device's supported features and capacity, by purchasing and installing a new License Key that match your requirements.



Note: The availability of certain Web pages depends on the installed License Key.

37.2.1 Viewing the License Key

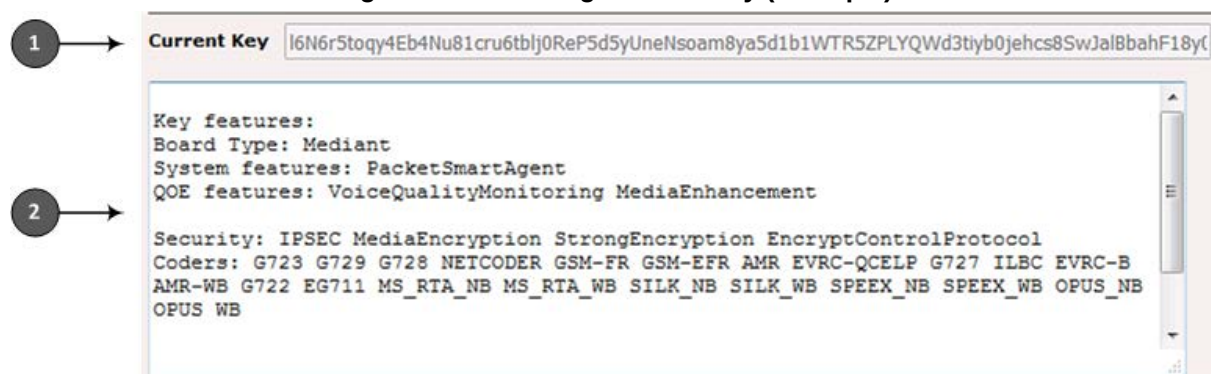
The following procedure describes how to view the device's License Key.

➤ **To view the License Key:**

- Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** folder > **Software Upgrade Key**).

The License Key is displayed in encrypted-string format in the 'Current Key' field (1) and the main features provided by the License Key are displayed in the pane (2) below it, as shown in the example below:

Figure 37-3: Viewing License Key (Example)



37.2.2 Installing a New Software License Key

This section describes how to install a new License Key on the device.



Note: When you install a new License Key, it overwrites the previously installed License Key. Any license-based features that were included in the old License Key, but not included in the new License Key, will no longer be available.

37.2.2.1 Installing Software License Key through Web Interface

The following procedure describes how to install the Software License Key through the Web interface.

➤ **To install the Software License Key through 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 re-load this backup to restore the device's original capabilities.
 - a. In the 'Current Key' field, select the entire text string and copy it to any standard text file (e.g., Notepad):

Figure 37-4: Current Key Field

- b. Save the text file with any file name and file extension (e.g., key.txt) to a folder on your computer.
3. If the device is operating in High-Availability (HA) mode, load the License Key as follows (otherwise, skip this step):
 - a. Under the 'Load Upgrade Key file...' text, click the **Browse** button, and then navigate to and select the License Key file on your computer:

Figure 37-5: Load File Button

- b. Click **Load File**; the new License Key is installed on the device and saved to flash memory. The License Key is displayed in the 'Current Key' field.



Note: The License Key file for HA includes two License Keys - one for the active device and one for the redundant device. Each License Key has a different serial number ("S/N").

4. (For a non-HA standalone device only) Load the License Key as follows:
 - a. Open the License Key file using a text-based program such as Notepad.
 - b. Copy-and-paste the contents of the file into the 'Add a Software Upgrade Key' field.
 - c. Click **Add Key**.

Figure 37-6: Add Key Button

5. 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"

6. 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.

37.2.2.2 Installing Software License Key through CLI

To install the Software License Key through 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
```

37.2.3 Viewing the Device's Product Key

The Product Key identifies a specific purchase of your device installation for the purpose of subsequent communication with AudioCodes (e.g., for support and software upgrades). The Product Key is your chassis' serial number--"S/N(Product Key)"--which also appears on the product label affixed to the chassis.

The Product Key is included in the License Key. Once the License Key is installed, you can view the Product Key in the following Web pages:

- Software Upgrade Key Status page (**Maintenance** tab > **Software Update** folder > **Software Upgrade Key**). The Product Key is displayed in the read-only 'Product Key' field, as shown in the example below:

Figure 37-7: Viewing Product Key

Product Key	1352798076accedd

- Device Information page (see Viewing Device Information on page 631).

37.3 Upgrading SBC Capacity Licenses by License Pool Manager Server

The device can receive SBC capacity licenses from a centralized pool of SBC resources managed by the License Pool Manager Server running on AudioCodes EMS. The License Pool Manager Server can dynamically allocate and de-allocate SBC capacity licenses from the pool to devices in the network to meet capacity demands of each device whenever required. The License Pool Manager Server holds a pool of customer-ordered SBC capacity (resource) licenses, which can include any of the following license types:

- SBC sessions (media and signaling)
- SBC signaling sessions

- SBC transcoding sessions
- SBC registrations (number of SIP endpoints that can register with the SBC)

Therefore, the device can be upgraded by the License Pool Manager Server with any of the above SBC license types.

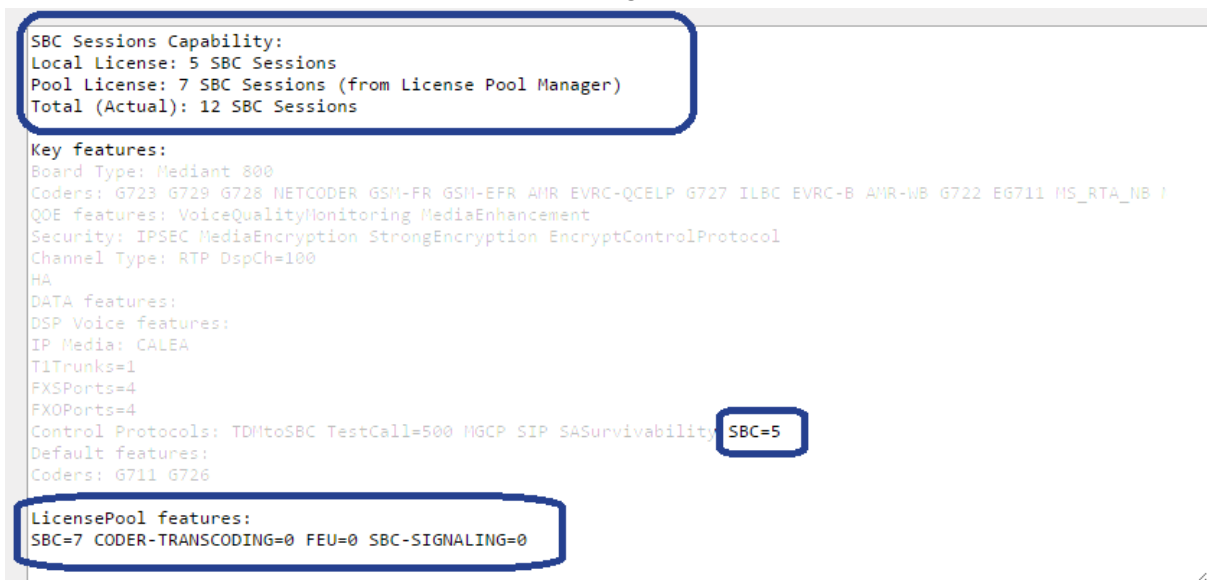
Communication between the device and License Pool Manager Server is through HTTPS (port 443) and SNMP. If a firewall exists in the network, ensure that ports for these applications are opened. The device periodically checks with the License Pool Manager Server for SBC capacity licenses. The License Pool Manager Server identifies the device by serial number. If it has an SBC license for the device, it sends it to the device. If the device's installed Software Feature Key already includes SBC capacity figures, the SBC license allocated from the pool is simply added to it (but up to the device's maximum supported capacity capabilities). A device reset is required for the allocated SBC license to take effect.

The Web interface's Software Upgrade Key Status page (**Maintenance** tab > **Software Update** menu > **Software Upgrade Key**) indicates the SBC license allocated by the License Pool Manager Server:

- "Local License": Number of SBC sessions according to the installed Software Feature Key file. The actual license is indicated on the page in the "SBC=" field (e.g., SBC=5, as shown in the example figure below).
- "Pool License": Number of SBC sessions allocated by the License Pool Manager Server.
- "Total (Actual)": Total number of SBC sessions permitted on the device based on the installed Software Feature Key file and the SBC sessions allocated by the License Pool Manager Server.
- "LicensePool features":
 - "SBC": Number of SBC sessions (media and signaling) allocated by the License Pool Manager Server.
 - "CODER-TRANSCODING": Number of SBC transcoding sessions allocated by the License Pool Manager Server.
 - "FEU": Number of SBC registrations allocated by the License Pool Manager Server.
 - "SBC-SIGNALING": Number of SBC signaling sessions allocated by the License Pool Manager Server.

The following displays an example of the indication of SBC licenses allocated by the License Pool Manager Server in the Software Upgrade Key Status page:

Figure 37-8: Software Upgrade Key Status Page Displaying SBC Licenses from License Pool Manager



If communication with the License Pool Manager Server is lost for a long duration, the device discards the allocated SBC license (i.e., expires) and resets with its initial, "local" SBC license. This mechanism prevents misuse of SBC licenses allocated by the License Pool Manager Server.

The following SNMP alarms relate to the allocation/de-allocation of SBC licenses by the License Pool Manager Server:

- acLicensePoolInfraAlarm (1.3.6.1.4.1.5003.9.10.1.21.2.0.106):
 - Sent when the device receives a new SBC license from the License Pool Manager Server and a device reset is required.
 - Sent when the device is unable to access the License Pool Manager Server.
 - Sent when the SBC license allocated by the License Pool Manager Server is about to expire (e.g., when communication with the License Pool Manager Server is lost)
- acLicensePoolApplicationAlarm (1.3.6.1.4.1.5003.9.10.1.21.2.0.107):
 - Sent when the device receives an SBC license from the License Pool Manager Server that exceeds the maximum SBC session capacity that can be supported by the device.
 - Sent when the device resets with an SBC license allocated by the License Pool Manager Server that exceeds the maximum SBC session capacity that can be supported by the device. The device sets the capacity to its maximum (and values beyond the device's capability are not applied)

**Notes:**

- No configuration is required on the device; the License Pool Manager Server controls the allocation/de-allocation of its resource pool to the managed devices. For more information on the License Pool Manager Server, refer to the *EMS User's Manual*.
- The allocation/de-allocation of SBC licenses to the device by the License Pool Manager Server is service affecting and requires a device reset.
- For HA systems, the License Pool Manager Server automatically allocates an equal number of SBC licenses (sessions) to both the active and redundant devices. For example, if the License Pool Manager Server allocates 200 sessions to the active device, it also allocates 200 to the redundant. Thus, it is important to take this into consideration when ordering a license pool.
- If the device is restored to factory defaults, the SBC license allocated by the License Pool Manager Server is deleted.
- If the device is allocated an SBC license by the License Pool Manager Server that exceeds the maximum number of sessions that it can support, the device sets the number of sessions to its maximum supported

37.4 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 Hitless Upgrade method operates 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 <http://www.audiocodes.com/downloads>.
- 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.
- To upgrade the device from any earlier version to 7.0, do the following:
 - 1) Delete core dumps from the redundant device through CLI (Telnet). Core dump deletion can take up to 10 minutes.
 - 2) Perform a manual switchover from active to redundant.
 - 3) When the system is operational again, delete core dumps from the current redundant device through CLI (Telnet). Core dump deletion can take up to 10 minutes.
 - 4) Start the Hitless Software Upgrade procedure.
- Instead of manually upgrading the device, you can use the device's Automatic Update feature for automatic provisioning (see "Automatic Provisioning" on page 609).

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>

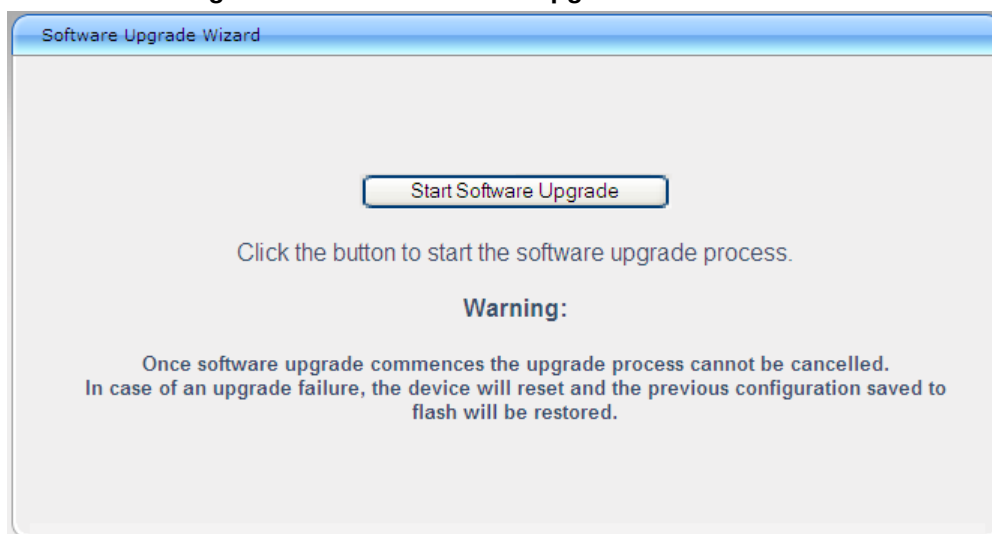
• Non-Hitless Software Upgrade:

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

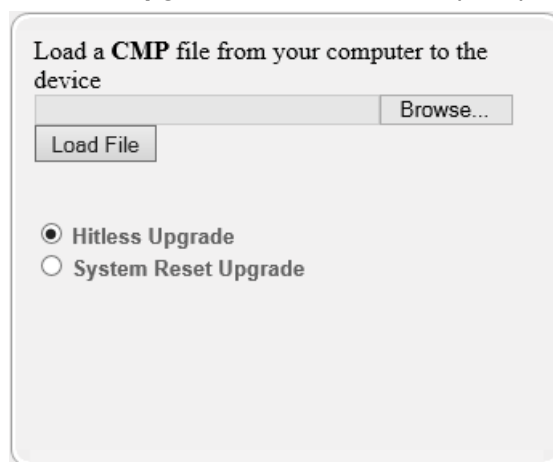
➤ **To upgrade the device using the Software Upgrade Wizard:**

1. Make sure that you have installed a new Software License Key (see "Software License Key" on page 596) 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 577). 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 607.
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 37-9: Start Software Upgrade Wizard Screen



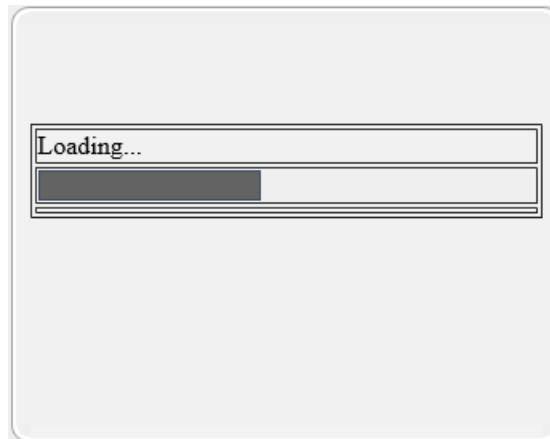
5. Click **Start Software Upgrade**; the wizard starts, prompting you to load a .cmp file:





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.

6. Click **Browse**, and then navigate to where the .cmp file is located on your computer. Select the file, and then click **Open**.
7. 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:



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.

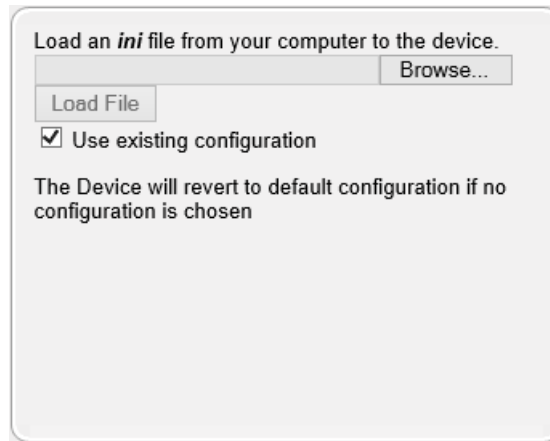


Note: Device reset may take a few minutes (even up to 30 minutes), depending on cmp file version.

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.

- **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).

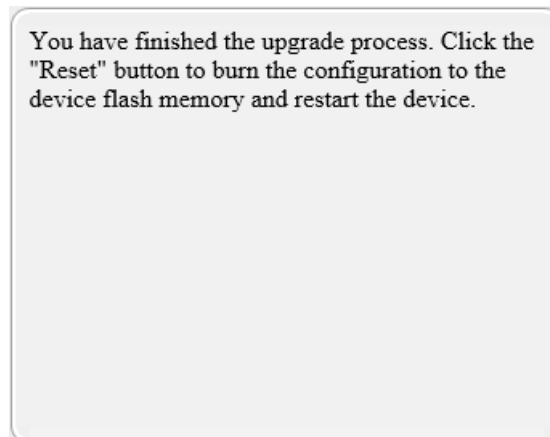
Figure 37-10: Software Upgrade Wizard - Load INI File



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.

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 37-11: Software Upgrade Wizard - Files Loaded



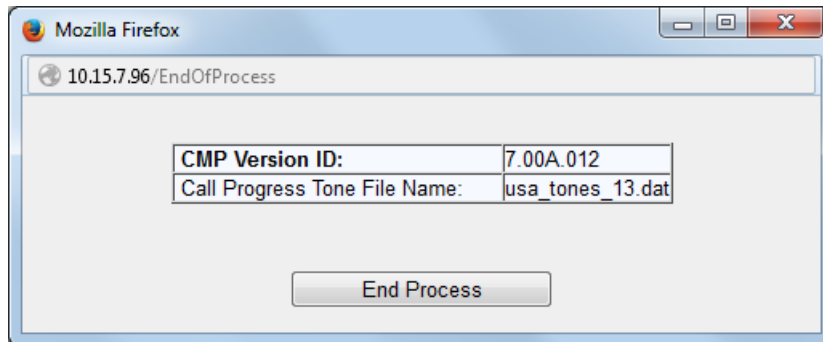
13. Click **Reset**  to burn the files to the device's flash memory; the "Burn and reset in progress" message is displayed and the device 'burns' the newly loaded files to flash memory and then resets.



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 37-12: Software Upgrade Process Completed Successfully (Example)



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.

38 Backing Up and Loading Configuration File

You can save a copy of the device's current configuration settings as a file on a local PC (ini file), remote server. This can be used as a backup file for your configuration. If needed, you can load the file to the device at a later stage to restore your configuration settings. The saved file includes only parameters that were modified and parameters with other than default values.

You can also save (create) the current configuration as a configuration file on the device's flash memory and send it 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
```



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 38-1: 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.

39 Automatic Provisioning

This chapter describes the device's automatic provisioning mechanisms.

39.1 Automatic Configuration Methods

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

Table 39-1: 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

39.1.1 DHCP-based Provisioning

A third-party DHCP server can be configured to automatically provide each device, acting as a DHCP client, 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" on page 740.

➤ **To enable the device as a DHCP client:**

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

Figure 39-1: Enabling DHCP - Application Settings Page

The screenshot shows the 'DHCP Settings' section of the Application Settings page. It features a dropdown menu labeled 'Enable DHCP' which is currently set to 'Enable'. There is a blue circular icon with a pencil inside a square button to the right of the dropdown.

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

39.1.2 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 609 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 612.

39.1.3 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 610 is that the protocol in the URL is "ftp" (instead of "http").

39.1.4 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.

39.2 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.



Warning: 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.



Notes:

- For a description of all the Automatic Update parameters, see "Automatic Update Parameters" on page 731 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>.

39.2.1 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)

39.2.2 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" on page 731. 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'
FeatureKeyURL = 'https://www.company.com/License_Key.txt'
AutoCmpFileUrl = 'http://www.corp.com/SIP_F7.00A.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)# feature-key
http://www.company.com/License_Key.txt
(automatic-update)# call-progress-tones http://www.company.com/call_progress.dat
(automatic-update)# auto-firmware http://www.company.com/SIP_F7.00A.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 Placeholder in Configuration File Name" on page 618.

39.2.3 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.

39.2.4 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.

39.2.5 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 614). 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 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 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., "7.00.200.001"
- **<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:

```
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;7.00.200.001(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.

You can also enable the device to run a CRC on the downloaded configuration file (ini) to determine whether the file has changed in comparison to the previously downloaded file. Depending on the CRC result, the device can install or discard the downloaded file. For more information, see "Cyclic Redundancy Check on Downloaded Configuration Files" on page 618.



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., License Key, CPT and Dial Plan) 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'
FeatureKeyURL =
'https://www.company.com/License_Key.txt'
```

- ◆ 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.
- After the device downloads the License Key file (FeatureKeyURL), it checks that the serial number in the file ("S/N <serial number>") is the same as that of the device. If the serial number is the same and the license key is different to the one currently installed on the device, it applies the new License Key. For devices in HA mode, the License Key is applied to both active and redundant units.

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.

39.2.6 File Download Sequence

Whenever the Automatic Update feature is triggered (see "Triggers for Automatic Update" on page 613), 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 the 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.

39.2.7 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" on page 731.

39.2.8 MAC Address Placeholder 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 612, 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`).

39.2.9 File Template for Automatic Provisioning

To facilitate automatic provisioning setup, you can use a single template to define the files to download during automatic provisioning. The template uses special keywords to denote the different file types to download and in the URL address of the provisioning server it uses a placeholder for the file names which is replaced by hardcoded file names and extensions according to file type, as described in more detail below.



Note:

- Unlike the parameters that define specific URLs for Auxiliary files (e.g., CptFileURL), the file template feature always retains the URLs after each automatic update process. Therefore, with the file template the device always attempts to download the files upon each automatic update process.
- If you configure a parameter that defines a URL for a specific file (e.g., CptFileURL), the settings of the file template (TemplateUrl parameter) is ignored for the specific file type (e.g., CPT file).
- Additional placeholders can be used in the file name in the URL, for example, <MAC> for MAC address (see MAC Address Placeholder in Configuration File Name on page 618).

➤ **To use a file template for automatic provisioning:**

1. Define the file **types** to download by the file template, using the AupdFilesList parameter. Use the keywords listed in the table below to specify each file type. For example, to specify ini, Feature Key, and CPT files:

- ini File:

```
AupdFilesList = 'ini', 'fk', 'cpt'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# template-files-list ini,fk,cpt
```

2. Define the URL address of the provisioning server on which the files (specified in Step 1) are located, using the TemplateUrl parameter. When you configure the URL, you must include the file type placeholder, "<FILE>", which represents the file name. For each file type specified in Step 1, the device sends an HTTP request to the server, where the placeholder in the URL is replaced with the filename and extension, as listed in the below table. For example, if you configure the AupdFilesList parameter as in Step 1 and the TemplateUrl parameter to:

- ini File:

```
TemplateUrl = 'http://10.8.8.20/Site1_<FILE>'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# template-url http://10.8.8.20/Site1_<FILE>
```

The device sends HTTP requests to the following URLs:

- http://10.8.8.20/Site1_**device.ini**
- http://10.8.8.20/Site1_**fk.ini**
- http://10.8.8.20/Site1_**cpt.data**

3. Place the files to download on the provisioning server. Make sure that their file names and extensions are based on the hardcoded string values specific to the file type for the <FILE> placeholder (e.g., "Site1_device.ini" for the ini file), as shown in the table below.

File Template Keywords and Placeholder Values per File Type

File Type	Keywords for Template File	Value Replacing <FILE> Placeholder
ini file	ini	device.ini
CMP file based on timestamp	acmp	autoFirmware.cmp
User Info file	usrinf	userInfo.txt
CMP file	cmp	firmware.cmp
Feature Key file	fk	fk.ini
Call Progress Tone (CPT) file	cpt	cpt.dat
Prerecorded Tones (PRT) file	prt	prt.dat
Dial Plan file	dpln	dialPlan.dat
Answering Machine Detection (AMD) file	amd	amd.dat
SSL/TLS Private Key file	sslp	pkey.pem pkey<ID>.pem (for multi-certificate system)
SSL/TLS Root Certificate file	sslr	root.pem root<ID>.pem (for multi-certificate system)
SSL/TLS Certificate file	sslc	cert.pem cert<ID>.pem (for multi-certificate system)

39.2.10 Automatic Update Configuration Examples

This section provides a few examples on configuring the Automatic Update feature.

39.2.10.1 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.

39.2.10.2 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 configuration file (ini).
 - 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:

- ◆ ini File:
VPFileUrl =
'ftps://root:wheel@ftpserver.corp.com/vp.dat'
- ◆ CLI:
configure system
(config-system)# automatic update
(automatic-update)# voice-prompts
'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 configuration files to the server.
- b. Configure the device with the URL paths of the .cmp and ini files:

- ◆ ini File:

```
AutoCmpFileUrl =
'http://www.company.com/device/sw.cmp'
IniFileURL = 'http://www.company.com/device/inifile.ini'
```

- ◆ CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# auto-firmware 'http://www.company.com/sw.cmp'
```

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):

- ini File:

```
AutoUpdatePredefinedTime = '03:00'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# predefined-time 03:00
```

39.2.10.3 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 Placeholder in Configuration File Name" on page 618.
- 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 a "master" configuration file template named "master_configuration.ini" with the following settings:
 - Common configuration for all devices:
 - ◆ ini file:


```
AutoUpdatePredefinedTime = '24:00'
CptFileURL = 'https://www.company.com/call_progress.dat'
AutoCmpFileUrl = 'https://www.company.com/sw.cmp'
```
 - ◆ CLI:


```
# configure system
(config-system)# automatic update
(automatic-update)# update-frequency 24:00
(automatic-update)# call-progress-tones
https://www.company.com/call_progress.dat
(automatic-update)# auto-firmware https://www.company.com/sw.cmp
```
 - Configuration per device based on MAC address:
 - ◆ ini file:


```
IniFileURL = 'http://www.company.com/config_<MAC>.ini'
```
 - ◆ CLI:


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

(automatic-update)# voice-configuration
http://www.company.com/config_<MAC>.ini
```
2. Copy the master configuration file that you created in Step 1 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 file:

◆ 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.

This page is intentionally left blank.

40 Restoring Factory Defaults

This section describes the different ways that you can restore the device's configuration to factory defaults.

40.1 Restoring Factory Defaults through CLI

You can restore the device to factory defaults through CLI, as described in the following procedure.

➤ **To restore factory defaults through 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
```

40.2 Restoring Factory Defaults through Web Interface

You can restore the device to factory defaults through the Web interface.

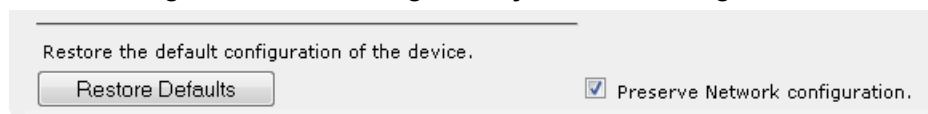


Note: When restoring to factory defaults, you can preserve your IP network settings that are configured in the Interface table (see "Configuring IP Network Interfaces" on page 138), as described in the procedure below. This may be important, for example, to maintain connectivity with the device (through the OAMP interface) after factory defaults have been applied.

➤ To restore factory defaults through Web interface:

1. Open the Configuration File page:
 - Toolbar: From the **Device Actions** drop-down list, choose **Restore Defaults**
 - Navigation Tree: **Maintenance** tab > **Software Update** > **Configuration File**

Figure 40-1: Restoring Factory Defaults through Web



2. To keep your current IP network settings, select the **Preserve Network Configuration** check box. To overwrite all your IP network settings with the default IP network interface, clear the **Preserve Network Configuration** check box.
3. Click the **Restore Defaults** button; a message appears requesting you to confirm.
4. Click **OK** to confirm or **Cancel** to return to the page.
5. Once the device is restored to factory defaults, reset the device for the settings to take effect.

40.3 Restoring Defaults through ini File

You can restore the device to factory defaults 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 607). If the *ini* file does include content (e.g., parameters), ensure that they are on lines beginning with comment signs (i.e., semicolons ";") so that the device ignores them.



Note: The only settings that are not restored to default are the management (OAMP) LAN IP address and the Web interface's login user name and password.

Part IX

Status, Performance Monitoring and Reporting

41 System Status

This section describes how to view various system statuses.

41.1 Viewing Device Information

The Device Information page displays hardware and software information about the device.

➤ **To access the Device Information page:**

- Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**).

▼ General Settings	
MAC Address:	00908f3b463f
Serial Number:	3884606
Board Type:	Mediant 4000
Device Up Time:	1d:22h:23m:26s:33th
Device Administrative State:	Unlocked
Device Operational State:	Enabled
Flash Size [Mbytes]:	252
RAM Size [Mbytes]:	2048
CPU Speed [MHz]:	1250
▼ Versions	
Version ID:	7.00A.011.009
DSP Type:	1
DSP Software Version:	70032
DSP Software Name:	5039AE3_R
Flash Version:	560
▼ Loaded Files	
Loaded Call Progress Tones:	Default Progress Tones

Device Information Description

Parameter	Description
General Settings	
MAC Address	Media access control (MAC) address.
Serial Number	Serial number of the CPU. This serial number also appears on the product label that is affixed to the chassis, as "CPU S/N".
Product Key	Product Key, which identifies the specific device purchase. The Product Key also appears on the product label that is affixed to the chassis, as "S/N(Product Key)". For more information, see Viewing the Device's Product Key on page 598.
Board Type	Product name of the device.
Device Up Time	Duration that the device has been up and running since the last reset. The duration is displayed in the following format: <i>dd:hh:mm:ss:100th of a second</i>
Device Administrative State	Administrative status ("Unlocked" or "Locked"), as performed in Locking and Unlocking the Device on page 577.

Parameter	Description
Device Operational State	Operational status: "Disabled" "Enabled" "Error" "Unknown"
Flash Size [Mbytes]	Size of the non-volatile storage memory (flash), measured in megabytes.
RAM Size [Mbytes]	Size of the random access memory (RAM), measured in megabytes.
CPU Speed [MHz]	Clock speed of the CPU, measured in megahertz (MHz).
Versions	
Version ID	Software version number.
DSP Type	Type of DSP.
DSP Software Version	DSP software version.
DSP Software Name	DSP software name.
Flash Version	Flash memory version number.
Loaded Files: Displays installed Auxiliary files. You can also delete a file, by clicking the corresponding Delete button, as described in Deleting Auxiliary Files on page 585.	

41.2 Viewing Ethernet Port Information

The Ethernet Port Information page displays read-only information about the Ethernet Port Group connections.



Note: If the device is operating in High-Availability mode, you can also view Ethernet port information of the redundant device, by opening the Redundant Ethernet Port Information page (Status & Diagnostics tab > System Status menu > Redundant Ethernet Port Info).

➤ 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 65)

	Port Name	Active	Speed	Duplex Mode	State	Group Member
1	GE_1	Yes	1 Gbps	Full Duplex	Forwarding	GROUP_1
2	GE_2	No	10 Mbps	Half Duplex	Disabled	GROUP_1
3	GE_3	No	10 Mbps	Half Duplex	Disabled	GROUP_2
4	GE_4	Yes	1 Gbps	Full Duplex	Forwarding	GROUP_2
5	GE_5	No	10 Mbps	Half Duplex	Forwarding	GROUP_3
6	GE_6	No	10 Mbps	Half Duplex	Disabled	GROUP_3
7	GE_7	No	10 Mbps	Half Duplex	Forwarding	GROUP_4
8	GE_8	No	10 Mbps	Half Duplex	Disabled	GROUP_4

Table 41-1: Ethernet Port Information Parameters

Parameter	Description
Port Name	Displays the name of the port.
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.

41.3 Viewing Hardware Components Status

The Components Status page provides read-only, real-time status of the device's chassis components such as slot occupants, fans, and power supply units.



Note: You can also access this page from the Home page (see "Viewing the Home Page" on page 65).

- To view the status of the device's hardware components:
 - Open the Components Status page (**Status & Diagnostics** tab > **System Status** menu > **Components Status**).

Figure 41-1: Components Status Page

Active Board Components Status	
CPU Slot Number	3
Fan	Exists
Power Supply no.1	Does not exist
Power Supply no.2	No Alarm
Redundant Board Components Status	
CPU Slot Number	3
Fan	Exists
Power Supply no.1	Exists
Power Supply no.2	Does not exist

41.4 Reporting DSP Utilization through SNMP MIB

You can obtain information on the percentage of DSP resources utilized by the device, through the SNMP MIB table, acPMDSPUsage. You can also configure low and high DSP utilization thresholds for this MIB, that if crossed, the SNMP trap event, acPerformanceMonitoringThresholdCrossing is sent by the device. For more information on this MIB, refer to the *SNMP Reference Guide*.

This page is intentionally left blank.

42 Carrier-Grade Alarms

This section describes how to view SNMP alarms raised by the device.

42.1 Viewing Active Alarms

The Active Alarms table displays a list of currently active alarms that have been raised by the device. Once an alarm has been resolved (cleared), the device moves it into the History Alarms table (see "Viewing History Alarms" on page 637). For detailed information on SNMP alarms, refer to the *SNMP Reference Guide* document.



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.

➤ **To view active alarms:**

- Open the Active Alarms table (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Active Alarms**). You can also access the table from the Home page (see "Viewing the Home Page" on page 65).

Figure 42-1: Active Alarms Page

Sequential number	Severity	Source	Description	Date
2	Major	Chassis#0/PowerSupply#1	Power-Supply Alarm. Power-Supply is missing	17.3.2015 , 08:13:46
5	Minor	Board#1/EthernetLink#2	Ethernet link alarm. LAN port number 2 is down.	17.3.2015 , 08:13:55
6	Minor	Board#1/EthernetLink#3	Ethernet link alarm. LAN port number 3 is down.	17.3.2015 , 08:13:55
8	Minor	Board#1/EthernetLink#5	Ethernet link alarm. LAN port number 5 is down.	17.3.2015 , 08:13:55
9	Minor	Board#1/EthernetLink#6	Ethernet link alarm. LAN port number 6 is down.	17.3.2015 , 08:13:55
10	Minor	Board#1/EthernetLink#7	Ethernet link alarm. LAN port number 7 is down.	17.3.2015 , 08:13:55
11	Minor	Board#1/EthernetLink#8	Ethernet link alarm. LAN port number 8 is down.	17.3.2015 , 08:13:55

For each alarm, the following information is provided:

- **Sequential Number:** number of the alarm (sequential numbering of each alarm)
- **Severity:** severity level of the alarm:
 - Critical (red)
 - Major (orange)
 - Minor (yellow)
- **Source:** device component from which the alarm was raised
- **Description:** brief explanation of the reason 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.

42.2 Viewing History Alarms

The Alarms History table displays a list of alarms that have been cleared (resolved). You can configure the maximum number of alarms displayed in the table, using the AlarmHistoryTableMaxSize ini file parameter. If the maximum is reached and a new alarm is added to the table, the oldest alarm is removed from the table to accommodate the new alarm.

➤ **To view history alarms:**

- Open the Alarms History table (**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 (orange)
 - 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.

43 Performance Monitoring

This section describes how to view performance monitoring in the device's Web interface.

43.1 Viewing MOS per Media Realm

The MOS Per Media Realm page displays statistics on Media Realms (configured in "Configuring Media Realms" on page 323). 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 43-1: 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.

43.2 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).

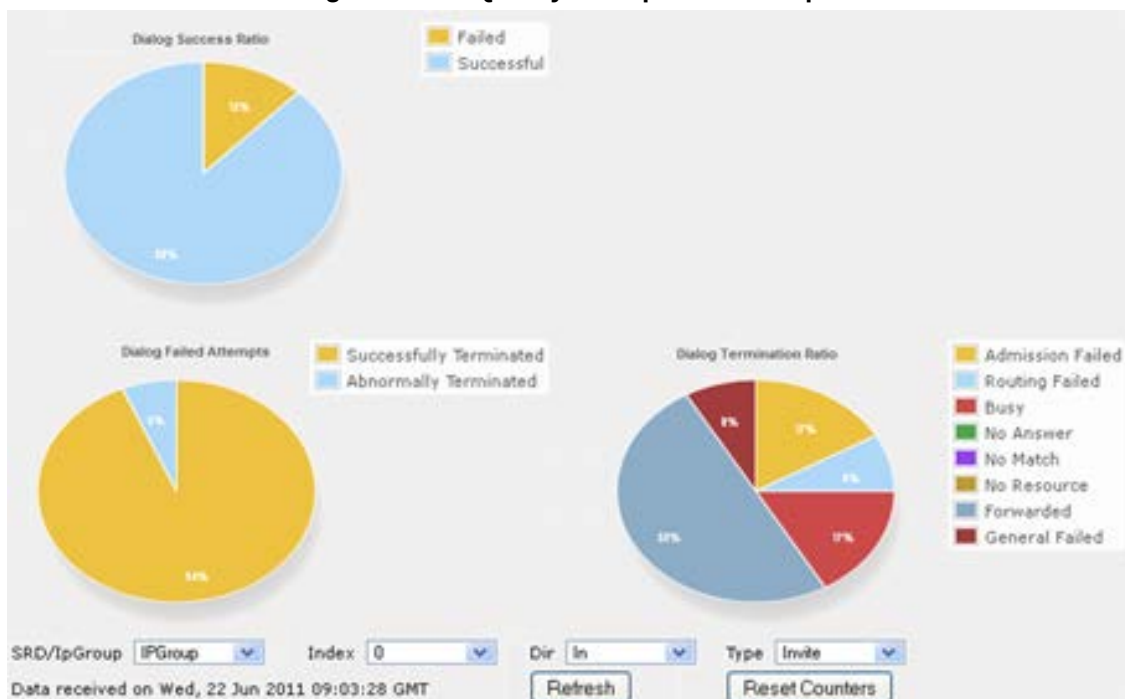
This page provides three pie charts:

- 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 43-2: 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**.

43.3 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 43-3: 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.

44 VoIP Status

This section describes how to view VoIP status and statistics.

44.1 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 138).

➤ **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

44.2 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 135.

➤ **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

44.3 Viewing Static Routes Status

The IP Routing Status table displays the status of the static routes. These are routes configured in the Static Route table (see "Configuring Static IP Routing" on page 146) 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 (**Status & Diagnostics** tab > **VoIP Status** menu > **Static Route Status**).

Figure 44-1: 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	

44.4 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 10 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 44-2: 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>

44.5 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 369).

➤ **To view registration status:**

- Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registration Status**).

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 served IP Group, if applicable
 - ◆ **Status:** "Registered" or "Unregistered"

44.6 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 360.

➤ **To view Proxy Set status:**

- Open the Active Proxy Set Status page (**Monitor** menu > **Monitor** tab > **VoIP Status** folder > **Proxy Sets Status**).

Figure 44-3: 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 44-1: Proxy Sets Status Table Description

Parameter	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 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.

Parameter	Description
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.

45 Reporting Information to External Party

This section describes features for reporting various information to an external party.

45.1 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 RTCP XR feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 596.
- 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. You can configure the device to send RTCP XR reports to an IP Group, 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 45-1: 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
	Packet Loss Concealment

Group	Metric Name
	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

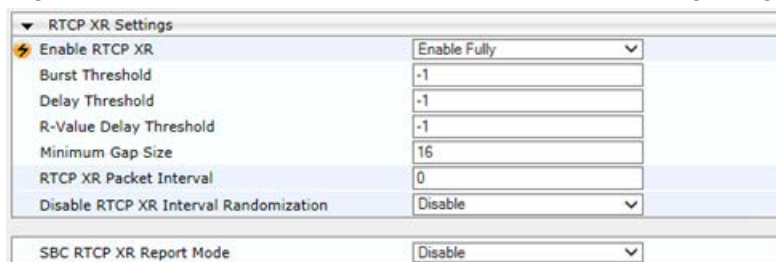
Below shows an example of a SIP PUBLISH message sent with RTCP XR and QoE information:

```
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.227.005
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 45-1: 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
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. Using the *PublicationIPGroupID* ini file parameter, define the IP Group to where you want to send the RTCP XR.
5. Click **Submit**, and then reset the device with a save ("burn") for your settings to take effect.

45.2 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 can send the CDRs to any of the following:

- Syslog server. The CDR Syslog message complies with RFC 3164 and is identified by Facility 17 (local1) and Severity 6 (Informational).
- RADIUS server. For CDR in RADIUS format, see "Configuring RADIUS Accounting" on page 667. For configuring RADIUS servers for CDR reporting, see "Configuring RADIUS Servers" on page 233.

45.2.1 CDR Field Description

This section describes the default CDR fields that are generated by the device.



Note: You can customize the default CDR fields if desired. For customizing SBC-related CDRs, see Customizing CDRs for SBC Calls on page 660.

45.2.1.1 CDR Fields for SBC Signaling

The default 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 45-2: Default 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
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

CDR Field Name	Description	Format
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_FORWARD" ▪ GENERAL_FAILED: Any other reason Calls with duration: ▪ NORMAL_CALL_CLEAR: ✓ "GWAPP_NORMAL_CALL_CLEAR" ✓ ABNORMALLY_TERMINATED: Anything else N/A - Reasons not belonging to above categories.	String of up to 17 characters
SetupTime	Call setup 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

CDR Field Name	Description	Format
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

CDR Field Name	Description	Format
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
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 |
```

45.2.1.2 CDR Fields for SBC Media

The default 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 45-3: Default CDR Fields for SBC Media

CDR Field Name	Description
MediaReportType	Report type (media start, update, or end)
SIPCallId	Unique call ID
Cid	Channel CID
MediaType	Media type (audio, video, or text)
Coder	Coder name
PacketInterval	Coder packet interval
LocalRtpIp	Local RTP IP address
LocalRtpPort	Local RTP port
RemoteRtpIp	Remote RTP IP address
RemoteRtpPort	Remote RTP port
InPackets	Number of received packets
OutPackets	Number of sent packets
LocalPackLoss	Local packet loss
RemotePackLoss	Remote packet loss
RTPdelay	RTP delay
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

45.2.1.3 CDR Fields for Locally Stored SBC

The CDR fields for SBC calls that are stored locally (history) on the device are listed in the table below. For storing CDRs locally, see "Storing CDRs on the Device" on page 663.

Table 45-4: Default CDR Fields for Locally Stored (History) CDRs

CDR Field	Title
Report Type	SBCReportType
Endpoint Type	EPTyp
Call Id	SIPCallId
Session ID	SessionId
Call Orig	Orig
Source IP	SourceIp
Source Port	SourcePort
Destination IP	DestIp
Destination Port	DestPort
Transport Type	TransportType
Source URI	SrcURI
Source URI Before Manipulation	SrcURIBeforeMap
Destination URI	DstURI
Destination URI Before Manipulation	DstURIBeforeMap
Call Duration	Durat
Termination Side	TrmSd
Termination Reason	TrmReason
Termination Reason Category	TrmReasonCategory
Setup Time	SetupTime Note: To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter.
Connect Time	ConnectTime Note: To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter.
Release Time	ReleaseTime Note: To configure the time zone string (e.g., "UTC" - default, "GMT+1", and "EST"), use the TimeZoneFormat parameter.
Redirect Reason	RedirectReason
Redirect URI	RedirectURINum

CDR Field	Title
Redirect URI Before Manipulation	RedirectURINumBeforeMap
Signaling IP DiffServ	TxSigIPDiffServ
IP Group Description	IPGroup (description)
SRD Name	SrdId (name)
SIP Interface ID	SIPInterfaceId
Proxy Set ID	ProxySetId
IP Profile ID	IpProfileId (name)
Media Realm Name	MediaRealmId (name)
Direct Media	DirectMedia
SIP Termination Reason	SIPTrmReason
SIP Termination Description	SIPTermDesc
Caller Display ID	Caller
Callee Display ID	Callee

45.2.2 Customizing CDRs for SBC Calls

The SBC CDR Format table lets you customize SBC-related CDRs that are generated by the device for the following:

- CDRs (media and SIP signaling) sent in Syslog messages. For CDRs sent in Syslog messages, you can customize the name of the CDR field. The table lets you configure up to 128 Syslog CDR customization rules.
- CDRs related to RADIUS accounting and sent in RADIUS accounting request messages. For RADIUS accounting CDRs, you can customize the RADIUS Attribute's prefix name and RADIUS Attribute's ID, for standard RADIUS Attributes and vendor-specific RADIUS Attributes (VSA). For example, instead of the default VSA name, "h323-connect-time" with RADIUS Attribute ID 28, you can change the name to "Call-Connect-Time" with ID 29. The table lets you configure up to 40 RADIUS-accounting CDR customization rules. For more information on RADIUS accounting, see [Configuring RADIUS Accounting](#) on page 667.
- CDRs stored locally on the device. For local storage of CDRs, you can customize the name of the CDR field. The table lets you configure up to 64 locally-stored CDR customization rules. For more information on storing CDRs on the device, see [Storing CDRs on the Device](#) on page 663.

If you do not configure a CDR customization rule for a specific CDR, the device generates the CDR in a predefined default CDR format (see [CDR Field Description](#) on page 653).



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.

The following procedure describes how to customize SBC-related CDRs through the Web interface. You can also configure it through ini file (SBCCDRFormat) or CLI (configure voip > services cdr > cdr-format sbc-cdr-format).

➤ **To customize SBC-related CDRs:**

1. Open the SBC CDR Format table (**Configuration** tab > **System** menu > **Call Detail Record** > **SBC CDR Format**).
2. Click **Add**; the following dialog box appears:

Figure 45-2: SBC CDR Format Table - Add Row Dialog Box

3. Configure the CDR according to the parameters described in the table below.
4. Click **Add**.

An example of CDR customization rules configured in the table is shown below:

Figure 45-3: Example of CDR Customization Rules for SBC Calls

Index	CDR Type	Column Type	Title	Radius Attribute Type	Radius Attribute ID
0	Syslog SBC	Source IP	"Source IP Address"	Standard	0
1	Radius SBC	Release Time	disconnect-time=	Vendor Specific	29
2	Radius SBC	Local Output Packets		Standard	48

SBC CDR Format Table Parameter Descriptions

Parameter	Description
Index [SBCCDRFormat_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
CDR Type cdr - type [SBCCDRFormat_CDRT ype]	Defines the application type for which you want to customize CDRs. ▪ [1] Syslog SBC = (Default) Customizes CDR fields for SIP signaling-related CDRs sent in Syslog messages. ▪ [3] Syslog Media = Customizes CDR fields for media-related CDRs sent in Syslog messages. ▪ [5] History SBC = Customizes CDR fields that are stored locally on the device. ▪ [7] RADIUS SBC = Customizes CDR fields (i.e., RADIUS Attributes)

Parameter	Description
	for CDRs sent in RADIUS accounting request messages.
Column Type col-type [SBCCDRFormat_ColumnType]	Defines the CDR field (column) that you want to customize. The applicable CDR field depends on the settings of the 'CDR Type' parameter: ▪ For all types: [0] CDR Type (default); [1] Call ID; [2] Session ID; [3] Report Type; [4] Media Type; [5] Accounting Status Type; [6] H323 ID; [7] Radius Call ID; [8] Blank. ▪ Syslog SBC, History SBC, and RADIUS SBC: [10] Endpoint Type; [11] Call Orig; [12] Source IP; [13] Destination IP; [14] Remote IP; [15] Source Port; [16] Dest Port; [17] Remote Port; [18] Call Duration; [19] Termination Side; [20] Termination Reason; [21] Setup Time; [22] Connect Time; [23] Release Time; [24] Redirect Reason; [25] Was Call Started; [26] IP Group ID; [27] IP Group Name; [28] SRD ID; [29] SRD Name; [30] SIP Interface ID; [31] Transport Type; [32] Signaling IP DiffServ; [33] Termination Reason Category; [34] Proxy Set ID; [35] IP Profile ID; [36] IP Profile Name; [37] Media Realm ID; [38] Media Realm Name; [39] SIP Termination Reason; [40] SIP Termination Description; [41] Caller Display ID; [42] Callee Display ID; [43] SIPInterface Name; [44] Call Orig Radius; [45] Termination Side Radius; [46] Termination Side Yes No; [47] Termination Reason Value; [48] ProxySet Name. ▪ Syslog Media and RADIUS SBC: [150] Channel ID; [151] Coder Type; [152] Packet Interval; [153] Payload Type; [154] Local Input Packets; [155] Local Output Packets; [156] Local Input Octets; [157] Local Output Octets; [158] Local Packet Loss; [159] Local Round Trip Delay; [160] Local Jitter; [161] Local SSRC Sender; [162] Remote Input Packets; [163] Remote Output Packets; [164] Remote Input Octets; [165] Remote Output Octets; [166] Remote Packet Loss; [167] Remote Round Trip Delay; [168] Remote Jitter; [169] Remote SSRC Sender; [170] Local RTP IP; [171] Local RTP Port; [172] Remote RTP IP; [173] Remote RTP Port; [174] RTP IP DiffServ; [175] Local R Factor; [176] Remote R Factor; [177] Local MOS CQ; [178] Remote MOS CQ; [179] AMD Decision; [180] AMD Decision Probability; [181] Latched RTP IP; [182] Latched RTP Port; [183] Latched T38 IP; [184] Latched T38 Port. ▪ Syslog SBC, History SBC, and RADIUS SBC: [200] Source URI; [201] Destination URI; [202] Source URI Before Manipulation; [203] Destination URI Before Manipulation; [204] Redirect URI; [205] Redirect URI Before Manipulation; [206] SIP Method; [207] Direct Media; [208] Source Username; [209] Destination Username; [210] Source Username Before Manipulation; [211] Destination Username Before Manipulation; [212] Source Host; [213] Destination Host; [214] Source Host Before Manipulation; [215] Destination Host Before Manipulation.
Title title [SBCCDRFormat_Title]	Defines a new name for the CDR field (for Syslog or local storage) or for the RADIUS Attribute prefix name (for RADIUS accounting) that you selected in the 'Column Type' parameter. You can configure the name to be enclosed by apostrophes (single or double). For example, if you want the CDR field name to appear as 'Phone Duration', you must configure the parameter to 'Phone Duration'. You can also configure the CDR field name with an equals (=) sign, for example "call-connect-time=". Notes: ▪ For RADIUS Attributes that do not require a prefix name, leave the

Parameter	Description
	parameter undefined. The parameter's value is case-sensitive. For example, if you want the CDR field name to be Phone-Duration, you must configure the parameter to "Phone-Duration" (i.e., upper case "P" and "D").
RADIUS Attribute Type radius-type [SBCCDRFormat_RadiusType]	Defines whether the RADIUS Attribute of the CDR field is a standard or vendor-specific attribute. [0] Standard = (Default) For standard RADIUS Attributes. [1] Vendor Specific = For vendor-specific RADIUS Attributes (VSA). Note: The parameter is applicable only for RADIUS accounting (i.e., 'CDR Type' parameter configured to RADIUS SBC).
RADIUS Attribute ID radius-id [SBCCDRFormat_RadiusID]	Defines an ID for the RADIUS Attribute. For VSAs, this represents the VSA ID; for standard Attributes, this represents the Attribute ID (first byte of the Attribute). The valid value is 0 to 255 (one byte). The default is 0. Notes: - The parameter is applicable only for RADIUS accounting (i.e., 'CDR Type' parameter configured to RADIUS SBC). - For VSA's (i.e., 'RADIUS Attribute Type' parameter configured to Vendor Specific), the parameter must be configured to any value other than 0. - For standard RADIUS Attributes (i.e., 'RADIUS Attribute Type' parameter configured to Standard), the value must be a "known" RADIUS ID (per RFC for RADIUS). However, if you configure the ID to 0 (default) for any of the RADIUS Attributes (configured in the 'Column Type' parameter) listed below and then apply your rule (click Add), the device automatically replaces the value with the RADIUS Attribute's ID according to the RFC: ✓ Destination Username: 30 ✓ Source Username: 31 ✓ Accounting Status Type: 40 ✓ Local Input Octets: 42 ✓ Local Output Octets: 43 ✓ Call Duration: 46 ✓ Local Input Packets: 47 ✓ Local Output Packets: 48 If you configure the value to 0 and the RADIUS Attribute is not any of the ones listed above, the configuration is invalid.

45.2.3 Storing CDRs on the Device

The CDRs generated by the device can also be stored locally on the device (RAM or internal SD card).



Note: When the device is reset or powered off, CDRs stored on RAM are deleted.

You can specify the calls (configuration entities) for which you wish to create and store CDRs locally. This is done using Logging Filter rules in the Logging Filters table. For example, you can configure a rule to create CDRs for traffic belonging only to IP Group 2 and store the CDRs locally.

The CDRs are saved in a comma-separated values file (*.csv). The CSV format consists of a table where the first row displays the field names and the second row the corresponding values. For example:

```
Title: Session ID,Duration,Source URI,Destination URI,Termination Reason
```

```
CDR Data: 5678123,45,1000@abc.com,2000@company.com,BYE
```

The value line in the CDR can contain up to 1023 characters. If it contains more than this, the device removes the extra characters.

Once CDR files (*.csv) are saved locally, you can view them or send them to a remote destination (HTTP or FTP) through the CLI:

■ View stored CDR files:

- View all stored CDR files:
`show storage-history`
- View all stored, unused CDR files:
`show storage-history unused`

■ Delete stored CDR files:

- Delete all stored files:
`clear storage-history cdr-storage-history all`
- Delete all stored, unused CDR files:
`clear storage-history cdr-storage-history unused`

■ Save stored CDR files to an external destination:

```
copy storage-history cdr-storage-history <filename> to  
<protocol://destination>
```

Where:

- *filename*: name you want to assign the file. Any file extension name can be used, but as the file content is in CSV format, it is recommended to use the .csv file extension.
- *protocol*: protocol over which the file is sent (tftp, http, or https).

For example:

```
copy storage-history cdr-storage-history my_cdrs.csv to  
tftp://company.com/cdrs
```

The following procedure describes how to configure local CDR storage through the Web interface.

➤ **To configure local CDR storage:**

1. Open the Logging Settings page (**Configuration** tab > **System** menu > **Logging** > **Logging Settings**), and then scroll down to the Local Storage group:

Figure 45-4: CDR Local Storage on Logging Settings Page

CDR Local Storage		
Local Storage Max File Size [KB]	1024	
Local Storage Max Number of Files	5	
Local Storage File Creation Interval [minutes]	60	

2. Configure the following parameters:
 - 'Local Storage Max File Size' (CDRLocalMaxFileSize): Enter the maximum size (in kilobytes) of the CDR file. Once the file size is reached, the device creates a new file for subsequent CDRs, and so on.
 - 'Local Storage Max Number Of Files' (CDRLocalMaxNomOfFiles): Enter the maximum number of CDR files. Once the maximum is reached, a subsequent CDR file replaces the oldest created file.
 - 'Local Storage File Creation Interval' (CDRLocalInterval): Enter the time in minutes for how often the device creates a new CDR file. For example, if configured to 60, it creates a new file every hour even if the maximum file size has not been reached.

For a detailed description of each parameter, see Syslog, CDR and Debug Parameters on page 745.

3. Configure the local storage location--RAM (default) or internal SD card--using the LocalStorageMedia ini file parameter.
4. Open the Logging Filters table (**Configuration** tab > **System** menu > **Logging** > **Logging Filters Table**), and then configure a log filtering rule with the following parameter settings:
 - 'Filter Type' and 'Value': (as desired)
 - 'Log Destination': Local Storage
 - 'Log Type': CDR Only
 - 'Mode': Enable

For more information on the Logging Filters table, see Configuring Log Filter Rules on page 677.



Notes:

- If you have enabled the CDR storage feature and you later decide to change the maximum number of files (CDRLocalMaxNomOfFiles) to a lower value (e.g., from 50 to 10), the device stores the remaining files (e.g., 40) in its memory (i.e., unused files).
- When the device operates in High-Availability mode, stored CDRs are deleted upon device switchover.
- For customizing CDR fields, see "Customizing CDR Fields for SBC Calls" on page 660.

45.2.4 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 690.
2. Open the Call Detail Record Settings page (**Configuration** tab > **System** menu > **Call Detail Record** > **Call Detail Record Settings**).

Figure 45-5: CDR Parameters in Call Detail Record Settings 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 745.
4. 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 690.
- 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.

45.3 Configuring RADIUS Accounting

The device can send accounting data of SIP calls as call detail records (CDR) to a RADIUS Accounting server. CDR-based accounting messages can be sent upon call release, call connection and release, or call setup and release. For a list of the CDR attributes for RADIUS accounting, see the table following the procedure below.

RADIUS CDR attributes have the following format:

- **Standard RADIUS attributes (per RFC):** A typical standard RADIUS attribute is shown below. The RADIUS attribute ID depends on the attribute.

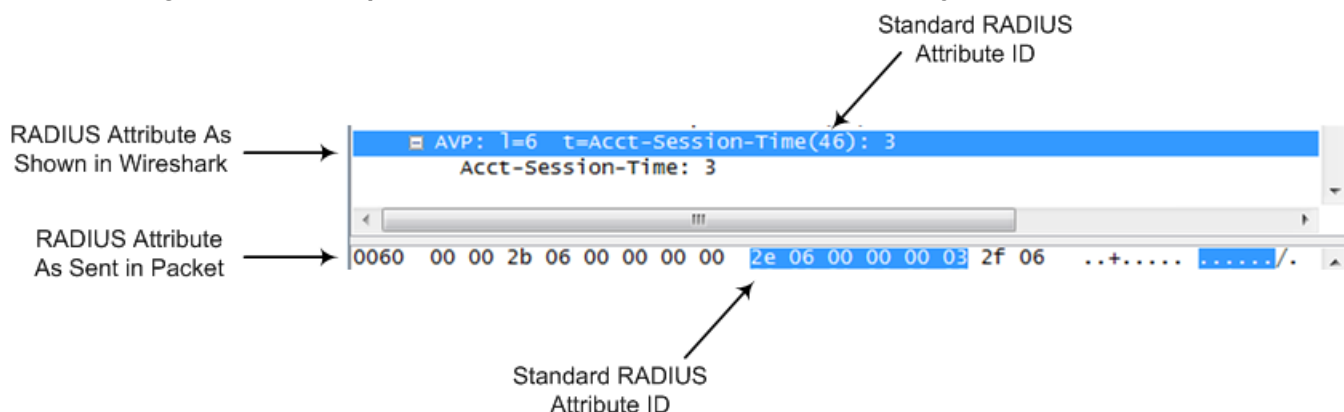
Figure 45-6: Typical Standard RADIUS Attribute

```

2e 06 00 00 00 03 --- Data
|  |
|  | Length (including header)
RADIUS ID
  
```

The following figure shows a standard RADIUS attribute collected by Wireshark. The bottom pane shows the RADIUS attribute information as sent in the packet; the upper pane is Wireshark's interpretation of the RADIUS information in a more readable format. The example shows the attribute in numeric format (32-bit number in 4 bytes).

Figure 45-7: Example of Standard RADIUS Attribute Collected by Wireshark



- **Vendor-specific RADIUS attributes:** RADIUS attributes that are specific to the device (company) are referred to as Vendor-specific attributes (VSA). The CDR of VSAs are sent with a general RADIUS ID of 26 to indicate that they are vendor-specific (non-standard). In addition, the company's registered vendor ID (as registered with the Internet Assigned Numbers Authority or IANA) is also included in the packet. The device's default vendor ID is 5003, which can be changed by using the RadiusVSAVendorID parameter. The VSA ID is also included in the packet.

Figure 45-8: Example of a Vendor-Specific Attribute

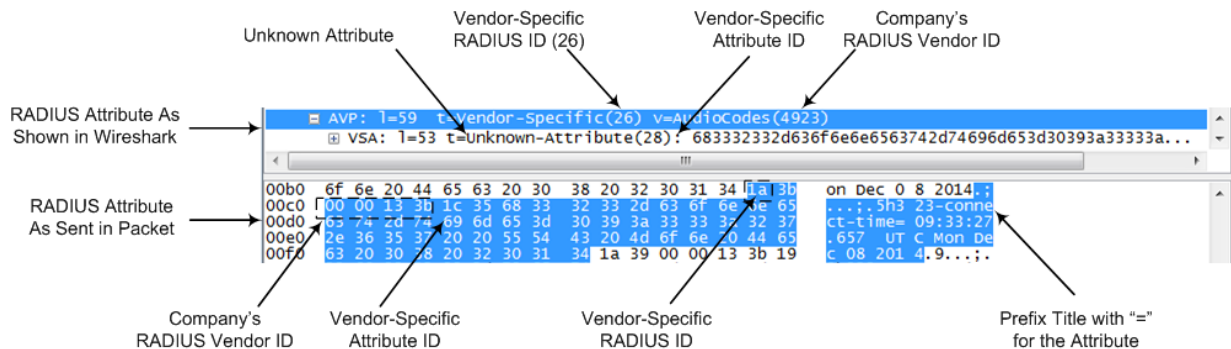
```

1a 13 00 00 13 8b 21 0d 68 33 32 33 2d 67 77 2d 69 64 3d --- Data
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Vendor ID | Vendor part length
|  | (5003)      Vendor-Specific Attribute (VSA) ID
|  | Length (including header)
RADIUS ID indicating vendor-specific (26)
  
```

The following figure shows a vendor-specific RADIUS attribute collected by Wireshark. The bottom pane shows the RADIUS attribute information as sent in the packet; the

upper pane is Wireshark's interpretation of the RADIUS information in a more readable format. The example shows the attribute in string-of-characters format.

Figure 45-9: Example of Vendor-Specific RADIUS Attribute Collected by Wireshark



The table below lists the RADIUS Accounting CDR attributes included in the communication packets transmitted between the device and a RADIUS server.

Table 45-5: 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

Attribute ID	Attribute Name	Vendor-Specific Attribute (VSA) ID	Description	Value Format	Example	AAA
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
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	Start Acc Stop Acc

Attribute ID	Attribute Name	Vendor-Specific Attribute (VSA) ID	Description	Value Format	Example	AAA
					string>	
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: "yes": Call terminated by the Tel side (Gateway) or outgoing leg (SBC) "no": Call terminated by the IP side (Gateway) or incoming leg (SBC)	String	call-terminator=yes	Stop Acc
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 URI	String	8004567145	Start Acc
31	calling-station-id	(Standard)	Source URI	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
42	acct-input-octets	(Standard)	Number of octets received for that call duration (applicable only if media anchoring)	Numeric	-	Stop Acc
43	acct-output-octets	(Standard)	Number of octets sent for that call duration (applicable only if media anchoring)	Numeric	-	Stop Acc

Attribute ID	Attribute Name	Vendor-Specific Attribute (VSA) ID	Description	Value Format	Example	AAA
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 non-standard parameters are preceded with brackets:

```
Accounting-Request (4)
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 X

Diagnostics

46 Syslog and Debug Recording

For debugging and troubleshooting, you can use the device's Syslog and/or Debug Recording capabilities:

- **Syslog:** 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.
- **Debug Recording:** The device can send debug recording packets to a debug 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 server defined by IP address. The debug recording can be done for different types of traffic such as 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: You can include Syslog messages in debug recording (see "Configuring Log Filter Rules" on page 677).

46.1 Configuring Log Filter Rules

The Logging Filters table lets you configure up to 60 rules for filtering debug recording (DR) packets, Syslog messages, and Call Detail Records (CDR). The log filter determines the calls for which you want to generate DR packets, Syslog messages or CDRs. For example, you can add a rule to generate Syslog messages only for calls belonging to IP Groups 2 and 4, or for calls belonging to all IP Groups except for IP Group 3. You can also configure log filters for generating CDRs only and saving them on the device (local storage). DR log filters can include signaling information such as SIP messages, Syslog messages, CDRs, media (RTP, RTCP, and T.38), and pulse-code modulation (PCM).

If you don't configure any rules in the Logging Filters table and you have enabled DR, Syslog, and/or CDR generation (done by simply configuring an IP address for the relevant servers - see Note below), logs are generated for all calls. Thus, the benefit of log filtering is that it allows you to create logs per specific calls, eliminating the need for additional device resources (CPU consumption), otherwise required when logs are generated for all calls.

You can enable and disable configured Logging Filter rules. Enabling a rule activates the rule, whereby the device starts generating the DR packets, Syslog messages, or CDRs. Disabling a rule is useful, for example, if you no longer require the rule, but may need it in the future. Thus, instead of deleting the rule entirely, you can simply disable it.



Notes:

- If you want to configure a Logging Filter rule that logs Syslog messages to a Syslog server (i.e., not to a Debug Recording server), you must enable Syslog functionality, using the 'Enable Syslog' (EnableSyslog) parameter (see "Enabling Syslog" on page 690). Enabling Syslog functionality is not required for rules that include Syslog messages in the DR sent to a Debug Recording server.
- To configure the Syslog server's address, see "Configuring Address of Syslog

Server" on page 690. To configure additional, global Syslog settings, see Configuring Syslog on page 682.

- To configure the Debug Recording server's address, see "Configuring Address of Debug Recording Server" on page 693.
- To configure additional, global CDR settings such as at what stage of the call the CDR is generated (e.g., start and end of call), see Configuring CDR Reporting on page 666.

The following procedure describes how to configure Logging Filter rules through the Web interface. You can also configure it through ini file (LoggingFilters) or CLI (configure system > logging > logging-filters).

➤ **To configure a logging filtering rule:**

1. Open the Logging Filters table (**Configuration** tab > **System** menu > **Logging** > **Logging Filters Table**).
2. Click **Add**; the following dialog box appears:

Figure 46-1: Logging Filters Table - Add Row Dialog Box

3. Configure a logging filter according to the parameters described in the table below.
4. Click **Add**.

Logging Filters Table Parameter Descriptions

Parameter	Description
Index [LoggingFilters_Index]	Defines an index number for the new table row. Note: Each row must be configured with a unique index.
Filter Type filter-type [LoggingFilters_Filter Type]	Defines the filter type criteria. ▪ [1] Any (default) ▪ [8] IP Group = Filters log according to an IP Group. For configuring IP Groups, see "Configuring IP Groups" on page 348. ▪ [9] SRD = Filters log according to an SRD. For configuring SRDs, see Configuring SRDs on page 332. ▪ [10] Classification = Filters log according to a Classification rule. For configuring Classification rules, see Configuring Classification Rules on page 471. Note: Applicable only to the SBC application. ▪ [11] IP-to-IP Routing = Filters log according to an IP-to-IP routing rule. For configuring IP-to-IP routing rules, see Configuring SBC IP-to-IP

Parameter	Description
	Routing Rules on page 480. Note: Applicable only to the SBC application. [12] User = Filters log according to a user. The user is defined by username or username@hostname in the Request-URI of the SIP Request-Line. For example, "2222@10.33.45.201", which represents the following INVITE: <pre>INVITE sip:2222@10.33.45.201;user=phone SIP/2.0</pre> [13] IP Trace = Filters log according to an IP network trace, Wireshark-like expression. For more information on configuring IP traces, see "Filtering IP Network Traces" on page 681. [14] SIP Interface = Filters log according to SIP Interface. For configuring SIP Interfaces, see Configuring SIP Interfaces on page 342.
Value value [LoggingFilters_Value]	Defines the value for the selected filtering type in the 'Filter Type' parameter. The value can include the following: - A single value. - A range, using a hyphen "-" between the two values. For example, to specify IP Groups 1, 2 and 3, configure the parameter to "1-3" (without apostrophes). - Multiple, non-contiguous values, using commas "," between each value. For example, to specify IP Groups 1, 3 and 9, configure the parameter to "1,3,9" (without apostrophes). - The exclamation (!) wildcard character can be used for excluding a specific configuration entity from the filter. For example, to include all IP Groups in the filter except IP Group ID 2, configure the 'Filter Type' parameter to IP Group and the 'Value' parameter to "!2" (without apostrophes). Note that for SBC calls, a Logging Filter rule applies to the entire session, which is both legs (i.e., not per leg). For example, a call between IP Groups 1 and 2 are logged for both legs even if the 'Value' parameter is configured to "!2". Any to indicate all. Notes: - You can use the index number or string name to specify the configuration entity for the following 'Filter Types': IP Group, SRD, Classification, IP-to-IP Routing, or SIP Interface. For example, to specify IP Group at Index 2 with the name "SIP Trunk", configure the parameter to either "2" or "SIP Trunk" (without apostrophes). - For IP trace expressions, see "Filtering IP Network Traces" on page 681.
Log Destination log-dest [LoggingFilters_Log Destination]	Defines where the device sends the log file. [0] Syslog Server = The device generates Syslog messages based on the configured log filter and sends them to a user-defined Syslog server. The Syslog messages can contain one of the following types of information, depending on the settings of the 'Log Type' parameter (described later in the table): ✓ Not configured (default): The Syslog messages contain the regular syslog information. ✓ CDR Only: The Syslog messages contain only CDRs (no system information and alerts). [1] Debug Recording Server = (Default) The device generates DR packets based on the configured log filter and sends them to a user-defined Debug Recording server.

Parameter	Description
	[2] Local Storage = The device generates CDRs based on the configured log filter and stores them locally on the device. For more information on local CDR storage, see Storing CDRs on the Device on page 663. Notes: - If the 'Filter Type' parameter is configured to IP Trace, you must configure the parameter to Debug Recording Server. - If you configure the parameter to Local Storage, you must configure the 'Log Type' parameter to CDR Only. - If you configure the parameter to Syslog Server and the debug level (GwDebugLevel) is configured to No Debug (see "Configuring Syslog Debug Level" on page 689), the Syslog messages include only system Warnings and Errors.
Log Type log-type [LoggingFilters_CaptureType]	Defines the type of messages to include in the log file. [0] = (Default) Not configured. The option is applicable only for sending Syslog messages to a Syslog server (i.e., 'Log Destination' parameter is configured to Syslog Server). [1] Signaling = The option is applicable only for DR (i.e., 'Log Destination' parameter is configured to Debug Recording Server). The DR includes signaling information such as SIP signaling messages, Syslog messages, CDRs, and the device's internal processing messages. [2] Signaling & Media = The option is applicable only for DR (i.e., 'Log Destination' parameter is configured to Debug Recording Server). The DR includes signaling, Syslog messages, and media (RTP/RTCP/T.38). [3] Signaling & Media & PCM = The option is applicable only for DR (i.e., 'Log Destination' parameter is configured to Debug Recording Server). The DR includes signaling, Syslog messages, media, and PCM. [5] CDR Only = Only CDRs are generated. The option is applicable only if the 'Log Destination' parameter is configured to Syslog Server or Local Storage. When configured to Syslog Server, only CDRs are included in the Syslog messages (excluding all system logs and alerts) sent to the Syslog server. Notes: - If you configure the 'Log Destination' parameter to Local Storage, the 'Log Type' parameter must be configured to CDR Only. - The parameter is not applicable when the 'Filter Type' parameter is configured to IP Trace. - To include Syslog messages in DR, it is unnecessary to enable Syslog functionality..
Mode mode [LoggingFilters_Mode]	Enables and disables the rule. [0] Disable [1] Enable (default)

46.1.1 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 46-1: 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



Notes:

- If the 'Value' field is undefined, the device records all IP traffic types.
- You cannot use ip.addr or udp/tcp.port together with ip.src/dst or udp/tcp.srcport/dstport. For example, "ip.addr==1.1.1.1 and ip.src==2.2.2.2" is an invalid configuration value.

46.2 Configuring Syslog

This section describes the Syslog message format, how to configure and enable Syslog, and how to view the generated Syslog messages. For filtering Syslog messages for specific calls, see "Configuring Log Filter Rules" on page 677.

46.2.1 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 690).

Syslog includes two types of log messages:

- SIP call session logs: Logs relating to call sessions (e.g., call established). These logs are identified by a session ID ("SID"), described in detail in the table below. The following is an example of a SIP-session related Syslog message:

```
13:10:57.811 : 10.13.4.12 : NOTICE : [S=235][SID=2ed1c8:96:5]
(lgr_flow)(63) UdpTransportObject#0- Adding socket event for
address 10.33.2.42:5060 [Time: 04-19-2012@18:29:39]
```

- Board logs: Logs relating to the operation of the device (infrastructure) that are non-call session related (e.g., device reset or Web login). These logs are identified by a board ID ("BID"), described in detail in the table below. The following is an example of a board Syslog message:

```
10:21:28.037 : 10.15.7.95 : NOTICE : [S=872] [BID=3aad56:32]
Activity Log: WEB: Successful login at 10.15.7.95:80. User:
Admin. Session: HTTP (10.13.22.54)
```

The format of the Syslog message is described in the following table below:

Table 46-2: 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 and 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, 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:2ed1c8:96:5] (lgr_flow)(620) #0:DIGIT_EV [File: Line:-1] 18:38:14. 83 : 10.33.45.72 : NOTICE: [S=237][SID:2ed1c8:96:5] (lgr_flow)(621) #0:DIGIT_EV [File: Line:-1] 18:38:14.958 : 10.33.45.72 : NOTICE: [S=301][SID:2ed1c8:96:5] (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 Syslog Sequence Number' parameter to Disable (see "Configuring Syslog" on page 690).
Log Number (lgr)(number)	Ignore this number; it has been replaced by the Message Sequence Number (described previously).
Session ID (SID)	Unique SIP call session and device identifier. The device identifier facilitates debugging by clearly identifying the specific device that sent the log message, especially useful in deployments consisting of multiple devices. In addition, the benefit of unique numbering is that it enables you to filter the information (such as SIP, Syslog, and media) according to device or session ID. The syntax of the session and device identifiers are as follows: [SID=<last 6 characters of device's MAC address>:<number of times device has reset>:<unique SID counter indicating the call session; increments consecutively for each new session; resets to 1 after a device reset>] For example: <pre>14:32:52.028: 10.33.8.70: NOTICE: [S=9369] [SID=2ed1c8:96:5] (lgr_psbrdex)(274) recv <-- OFF_HOOK Ch:4</pre> Where: ▪ 2ed1c8 is the device's MAC address. ▪ 96 is the number of times the device has reset. ▪ 5 is a unique SID session number (in other words, this is the fifth call session since the last device reset). ✓ A session includes both the outgoing and incoming legs, where both legs share the same session number. ✓ Forked legs and alternative legs share the same session number. Note: You can configure the device to maintain the same SID value for calls traversing multiple AudioCodes' devices. For more information, see "Maintaining Same Syslog SID/BID over Multiple Devices" on page 685.

Message Item	Description
Board ID (BID)	Unique non-SIP session related (e.g., device reset) and device identifier. The device identifier facilitates debugging by clearly identifying the specific device that sent the log message, especially useful in deployments consisting of multiple devices. In addition, the benefit of unique numbering is that it enables you to filter the information according to device. The syntax of the BID is as follows: [BID=<last 6 characters in MAC>:<number of times device has reset>] For example: 14:32:52.062: 10.33.8.70: WARNING: [S=9399] [BID=2ed1c8:96] invalid Physical index Where: 2ed1c8 is the device's MAC address. 96 is the number of times the device has reset. Note: You can configure the device to maintain the same BID value for calls traversing multiple AudioCodes' devices. For more information, see "Maintaining Same Syslog SID over Multiple Devices" on page 685.
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.

46.2.1.1 Event Representation in Syslog Messages

The Syslog message events that the device sends are denoted by unique abbreviations. The following example shows an abbreviated event in a Syslog message indicating packet loss (PL):

```
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 46-3: Syslog Error Name Descriptions

Error Abbreviation	Error Name Description
AA	Invalid Accumulated Packets Counter
AC	Invalid Channel ID
AL	Invalid Header Length
AO	Invalid Codec Type
AP	Unknown Aggregation Payload Type
AR	Invalid Routing Flag Received
AT	Simple Aggregation Packets Lost
CC	Command Checksum Error

Error Abbreviation	Error Name Description
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

46.2.1.2 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 46-4: 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)
```

46.2.1.3 Syslog Fields for Answering Machine Detection (AMD)

The Syslog message can include information relating to the Answering Machine Detection (AMD) feature. AMD is used to detect whether a human (including a fax machine), an answering machine, silence, or answering machine beeps have answered the call on the remote side.

- AMDSignal – this field can acquire one of the following values:
 - voice (V)
 - answer machine (A)
 - silence (S)
 - unknown (U)
- AMDDecisionProbability – probability (in %) success that correctly detects answering type

Below is an example of such a Syslog message with AMD information:

```
CallMachine:EVENT_DETECTED_EV - AMDSignal = <type - V/A/S/U>,
AMDDecisionProbability = <percentage> %
```

If there is no AMD detection, the AMDSignal field is shown empty (i.e. AMDSignal =).

For more information on the AMD feature, see "Answering Machine Detection (AMD)" on page 206.

46.2.1.4 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 46-5: Syslog Message Severity

ITU Perceived Severity (SNMP Alarm's Severity)	AudioCodes' Syslog Severity
Critical	RecoverableMsg
Major	RecoverableMsg
Minor	RecoverableMsg
Warning	Notice
Indeterminate	Notice
Cleared	Notice

- **Cleared Alarms:** CLEAR-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID >; If exists Additional Info1:/ Additional Info2:/ Additional Info3:

46.2.2 Configuring Web User Activities to Report to Syslog

The device can report operations (activities) performed in the Web interface by management users, by including them in Syslog messages. The Syslog message indicates these logs with the string, "Activity Log". Each logged user activity includes the following information:

- Username (e.g., "Admin") of the user that performed the action
- IP address of the client PC from where the Web user accessed the management interface
- Protocol used for the session (e.g., SSH or HTTP)

The following example shows a Web-user activity log (indicating a login action) with the above-mentioned information:

```
14:07:46.300 : 10.15.7.95 : Local 0 :NOTICE : [S=3149]
[BID=3aad56:32] Activity Log: WEB: Successful login at
10.15.7.95:80. User: Admin. Session: HTTP (10.13.22.54)
```

The device can report the following Web user activities:

- Modifications of individual parameters, for example:


```
14:33:00.162 : 10.15.7.95 : Local 0 :NOTICE : [S=3403]
[BID=3aad56:32] Activity Log: Max Login Attempts was changed
from '3' to '2'. User: Admin. Session: HTTP (10.13.22.54)
```
- Modifications of table fields, and addition and deletion of table rows, for example:


```
14:42:48.334 : 10.15.7.95 : NOTICE : [S=3546] [BID=3aad56:32]
Activity Log: Classification - remove line 2. User: Admin.
Session: HTTP (10.13.22.54)
```
- Entered CLI commands (modifications of security-sensitive commands are logged without the entered value).

- Configuration file load (reported without per-parameter notifications).
- Auxiliary file load and software update.
- Device reset and burn to flash memory.
- Access to unauthorized Web pages according to the Web user's access level.
- Modifications of "sensitive" parameters.
- Login and logout.
- Actions that are not related to parameter changes (for example, file uploads, file delete, lock-unlock maintenance actions, LDAP clear cache, register-unregister, and start-stop trunk. In the Web, these actions are typically done by clicking a button (e.g., the LOCK button).

For more information on each of the above listed options, see "Syslog, CDR and Debug Parameters" on page 745.

You can also configure the device to send an SNMP trap each time a user performs an activity. To enable trap notification, use the parameter, EnableActivityTrap (see "Configuring SNMP Community Strings" on page 94).


Notes:

- You can also view logged user activities in the Web interface (see "Viewing Web User Activity Logs" on page 692).
- Logging of CLI commands can only be configured through CLI or ini file.

The following procedure describes how to configure Web user activity logging through the Web interface. You can also configure it through ini file (ActivityListToLog) or CLI (config-system > logging > activity-log).

➤ **To configure Web user 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.

Figure 46-2: 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	☐
Non-Authorized Access	☐
Sensitive Parameters Value Change	☐
Login and Logout	☐
Action Executed	☐

3. Click **Submit**.

46.2.3 Configuring Syslog Debug Level

You can configure the amount of information (debug level) to include in Syslog messages. In addition, you can enable the device to send multiple Syslog messages bundled into a single packet as well as enable a protection mechanism that automatically lowers the debug level when the device's CPU resources become low, ensuring sufficient CPU resources are available for processing voice traffic.

➤ **To configure the Syslog debug level:**

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

Figure 46-3: Configuring Syslog Debug Level

Syslog CPU Protection	Enabled	▼
Syslog Optimization	Enabled	▼
Debug Level	Detailed	▼

2. From the 'Debug Level' (GwDebugLevel) drop-down list, select the desired debug level of the Syslog messages:
 - **No Debug:** Disables Syslog and no Syslog messages are sent.
 - **Basic:** Sends debug logs of incoming and outgoing SIP messages.
 - **Detailed:** Sends debug logs of incoming and outgoing SIP message as well as many other logged processes.
3. From the 'Syslog Optimization' (SyslogOptimization) drop-down list, select whether you want the device to accumulate and bundle multiple debug messages into a single UDP packet before sending 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.
4. From the 'Syslog CPU Protection' (SyslogCpuProtection) drop-down list, select

whether you want to enable the protection feature for 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 (user-defined 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 to its' previous setting. For example, if you set the 'Debug Level' to **Detailed** and CPU resources decrease to the defined threshold, the device automatically changes the level to **Basic**, and if that is not enough, it changes the level to **No Debug**. Once CPU resources are returned to normal, the device automatically changes the debug level back to its' original setting (i.e., **Detailed**). The threshold is configured by the DebugLevelHighThreshold parameter.

5. Click **Submit**.

46.2.4 Configuring Address of Syslog Server

The following procedure describes how to configure the Syslog server's address to where the device sends the Syslog messages.

- **To configure the address of the Syslog server:**

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

Figure 46-4: Configuring the Syslog Address

Syslog Server IP Address	10.15.50.1
Syslog Server Port	514

2. In the 'Syslog Server IP Address' field, define the IP address of the Syslog server.
3. In the 'Syslog Server Port' field, define the port of the Syslog server.
4. Click **Submit**.

46.2.5 Enabling Syslog

The following procedure describes how to enable Syslog.

- **To enable Syslog:**

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

Figure 46-5: Syslog Settings Page

▼ Syslog Settings	
Enable Syslog	Enable ▼

2. From the 'Enable Syslog' drop-down list, select **Enable**.
3. Click **Submit**.

46.2.6 Viewing Syslog Messages

You can receive and view Syslog messages generated by the device using any of the following Syslog server types:

- **Wireshark** - third-party network protocol analyzer (<http://www.wireshark.org>).



Note: When debug recording is enabled and Syslog messages are also included in the debug recording, to view Syslog messages using Wireshark, you must install AudioCodes' Wireshark plug-in (acsyslog.dll). Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and displayed using the "acsyslog" filter (instead of the regular "syslog" filter). For more information on debug recording, see "Debug Recording" on page 693.

- **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.
- **Device's CLI Console:** The device sends the error messages (e.g. Syslog messages) to the CLI console as well as to the 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
```
- **Device's Web Interface:** The device provides an embedded Syslog server, which is accessed through the Web interface (**Status & Diagnostics** tab > **System Status** menu > **Message Log**). This provides limited Syslog server functionality.

Figure 46-6: 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:0xal
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.



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.

46.2.7 Viewing Web User Activity Logs

If you have enabled the reporting of Web user activities, you can view logged activities in the Web interface's Activity Log table (read-only). For enabling the logging of Web user activities, see "Configuring Web User Activities to Report to Syslog" on page 687.

➤ **To view Web user activity logs:**

- Open the Activity Log table (**Status & Diagnostics** tab > **System Status** menu > **Activity Log**).

Figure 46-7: Activity Log Table

Activity Log Table					
Id	Time	Description	User	Interface	Client
8	03/03/2010, 19:35:55	WEB: Successful login at 10.15.7.96:80	Admin	WEB	10.13.2.17
7	03/03/2010, 19:28:12	User login succeeded	Admin	Telnet	10.13.22.25
6	03/03/2010, 19:20:22	WEB: Successful login at 10.15.7.96:80	Admin	WEB	10.13.22.25
5	03/03/2010, 19:20:13	WEB: User logout	Admin	WEB	10.13.22.25
4	03/03/2010, 19:20:02	Login and Logout was changed from '0' to	Admin	WEB	10.13.22.25
3	03/03/2010, 19:20:02	Device Software Update was changed from	Admin	WEB	10.13.22.25
2	03/03/2010, 19:20:02	Flash Memory Burning was changed from	Admin	WEB	10.13.22.25
1	03/03/2010, 19:20:02	Device Reset was changed from '0' to '1'	Admin	WEB	10.13.22.25

Page 1 of 1 20 View 1 - 8 of 8

The table includes the following information:

Table 46-6: Activity Log Table Description

Parameter	Description
Time	Date and time that the user activity was performed.
Description	Description of the user activity.
User	Username of the user that performed the activity.
Interface	Protocol used for the connection to the management interface (e.g., Telnet, SSH, Web, or HTTP).
Client	IP address of the client PC from where the user accessed the Web interface.

46.3 Configuring Debug Recording

This section describes how to configure and activate debug recording, and how to collect debug recording packets. For filtering debug recording packets for specific calls, see "Configuring Log Filter Rules" on page 677.

**Notes:**

- Debug recording is collected only on the device's OAMP interface.
- For a detailed description of the debug recording parameters, see "Syslog, CDR and Debug Parameters" on page 745.

46.3.1 Configuring Address of Debug Recording Server

The procedure below describes how to configure the address of the debug recording (capturing) server to where the device sends the captured traffic. Once you configure an address, the device generates DR packets for all calls. However, you can configure the device to generate DR packets for specific calls, using Logging Filter rules in the Logging Filters table (see "Configuring Log Filter Rules" on page 677).

➤ **To configure the debug recording server's address:**

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

Figure 46-8: Logging Settings Page

▼ Debug Recording	
Debug Recording Destination IP	10.13.4.22
Debug Recording Destination Port	925

2. In the 'Debug Recording Destination IP' field, configure the IP address of the debug capturing server.
3. In the 'Debug Recording Destination Port' field, configure the port of the debug capturing server.
4. Click **Submit**.

46.3.2 Collecting Debug Recording Messages

To collect debug recording packets, use the open source packet capturing 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 www.audiocodes.com/downloads.
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

46.3.3 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.

47 Creating Core Dump and Debug Files upon Device Crash

For debugging purposes, you can create a core dump file and/or debug file. The files may assist you in identifying 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. You can then provide the files 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_700-8-4_mac_00908F099096_1-02-2015_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 and a list of up to 50 of the most recent SNMP alarms that were raised by the device before it crashed.
- 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_700-8-4_mac_00908F099096_1-03-2015_3-29-29*.

The following procedure describes how to configure core dump file creation through the Web interface.

➤ **To enable core dump file generation:**

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 47-1: 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 through the Web interface.

➤ **To save the debug file from the device:**

- In the Debug Utilities page, click the **Save Debug File** button.

48 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).

48.1 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 a Tel-to-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 through the Web interface. You can also configure it through ini file (Test_Call) or CLI (configure system > test-call > test-call-table).

➤ **To configure a test call:**

1. Open the Test Call table (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).

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

Figure 48-1: Test Call Table - Add Row Dialog Box

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

Table 48-1: Test Call Table Parameter Descriptions

Parameter	Description
Common Tab	
Endpoint URI endpoint-uri [Test_Call_Endpoint URI]	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, the parameter is not configured. Note: The parameter is mandatory.
Called URI 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, the parameter is not configured.

Parameter	Description
Route By 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. To specify the IP Group, see the 'IP Group' parameter in the table. ▪ [2] Dest Address = Calls are matched by (or routed to) an SRD and application type. To specify the address, see the 'Destination Address' parameter in the table. Notes: ▪ If configured to GW Tel2IP or Dest Address, you must assign a SIP Interface (see the 'SIP Interface' parameter in the table). ▪ For REGISTER messages: ✓ The GW Tel2IP option cannot be used as the routing method. ✓ If configured to IP Group, only Server-type IP Groups can be used.
IP Group ip-group-id [Test_Call_IPGroupName]	Assigns an IP Group to the rule, which is the IP Group that the test call is sent to or received from. By default, no value is defined (None). Notes: ▪ The parameter is applicable only if the 'Route By' parameter is configured to IP Group [1]. ▪ The IP Group is used for incoming and outgoing calls.
Destination Address dst-address [Test_Call_DestAddress]	Defines the destination host. This can be defined as an IP address[:port] or DNS name[:port]. Note: The parameter is applicable only if the 'Route By' parameter is configured to Dest Address [2].
SIP Interface sip-interface-name [Test_Call_SIPInterfaceName]	Assigns a SIP Interface to the rule, which is the SIP Interface to which the test call is sent and received from. By default, no value is defined (None). Note: The parameter is applicable only if the 'Route By' parameter is configured to GW Tel2IP or Dest Address.
Application Type application-type [Test_Call_ApplicationType]	Defines the application type for the endpoint. This associates the IP Group and SRD to a specific SIP interface. ▪ [2] SBC = SBC application Note: The parameter must always be set to SBC [2].
Destination Transport Type dst-transport [Test_Call_DestTransportType]	Defines the transport type for outgoing calls. ▪ [-1] = Not configured (default) ▪ [0] UDP ▪ [1] TCP ▪ [2] TLS Note: The parameter is applicable only if the 'Route By' parameter is set to [2] (Dest Address).

Parameter	Description
QoE Profile qoe-profile [Test_Call_QOEProfile]	Assigns a QoE Profile to the test call. By default, no value is defined (None). To configure QoE Profiles, see "Configuring Quality of Experience Profiles" on page 313.
Bandwidth Profile bandwidth-profile [Test_Call_BWProfile]	Assigns a Bandwidth Profile to the test call. By default, no value is defined (None). To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 317.
Authentication Tab	
Note: These parameters are applicable only if the Call Party parameter is set to Caller .	
Auto Register 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' parameter settings (see above). [0] Disable (default) [1] Enable
Username user-name [Test_Call_Username]	Defines the authentication username. By default, no username is defined.
Password password [Test_Call_Password]	Defines the authentication password. By default, no password is defined.
Test Setting Tab	
Call Party 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 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 the 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 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: The parameter is applicable only if 'Call Party' is set to Caller .
Calls per Second calls-per-second [Test_Call_CallsPerSecond]	Defines the number of calls per second. Note: The parameter is applicable only if 'Call Party' is set to Caller .

Parameter	Description
Test Mode 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: The parameter is applicable only if 'Call Party' is set to Caller.
Test Duration 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: The parameter is applicable only if 'Call Party' is set to Caller.
Play 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 706. ▪ [2] PRT = Plays a non-DTMF tone from the PRT file (Test Call Tone). For this option, a PRT file must be loaded to the device (see "Prerecorded Tones File" on page 588). Notes: ▪ To configure the DTMF signaling type (e.g., out-of-band or in-band) use the 'DTMF Transport Type' parameter (see Configuring DTMF Transport Types). ▪ The parameter is applicable only if 'Call Party' is set to Caller.
Schedule Interval 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: The parameter is applicable only if 'Call Party' is set to Caller.

48.2 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 704).

48.3 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 703), 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 (**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 48-2: 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 703).
- **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.

48.4 Configuring DTMF Tones for Test Calls

By default, the device plays the DTMF signal tone "3212333" to remote tested endpoints for answered calls (incoming and outgoing). For basic test calls (as described in Configuring Basic Test Calls on page 706), the device can play only the configured DTMF tones (or none, if not configured). For test call endpoints that are configured in the Test Call Rules table, you can configure the device to play either DTMF tones or a tone from an installed PRT file (Test Call Tone). For more information, see Configuring Test Call Endpoints on page 699.



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 Configuring DTMF Transport Types.
- 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 48-3: DTMF in Test Call Settings Page

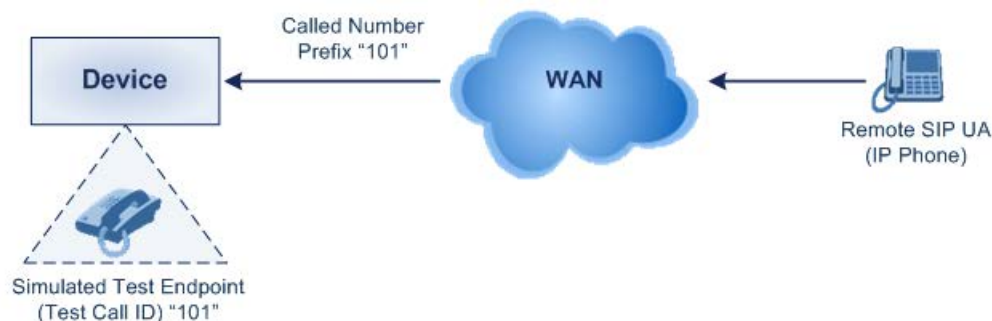
Test Call DTMF String	3212333

2. In the 'Test Call DTMF String' field, enter the DTMF string (up to 15 digits).
3. Click **Submit**.

48.5 Configuring Basic Test Calls

The Basic Test Call feature tests incoming calls from remote SIP (IP) endpoints to a simulated test endpoint on the device. The only required configuration is to assign a prefix number (test call ID) to the simulated endpoint. Incoming calls with this called (destination) prefix number are identified by the device as test calls and sent to the simulated endpoint. The figure below displays a basic test call example:

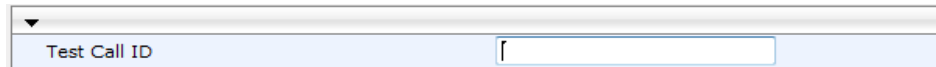
Figure 48-4: Incoming Test Call Example



➤ **To configure basic call testing:**

1. Open the Test Call Settings page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Settings**).
2. In the 'Test Call ID' field, enter a prefix for the simulated endpoint:

Figure 48-5: Configuring Basic Test Calls

A screenshot of a web interface showing a configuration field. The field is labeled 'Test Call ID' and contains a text input box with a cursor inside. The field is part of a larger form with a light blue header bar.

3. Click **Apply**.



Note:

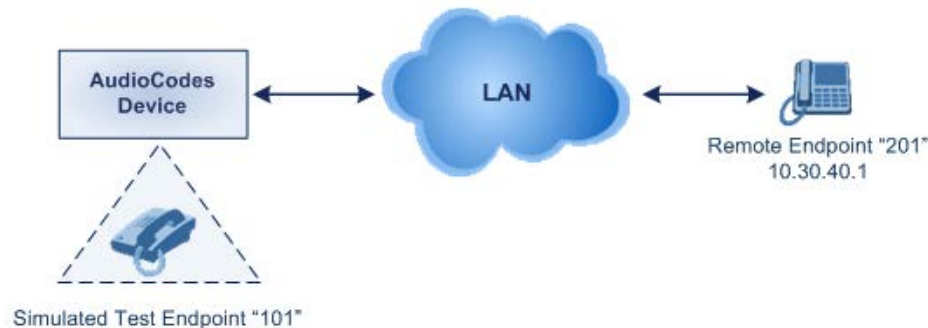
- The device can play DTMF tones to the remote endpoint. For more information, see Configuring DTMF Tones for Test Calls on page 706.
- Test calls are done on all SIP Interfaces.

48.6 Test Call Configuration Examples

Below are a few examples of test call configurations.

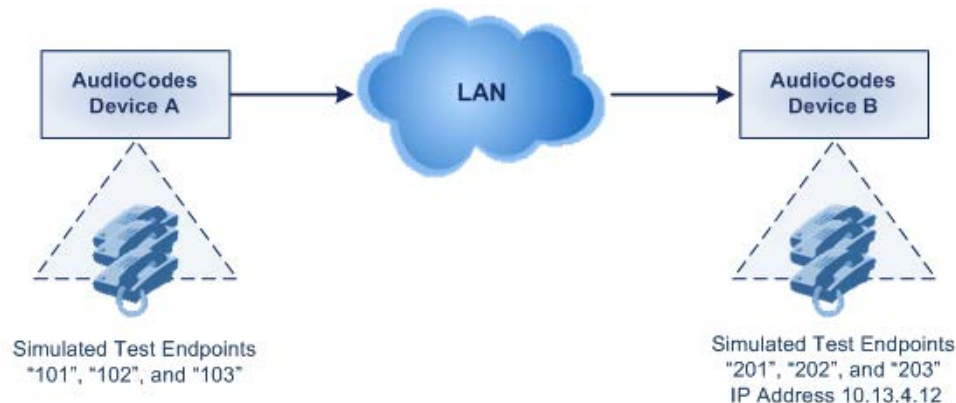
- **Single Test Call Scenario:** This example describes the configuration of a simple test call scenario that includes a single test call between a simulated test endpoint on the device and a remote endpoint.

Figure 48-6: Single Test Call Example



- Test Call table configuration:
 - ◆ Endpoint URI: "101"
 - ◆ Called URI: "201"
 - ◆ Route By: **Dest Address**
 - ◆ Destination Address: "10.30.40.01"
 - ◆ SIP Interface: SIPInterface_0
 - ◆ 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 48-7: Batch Test Call Example

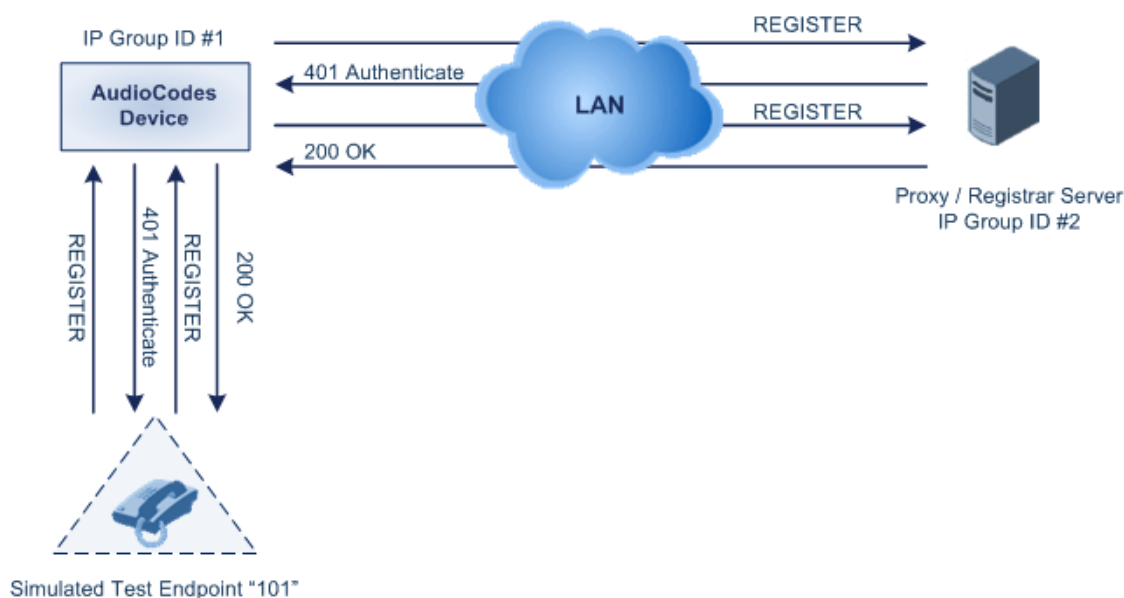


- Test Call table configuration at Device A:
 - ◆ Endpoint URI: "101"
 - ◆ Called URI: "201"
 - ◆ Route By: **Dest Address**

- ◆ Destination Address: "10.13.4.12"
- ◆ SIP Interface: SIPInterface_0
- ◆ 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 48-8: 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)
 - ◆ SIP Interface: SIPInterface_0
 - ◆ Auto Register: **Enable**

- ◆ User Name: "testuser"
- ◆ Password: "12345"
- ◆ Call Party: **Caller**

49 Pinging a Remote Host or IP Address

You can verify the network connectivity with a remote host or IP address by pinging the network entity.

- IPv4: The ping to an IPv4 address can be done from any of the device's VoIP interfaces that is configured with an IPv4 address. The ping is done using the following CLI command:

```
# ping <IPv4 ip address or host name> source [voip|data]  
interface
```

For a complete description of the ping command, refer to the *CLI Reference Guide*.

This page is intentionally left blank.

Part XI

Appendix

50 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.



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.

Table 50-1: 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]).	
[n-m] or (n-m)	Represents a range of numbers. Examples: To depict prefix numbers from 5551200 to 5551300:

Notation	Description
	✓ [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 received 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><tr><th>ASCII Character</th><th>HEX Value</th></tr><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></table>	ASCII Character	HEX Value	*	\2a	(	\28	)	\29	\	\5c	/	\2f
ASCII Character	HEX Value												
*	\2a												
(	\28												
)	\29												
\	\5c												
/	\2f												

This page is intentionally left blank.

51 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.

51.1 Management Parameters

This section describes the device's management-related parameters.

51.1.1 General Parameters

The general management parameters are described in the table below.

Table 51-1: General Management Parameters

Parameter	Description
[WebLoginBlockAutoComplete]	Disables autocompletion when entering the management login username in the 'Username' field of the device's Web interface. Disabling autocompletion may be useful for security purposes by hiding previously entered usernames and thereby, preventing unauthorized access to the device's management interface. ▪ [0] Disable = (Default) Autocompletion is enabled and the 'Username' field automatically offers previously logged in usernames. ▪ [1] Enable = Autocompletion is disabled.
[EnforcePasswordComplexity]	Enables the enforcement of management login-password complexity requirements to ensure strong passwords. ▪ [0] Disable (default) ▪ [1] Enable For more information on password complexity requirements, see the 'Password' parameter in Configuring Management User Accounts on page 71.
[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".

Parameter	Description
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 the parameter, see "Configuring Web and Telnet Access List" on page 79.

51.1.2 Web Parameters

The Web parameters are described in the table below.

Table 51-2: Web Parameters

Parameter	Description
Enable web access from all interfaces web-access-from-all-interfaces [EnableWebAccessFromAllInterfaces]	Enables Web access from any of the device's IP network interfaces. This feature applies to HTTP and HTTPS protocols. [0] = (Default) Disable – Web access is only through the OAMP interface. [1] = Enable - Web access is through any network interface. Note: For the parameter to take effect, a device reset is required.
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: The 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 the parameter's value.
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: The parameter is applicable only when using the Web Users table.
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

Parameter	Description
	15. Note: You can also configure the functionality per user in the Web Users table (see Advanced User Accounts Configuration on page 71), which overrides this global setting.
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.
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.
Display Login Information [DisplayLoginInformation]	Enables display of user's login information on each successful login attempt. ▪ [0] = Disable (default) ▪ [1] = Enable
[EnableMgmtTwoFactorAuthentication]	Enables Web login authentication using a third-party, smart card. ▪ [0] = Disable (default) ▪ [1] = Enable When enabled, the device retrieves the Web user's login username from the smart card, which is automatically displayed (read-only) in the Web Login screen; the user is then required to provide only the login password. Typically, a TLS connection is established between the smart card and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username. Thus, this feature implements a two-factor authentication - what the user has (the physical card) and what the user knows (i.e., the login password).
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 the parameter to take effect, a device reset is required.
[DisableWebConfig]	Determines whether the entire Web interface is read-only. ▪ [0] = (Default) Enables modifications of parameters. ▪ [1] = Web interface is read-only. When in read-only mode, parameters can't be modified. In addition, the following pages can't be accessed: 'Web User Accounts', 'Certificates', 'Regional Settings', 'Maintenance Actions' and all file-loading pages ('Load Auxiliary Files', 'Software Upgrade Wizard', and 'Configuration File'). Note: For the 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

Parameter	Description
	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: - Set acSysGenericINILine to WEBPasswordControlViaSNMP = 1, and reset the device with a flash burn (set acSysActionSetResetControl to 1 and acSysActionSetReset to 1). - Change the username and password in the acSysWEBAccessEntry table. Use the following format: Username acSysWEBAccessUserName: old/pass/new Password acSysWEBAccessUserCode: username/old/new
HA Device Name [HUnitIdName]	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.
Customizing Web GUI	
[WelcomeMessage] configure system > welcome-msg	Defines a welcome message displayed on the Web interface's Web Login page. The format of the ini file table parameter is: <pre>[WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text [WelcomeMessage]</pre> For Example: <pre>FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "*****" ; WelcomeMessage 2 = "***** This is a Welcome message ***" ; WelcomeMessage 3 = "*****" ;</pre> For more information, see Creating a Login Welcome Message on page 64. Note: - Each index row represents a line of text. 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 the parameter is not configured, no Welcome message is displayed.
[UseProductName]	Enables the option to customize the name of the device (product) that appears in the management interfaces. [0] = Disabled (default). [1] = Enables the display of a user-defined name, which is configured by the UserProductName parameter.

Parameter	Description
	For more information, see Customizing the Product Name on page 62.
[UserProductName]	Defines a name for the device instead of the default name. The value can be a string of up to 29 characters. For more information, see Customizing the Product Name on page 62. Note: To enable customization of the device name, see the UseProductName parameter.
[UseWebLogo]	Defines whether the Web interface displays a logo image or text. ▪ [0] = (Default) The Web interface displays a logo image, configured by the LogoFileName parameter. ▪ [1] = The Web interface displays text, configured by the WebLogoText parameter. For more information, see Replacing the Corporate Logo on page 60.
[WebLogoText]	Defines the text that is displayed instead of the logo in the Web interface. The valid value is a string of up to 15 characters. For more information, see Replacing the Corporate Logo with Text on page 61. Note: The parameter is applicable only when the UseWebLogo parameter is configured to 1.
[LogoWidth]	Defines the width (in pixels) of the logo image that you want displayed in the Web interface instead of the default logo. The valid value is 0 to 199. The default is 145. For more information, see Replacing the Corporate Logo with an Image on page 61. Notes: ▪ The optimal setting depends on your screen resolution. ▪ If the width of the loaded image is greater than the maximum value, the device automatically resizes the image to the default width size. ▪ The height is limited to 24 pixels. ▪ The parameter is applicable only when the UseWebLogo parameter is configured to 0. ▪ To define the image file, see the LogoFileName parameter.
[LogoFileName]	Defines the name of the image file that you want loaded to the device. This image is displayed as the logo in the Web interface (instead of AudioCodes logo). The file name can be up to 47 characters. For more information, see Replacing the Corporate Logo with an Image on page 61. Notes: ▪ The image file type can be one of the following: GIF, PNG, JPG, or JPEG. ▪ The size of the image file can be up to 64 Kbytes. ▪ The parameter is applicable only when the UseWebLogo parameter is configured to 0.

51.1.3 Telnet Parameters

The Telnet parameters are described in the table below.

Table 51-3: Telnet Parameters

Parameter	Description
Embedded Telnet Server 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 67).
Telnet Server TCP Port 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.
Telnet Server Idle Timeout 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 the parameter to take effect, a device reset is required.
Maximum Telnet Sessions telnet-max-sessions [TelnetMaxSessions]	Defines the maximum number of permitted, concurrent Telnet/SSH sessions. The valid range is 1 to 5 sessions. The default is 2. Note: Before changing the value, make sure that not more than this number of sessions are currently active; otherwise, the new setting will not take effect.
[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.

51.1.4 ini File Parameters

The parameters relating to ini-file management are described in the table below.

Table 51-4: 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 (*).

51.1.5 SNMP Parameters

The SNMP parameters are described in the table below.

Table 51-5: SNMP Parameters

Parameter	Description
Disable SNMP disable [DisableSNMP]	Enables and disables SNMP. ▪ [0] No = (Default) SNMP is enabled. ▪ [1] Yes = SNMP is disabled. Note: For the parameter to take effect, a device reset is required.
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 the 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.
auto-send-keep-alive [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 acKeepAlive. 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 the parameter to take effect, a device reset is required.
[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

Parameter	Description
[KeepAliveTrapPort]	Defines the port of the SNMP network management station to which the device sends keep-alive traps. The valid range is 0 - 65534. The default is port 1161. To enable NAT keep-alive traps, use the SendKeepAliveTrap parameter.
sys-oid [SNMPSysOid]	Defines the base product system OID. The default is eSNMP_AC_PRODUCT_BASE_OID_D. Note: For the 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 the parameter. Note: For the 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. The 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 the parameter to take effect, a device reset is required.
[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 600. The default is 200. For more information on the Active Alarms table, see Viewing Active Alarms on page 637. 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.
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 the parameter to take effect, a device reset is required. - Before setting the 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

Parameter	Description
	values input or it is set with the default value, the engine ID is generated according to RFC 3411.
SNMP Trap Destination Parameters (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
IP Address 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.
Trap Port 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.
Trap Enable 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.
Trap User 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.
Trap Manager Host Name 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: 'mngn.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".

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".
SNMP Trusted Managers Table	
SNMP Trusted Managers configure system > snmp > trusted-managers [SNMPTrustedMgr_x]	Defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. Notes: - By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests. - If no values are assigned to these parameters any manager can access the device. - Trusted managers can work with all community strings.
SNMP V3 Users Table	
SNMP V3 Users configure system > snmp v3-users [SNMPUsers]	The <i>parameter</i> table defines SNMP v3 users. The format of the ini file table parameter is: [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 the table, see "Configuring SNMP V3 Users" on page 98.

51.1.6 Serial Parameters

The serial interface parameters are described in the table below.

Table 51-6: 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 Installation Manual. Note: For the parameter to take effect, a device reset is required.
[SerialBaudRate]	Defines the serial communication baud rate. The valid values include the following: 1200, 2400, 9600, 14400, 19200, 38400, 57600, or 115200 (default). Note: For the parameter to take effect, a device reset is required.
[SerialData]	Defines the serial communication data bit. ▪ [7] = 7-bit ▪ [8] = (Default) 8-bit Note: For the parameter to take effect, a device reset is required.
[SerialParity]	Defines the serial communication polarity. ▪ [0] = (Default) None ▪ [1] = Odd ▪ [2] = Even Note: For the parameter to take effect, a device reset is required.
[SerialStop]	Defines the serial communication stop bit. ▪ [1] = (Default) 1-bit (default) ▪ [2] = 2-bit Note: For the parameter to take effect, a device reset is required.
[SerialFlowControl]	Defines the serial communication flow control. ▪ [0] = (Default) None ▪ [1] = Hardware Note: For the parameter to take effect, a device reset is required.

51.1.7 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 Auxiliary files, see "Loading Auxiliary Files" on page 583.

Table 51-7: 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: The 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	
Call Progress Tones File [CallProgressTonesFilename]	Defines the name of the file containing the Call Progress Tones definitions. For the ini file, the name must be enclosed by single apostrophes, for example, 'cpt_us.dat'. For more information on how to create and load this file, refer to <i>DConvert Utility User's Guide</i>. Note: For the parameter to take effect, a device reset is required.
Prerecorded Tones File [PrerecordedTonesFilename]	Defines the name of the file containing the Prerecorded Tones. Note: For the parameter to take effect, a device reset is required.
Dial Plan File [DialPlanFileName]	Defines the name of the Dial Plan file. This file should be created using AudioCodes DConvert utility (refer to <i>DConvert Utility User's Guide</i>). For the ini file, the name must be enclosed by single apostrophes, for example, 'dial_plan.dat'.
[UserInfoFileName]	Defines the name of the file containing the User Information data. For the ini file, the name must be enclosed by single apostrophes, for example, 'userinfo_us.dat'.

51.1.8 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

Table 51-8: Automatic Update of Software and Configuration Files Parameters

Parameter	Description
General Automatic Update Parameters	
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 the parameter to take effect, a device reset is required.
configure system > automatic-update > update-frequency [AutoUpdateFrequency]	Defines the interval (in minutes) that the device waits between consecutive automatic updates. The default is 0 (i.e., the update at fixed intervals mechanism is disabled). Note: For the parameter to take effect, a device reset is required.
configure system > automatic-update > predefined-time [AutoUpdatePredefinedTime]	Defines schedules (time of day) for performing automatic updates. The format syntax of the parameter is 'hh:mm', where <i>hh</i> denotes the hour and <i>mm</i> the minutes. The value must be enclosed in single apostrophes. For example, '20:18'. Notes: - For the 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.
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., "7.00.200.001" <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;7.00.200.001(00908F1DD0D3)</pre>

Parameter	Description
	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. - The tags must be defined adjacent to one another (i.e., no spaces).
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.
system > tls > aupd-verify-cert [AUPDVerifyCertificates]	Determines whether the Automatic Update mechanism verifies server certificates when using HTTPS. [0] = Disable (default) [1] = Enable
[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.
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).
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

Parameter	Description
	increased. The valid value is 512 to 8192. The default is 512. Notes: - A higher value does not necessarily mean better performance. - The block size should be small enough to avoid IP fragmentation in the client network (i.e., below MTU). - This feature is applicable only to TFTP servers that support this option.
[ResetNow]	Invokes an immediate device reset. This option can be used to activate offline (i.e., not on-the-fly) parameters that are loaded using the parameter IniFileUrl. [0] = (Default) The immediate restart mechanism is disabled. [1] = The device immediately resets after an <i>ini</i> file with the parameter set to 1 is loaded. Note: If you use the parameter in an ini file for periodic automatic provisioning with non-HTTP (e.g., TFTP) and without CRC, the device resets upon every file download.
Software/Configuration File URL Path for Automatic Update Parameters	
automatic-update > 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, http://192.168.0.1/filename. Notes: - For the parameter to take effect, a device reset is required. - When the 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.
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: http://192.168.0.1/filename <a href="http://192.8.77.13/config_<MAC>.ini">http://192.8.77.13/config_<MAC>.ini <a href="https://<username>:<password>@<IP address>/<file name>">https://<username>:<password>@<IP address>/<file name> Notes: - For the 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 Placeholder in Configuration File Name" on page 618. This option allows the loading of specific configurations for specific devices. - The maximum length of the URL address is 99 characters.
prerecorded-tones	Defines the name of the Prerecorded Tones (PRT) file and the path to the

Parameter	Description
[PrtFileURL]	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.
call-progress-tones [CptFileURL]	Defines the name of the CPT file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. Note: The maximum length of the URL address is 99 characters.
tls-root-cert [TLSTRootFileUrl]	Defines the name of the TLS trusted root certificate file and the URL from where it can be downloaded. Note: For the parameter to take effect, a device reset is required.
tls-cert [TLSCertFileUrl]	Defines the name of the TLS certificate file and the URL from where it can be downloaded. Note: For the parameter to take effect, a device reset is required.
tls-private-key [TLSPkeyFileUrl]	Defines the URL for downloading a TLS private key file using the Automatic Update facility.
user-info [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.
configure system > automatic-update > feature-key [FeatureKeyURL]	Defines the name of the License Key file and the URL address of the server on which the file is located.
configure system > automatic-update > template-url [TemplateUrl]	Defines the URL address in the File Template for automatic updates, of the provisioning server on which the files to download are located. For more information, see File Template for Automatic Provisioning on page 619.
configure system > automatic-update > template-files-list [AupdFilesList]	Defines the list of file types in the File Template for automatic updates, to download from the provisioning server. For more information, see File Template for Automatic Provisioning on page 619.
web-favicon [WebFaviconFileUrl]	Defines the name of the favicon image file and the URL address of the server on which the file is located. This is used for the Automatic Update feature. For more information, see Customizing the Favicon on page 62.

51.2 Networking Parameters

This subsection describes the device's networking parameters.

51.2.1 Ethernet Parameters

The Ethernet parameters are described in the table below.

Table 51-9: Ethernet Parameters

Parameter	Description
Physical Ports Settings Table	
Physical Ports Settings configure voip/physical-port [PhysicalPortsTable]	The table configures the physical Ethernet ports. The format of the ini file table parameter is as follows: <pre>[PhysicalPortsTable] FORMAT PhysicalPortsTable_Index = PhysicalPortsTable_Port, PhysicalPortsTable_Mode, PhysicalPortsTable_SpeedDuplex, PhysicalPortsTable_PortDescription, PhysicalPortsTable_GroupMember, PhysicalPortsTable_GroupStatus; [\PhysicalPortsTable]</pre> For a detailed description of the table, see Configuring Physical Ethernet Ports on page 131.
Ethernet Group Settings Table	
Ethernet Group Settings configure voip/ether-group [EtherGroupTable]	Defines the transmit (Tx) and receive (Rx) settings for the Ethernet port groups. The format of the ini file table parameter is: <pre>[EtherGroupTable] FORMAT EtherGroupTable_Index = EtherGroupTable_Group, EtherGroupTable_Mode, EtherGroupTable_Member1, EtherGroupTable_Member2; [\EtherGroupTable]</pre> For a detailed description of the table, see Configuring Ethernet Port Groups on page 133. Note: For the parameter to take effect, a device reset is required.
Ethernet Device Table	
Ethernet Device Table [DeviceTable]	Defines Ethernet Devices (VLANs). The format of the ini file table parameter is as follows: <pre>[DeviceTable] FORMAT DeviceTable_Index = DeviceTable_VlanID, DeviceTable_UnderlyingInterface, DeviceTable_DeviceName, DeviceTable_Tagging; [\DeviceTable]</pre> For a detailed description of the table, see Configuring Underlying Ethernet Devices on page 135.

51.2.2 Multiple VoIP Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

Table 51-10: IP Network Interfaces and VLAN Parameters

Parameter	Description
Interface Table	
Interface Table configure voip > interface network- if display [InterfaceTable]	The table configures the Interface table. The format of the ini file table parameter is as follows: [InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName, InterfaceTable_PrimaryDNSServerIPAddress, InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingDevice; [InterfaceTable] For a detailed description of the table, see "Configuring IP Network Interfaces" on page 138.
[EnableNTPasOAM]	Defines the application type for Network Time Protocol (NTP) services. [1] = OAMP (default) [0] = Control Note: For the parameter to take effect, a device reset is required.

51.2.3 Routing Parameters

The IP network routing parameters are described in the table below.

Table 51-11: IP Network Routing Parameters

Parameter	Description
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.
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.
Static Route Table	
Static Route Table configure voip > static [StaticRouteTable]	Defines up to 30 static IP routes for the device. The format of the ini file table parameter is as follows: [StaticRouteTable] FORMAT StaticRouteTable_Index = StaticRouteTable_DeviceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [\StaticRouteTable]

Parameter	Description
	For a description of the parameter, see "Configuring Static IP Routes" on page 146.

51.2.4 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

Table 51-12: QoS Parameters

Parameter	Description
Layer-2 Class Of Service (CoS) Parameters (VLAN Tag Priority Field)	
DiffServ Table configure voip > vlan-mapping [DiffServToVlanPriority]	The table 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 the table, see Configuring Quality of Service on page 149. Note: For the parameter to take effect, a device reset is required.
Layer-3 Class of Service (TOS/DiffServ) Parameters	
Media Premium QoS 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
Control Premium QoS 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
Gold QoS 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.

Parameter	Description
]	
Bronze QoS 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.

51.2.5 NAT Parameters

The Network Address Translation (NAT) parameters are described in the table below.

Table 51-13: NAT Parameters

Parameter	Description
NAT Mode disable-NAT-traversal [NATMode]	Enables the NAT feature for media when the device communicates with UAs located behind NAT. [0] Enable NAT Option = NAT traversal is performed only if the UA is located behind NAT: ✓ UA behind NAT: The device sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA. ✓ UA not behind NAT: The device sends the packets to the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message. Note: 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] Disable NAT = (Default) The device considers the UA as not located behind NAT and sends media packets to the UA using the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message. [2] Force NAT = The device always considers the UA as behind NAT and sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA. The device only sends packets to the UA after it receives the first packet from the UA (to obtain the IP address). [3] NAT By Signaling = The device identifies whether or not the UA is located behind NAT based on the SIP signaling. The device assumes that if signaling is behind NAT that the media is also behind NAT, and vice versa. If located behind NAT, the device sends media as described in option [2] Force NAT; if not behind NAT, the device sends media as described in option [1] Disable NAT. This option is applicable only to SBC calls. If the parameter is configured to this option, Gateway calls use option [0] Enable NAT Option, by default. For more information on handling calls from UAs behind NAT, see "First Incoming Packet Mechanism" on page 160.
NAT IP Address 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 the parameter to take effect, a device reset is required.
[NATBindingDefaultTimeo	The device sends SNMP keep-alive traps periodically - every 9/10 of the

Parameter	Description
ut]	time configured by the 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 the parameter to take effect, a device reset is required.
SIP NAT Detection 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.

51.2.6 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table 51-14: DNS Parameters

Parameter	Description
Internal DNS Table	
Internal DNS Table configure voip > voip-network dns Dns2Ip [DNS2IP]	The table defines the internal DNS table for resolving host names into IP addresses. The format of the ini file table parameter is: <pre>[Dns2Ip] FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress, Dns2Ip_FourthIpAddress; [\Dns2Ip]</pre> For example: Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, ; For a detailed description of the table, see "Configuring the Internal DNS Table" on page 152.
Internal SRV Table	
Internal SRV Table configure voip > voip-network dns Srv2Ip [SRV2IP]	The table 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 the ini file table parameter is: <pre>[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,</pre>

Parameter	Description
	SRV2IP_Weight3, SRV2IP_Port3; [\SRV2IP] For example: SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,\$\$,0,0,0; For a detailed description of the table, see "Configuring the Internal SRV Table" on page 153.

51.2.7 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

Table 51-15: DHCP Parameters

Parameter	Description
Enable DHCP [DHCPEnable]	Enables DHCP client functionality. ▪ [0] Disable (default) ▪ [1] Enable Notes: ▪ For the parameter to take effect, a device reset is required. ▪ For a detailed description of DHCP, see "DHCP-based Provisioning" on page 609. ▪ The 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 the parameter to take effect, a device reset is required.
DHCP Servers Table	

Parameter	Description
DHCP Servers Table configure voip > dhcp server <index> [DhcpServer]	Defines the device's embedded DHCP server. The format of the ini file table parameter is as follows: [DhcpServer] FORMAT DhcpServer_Index = DhcpServer_InterfaceName, DhcpServer_StartIPAddress, DhcpServer_EndIPAddress, DhcpServer_SubnetMask, DhcpServer_LeaseTime, DhcpServer_DNSServer1, DhcpServer_DNSServer2, DhcpServer_NetbiosNameServer, DhcpServer_NetbiosNodeType, DhcpServer_NTPTServer1, DhcpServer_NTPTServer2, DhcpServer_TimeOffset, DhcpServer_TftpServer, DhcpServer_BootFileName, DhcpServer_ExpandBootfileName, DhcpServer_OverrideRouter, DhcpServer_SipServer, DhcpServer_SipServerType; [\DhcpServer] For a detailed description of the table, see Configuring the Device's DHCP Server.
DHCP Vendor Class Table	
DHCP Vendor Class table 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: [DhcpVendorClass] FORMAT DhcpVendorClass_Index = DhcpVendorClass_DhcpServerIndex, DhcpVendorClass_VendorClassId; [\DhcpVendorClass] For a detailed description of the table, see Configuring the Vendor Class Identifier on page 220.
DHCP Option Table	
DHCP Option table 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: [DhcpOption] FORMAT DhcpOption_Index = DhcpOption_DhcpServerIndex, DhcpOption_Option, DhcpOption_Type, DhcpOption_Value, DhcpOption_ExpandValue; [\DhcpOption] For a detailed description of the table, see Configuring Additional DHCP Options on page 221.
DHCP Static IP Table	

Parameter	Description
DHCP Static IP table 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: [DhcpStaticIP] FORMAT DhcpStaticIP_Index = DhcpStaticIP_DhcpServerIndex, DhcpStaticIP_IPAddress, DhcpStaticIP_MACAddress; [\DhcpStaticIP] For a detailed description of the table, see Configuring Static IP Addresses for DHCP Clients on page 223.

51.2.8 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

Table 51-16: 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 125.	
NTP Server Address 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).
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.
NTP Update Interval 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 the parameter to beyond one month (i.e., 2592000 seconds).
NTP Authentication Key Identifier 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).
NTP Authentication Secret Key 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.

Parameter	Description
Regional Clock and Daylight Saving Time Parameters	
UTC Offset utc-offset [NTPServerUTCOffset]	Defines the Universal Time Coordinate (UTC) offset (in seconds) from the local time. The valid range is -43200 to 43200. The default is 0. Note: The offset setting is applied only on the hour. For example, if you configure the parameter at 15:42, the device applies the setting only at 16:00.
Daylight Saving Time summer-time [DayLightSavingTimeEnable]	Enables daylight saving time (DST). ▪ [0] Disable (default) ▪ [1] Enable
Start Time / Day of Month Start start [DayLightSavingTimeStart]	Defines the date and time when DST 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.
End Time / Day of Month End end [DayLightSavingTimeEnd]	Defines the date and time when DST ends. For a description of the format of this value, see the DayLightSavingTimeStart parameter.
Offset offset [DayLightSavingTimeOffset]	Defines the DST 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 the parameter at 15:42, the device applies the setting only at 16:00.

51.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

51.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

Table 51-17: 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 the parameter to take effect, a device reset is required.
Delay After Reset [sec] 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

51.3.2 SIP Test Call Parameters

The SIP Signaling Test Call parameters are described in the table below.

Table 51-18: SIP Test Call Parameters

Parameter	Description
Test Call DTMF String 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.
Test Call ID 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: The parameter is only for testing incoming calls destined to this prefix number.
Test Call Table	

Parameter	Description
Test Call Table configure system > test-call > test- call-table [Test_Call]	Defines Test Call rules. <pre>[Test_Call] FORMAT Test_Call_Index = Test_Call_EndpointURI, Test_Call_CalledURI, Test_Call_RouteBy, Test_Call_IPGroupName, Test_Call_DestAddress, Test_Call_DestTransportType, Test_Call_SIPInterfaceName, 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]</pre> For a description of the table, see "Configuring Test Call Endpoints" on page 699.

51.3.3 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table 51-19: Syslog, CDR and Debug Parameters

Parameter	Description
Enable Syslog sys log [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.
Syslog Server IP Address sys log-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.
Syslog Server Port sys log-port [SyslogServerPort]	Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514.
CDR Server IP Address cdr-srvr-ip-addr [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).

Parameter	Description
CDR Report Level 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, the 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).
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.
configure voip > sip-definition settings > time- zone-format [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.
Local Storage Max File Size	Defines the size (in kilobytes) of each stored CDR file. Once the file size is reached, the device creates a new file for subsequent CDRs, and so

Parameter	Description
configure voip > services cdr > cdr-local-max-file-size [CDRLocalMaxFileSize]	on. The valid value is 100 to 10000. The default is 1024.
Local Storage Max Number Of Files configure voip > services cdr > cdr-local-max-files [CDRLocalMaxNomOfFiles]	Defines the maximum number of stored CDR files. If the maximum number is reached, the device replaces (overwrites) the oldest created file with a subsequent new file, and so on. The valid value is 2 to 4096. The default is 5.
Local Storage File Creation Interval configure voip > services cdr > cdr-local-interval [CDRLocalInterval]	Defines how often (in minutes) the device creates a new CDR file. For example, if configured to 60, it creates a new file every hour. This occurs even if the maximum configured file size has not been reached (see the CDRLocalMaxFileSize parameter). However, if the maximum configured file size has been reached and the interval configured by the parameter has not been reached, a new CDR file is created. The valid value is 2 to 1440. The default is 60.
configure system > cdr > non-call-cdr-rprt [EnableNonCallCdr]	Enables creation of CDR messages for non-call SIP dialogs (such as SUBSCRIBE, OPTIONS, and REGISTER). [0] = (Default) Disable [1] = Enable
Debug Level configure system/logging/debug-level [GwDebugLevel]	Enables Syslog debug reporting and logging level. [0] No Debug = (Default) Debug is disabled and Syslog messages are not sent. [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.
Syslog Optimization 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 MaxBundleSyslogLength parameter.
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: The parameter is applicable only if the GwDebugLevel parameter is enabled.
Syslog CPU Protection configure system/logging/syslog-cpu-protection	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

Parameter	Description
[SyslogCpuProtection]	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)
Debug Level High Threshold 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.
Syslog Facility Number [SyslogFacility]	Defines the Facility level (0 through 7) of the device's Syslog messages, according to RFC 3164. This allows you to identify Syslog messages generated by the device. This is useful, for example, if you collect the device's and other equipments' Syslog messages, at one single server. The device's Syslog messages can easily be identified and distinguished from other Syslog messages by its Facility level. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level. [16] = (Default) local use 0 (local0) [17] = local use 1 (local1) [18] = local use 2 (local2) [19] = local use 3 (local3) [20] = local use 4 (local4) [21] = local use 5 (local5) [22] = local use 6 (local6) [23] = local use 7 (local7)
CDR Syslog Sequence Number cdr-seq-num [CDRSyslogSeqNum]	Enables or disables the inclusion of the sequence number (S=) in CDR Syslog messages. [0] Disable [1] Enable (default)
Activity Types to Report via Activity Log Messages config-system >	Defines the operations (activities) performed in the Web interface that are reported to a Syslog server. [pvc] Parameters Value Change = Changes made on-the-fly to

Parameter	Description
logging > activity-log [ActivityListToLog]	parameters and tables, and Configuration file load. Note that the <i>ini</i> file parameter, EnableParametersMonitoring can also be used to set this option. ▪ [afl] Auxiliary Files Loading = Loading of Auxiliary files. ▪ [dr] Device Reset = Resetting of the 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 the 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 log in to 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. ▪ [cli] = CLI commands entered by the user. ▪ [ae] Action Executed = Logs user actions that are not related to parameter changes. The actions can include, for example, file uploads, file delete, lock-unlock maintenance actions, LDAP clear cache, register-unregister, and start-stop trunk. In the Web, these actions are typically done by clicking a button (e.g., the LOCK button). Note: For the <i>ini</i> file parameter, enclose values in single quotation marks, for example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc'.
Activity Trap activity-trap [EnableActivityTrap]	Enables the device to send an SNMP trap to notify of Web user activities in the Web interface. The activities to report are configured by the ActivityListToLog parameter. ▪ [0] Disable (default) ▪ [1] Enable
[EnableParametersMonitoring]	Enables the monitoring, through Syslog messages, of parameters that are modified on-the-fly. ▪ [0] = (Default) Disable ▪ [1] = Enable
Debug Recording Destination IP configure system >	Defines the IP address of the server for capturing debug recording.

Parameter	Description
logging > dbg-rec-dest-ip [DebugRecordingDestIP]	
Debug Recording Destination Port configure system > logging > dbg-rec-dest-port [DebugRecordingDestPort]	Defines the UDP port of the server for capturing debug recording. The default is 925.
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.
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	
Logging Filters Table configure system > logging > logging-filters [LoggingFilters]	The table 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_LogDestination, LoggingFilters_CaptureType, LoggingFilters_Mode; [\LoggingFilters] For a detailed description of the table, see "Configuring Log Filter Rules" on page 677.
SBC CDR Format Table	
SBC CDR Format Table configure voip > services cdr > cdr-format sbc-cdr-format [SBCCDRFormat]	The table defines CDR customization rules for SBC calls. The format of the ini file table parameter is: [SBCCDRFormat] FORMAT SBCCDRFormat_Index = SBCCDRFormat_CDRTYPE, SBCCDRFormat_ColumnType, SBCCDRFormat_Title, SBCCDRFormat_RadiusType, SBCCDRFormat_RadiusID; [\SBCCDRFormat] For a detailed description of the table, see Customizing CDRs for SBC Calls on page 660.

51.3.4 Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

Table 51-20: 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 the 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. 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.

51.4 HA Parameters

The High Availability (HA) parameters are described in the table below.

Table 51-21: HA Parameters

Parameter	Description
HA Device Name configure system > high-availability > unit-id-name [HAGUnitIdName]	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 configure system > high-availability > remote-address [HARemoteAddress]	Defines the Maintenance interface address of the redundant device in the HA system. By default, no value is defined.
HA Revertive configure system > high-availability > revertive-mode [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 556.
HA Priority configure system > high-availability > priority [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).
HA Monitoring Parameters	
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
HA Network Reachability Destination Address [HAPingDestination]	Defines the IP address of the destination that the device pings. The default is 0.0.0.0.

Parameter	Description
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.

51.5 Security Parameters

This subsection describes the device's security parameters.

51.5.1 General Security Parameters

The general security parameters are described in the table below.

Table 51-22: General Security Parameters

Parameter	Description
Firewall Table	
Internal Firewall Parameters configure voip > access-list [AccessList]	The table defines the device's access list (firewall), which defines network traffic filtering rules. The format of the ini file table parameter is: [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 the table, see "Configuring Firewall Settings" on page 169.
Media Latching	
Inbound Media Latch Mode 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

Parameter	Description
	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.
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 [TimeoutToRelatchRTPMsec]	Defines a period (msec) during which if no packets are received from the current RTP session, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch SRTP [TimeoutToRelatchSRTPMsec]	Defines a period (msec) during which if no packets are received from the current SRTP session, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch Silence [TimeoutToRelatchSilenceMsec]	Defines a period (msec) during which if no packets are received from the current RTP/SRTP session and the channel is in silence mode, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch RTCP [TimeoutToRelatchRTCPMsec]	Defines a period (msec) during which if no packets are received from the current RTCP session, the channel can re-latch onto another RTCP stream. The valid range is any value from 0. The default is 10,000.

Parameter	Description
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.

51.5.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

Table 51-23: HTTPS Parameters

Parameter	Description
Secured Web Connection (HTTPS) 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 the parameter to take effect, a device reset is required.
https-port [HTTPSPort]	Defines the local Secured HTTPS port of the device. The 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 the parameter to take effect, a device reset is required.
HTTPS Cipher String https-cipher-string [HTTPSCipherString]	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). For the valid range values, refer to URL http://www.openssl.org/docs/apps/ciphers.html . The default is 'RC4:EXP' (Export encryption algorithms). For example, use 'ALL' for all ciphers suites (e.g., for ARIA encryption for TLS). The only ciphers available are RC4 and DES, and the cipher bit strength is limited to 56 bits. Notes: - For the parameter to take effect, a device reset is required. - If the installed Software License Key includes the Strong Encryption feature, the default of the 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.
Requires Client Certificates for HTTPS connection req-client-cert [HTTPSRequireClientCertificate]	Enables the requirement of client certificates 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 the parameter to take effect, a device reset is required. - For a description on implementing client certificates, see "TLS for Remote Device Management" on page 121.

51.5.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

Table 51-24: SRTP Parameters

Parameter	Description
Media Security media-security- enable [EnableMediaSecurity]	Enables Secure Real-Time Transport Protocol (SRTP). ▪ [0] Disable (default) ▪ [1] Enable Note: For the parameter to take effect, a device reset is required.
Master Key Identifier (MKI) Size 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
Symmetric MKI Negotiation 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
Offered SRTP Cipher Suites 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: The 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.
configure voip > sbc general- setting > sbc- dtls-mtu [SbcDtIsMtu]	Defines the maximum transmission unit (MTU) size for the DTLS handshake. The device does not attempt to send handshake packets that are larger than the configured value. Adjusting the MTU is useful when there are network constraints on the size of packets that can be sent. The valid value range is 228 to 1500. The default is 1500.

Parameter	Description
Authentication On Transmitted RTP Packets RTP-authentication-disable-tx [RTPAuthenticationDisableTx]	Enables authentication on transmitted RTP packets in a secured RTP session. [0] Enable (default) [1] Disable
Encryption On Transmitted RTP Packets RTP-encryption-disable-tx [RTPEncryptionDisableTx]	Enables encryption on transmitted RTP packets in a secured RTP session. [0] Enable (default) [1] Disable
Encryption On Transmitted RTCP Packets 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.
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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.

51.5.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

Table 51-25: TLS Parameters

Parameter	Description
TLS Contexts Table	
TLS Contexts Table configure system > tls # [TLSContexts]	Defines SSL/TLS certificates. The format of the ini file table parameter is as follows: [TLSContexts] FORMAT TLSContexts_Index = TLSContexts_Name, TLSContexts_TLSVersion, TLSContexts_ServerCipherString, TLSContexts_ClientCipherString, TLSContexts_OcspEnable, TLSContexts_OcspServerPrimary, TLSContexts_OcspServerSecondary, TLSContexts_OcspServerPort, TLSContexts_OcspDefaultResponse; [\TLSContexts] For a detailed description of the table, see "Configuring TLS Certificate Contexts" on page 109.
TLS Client Re-Handshake Interval 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).
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 the parameter to take effect, a device reset is required. ▪ This feature can be configured per SIP Interface (see "Configuring SIP Interfaces" on page 342). ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName.
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 the parameter is not

Parameter	Description
	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 the parameter to [2] (Server & Client), for this functionality to operate you also need to set the SIPRequireClientCertificate parameter to [1] (Enable).
TLS Client Verify Server Certificate 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.
Strict Certificate Extension Validation require-strict-cert [RequireStrictCert]	Enables the validation of the extensions (keyUsage and extenedKeyUsage) 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
TLS Remote Subject Name tls-rmt-subs-name [TLSRemoteSubjectName]	Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. If the CN uses a domain name, the certificate can also use wildcards ("*") to replace parts of the domain name. The valid range is a string of up to 49 characters. Note: The parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.
TLS Expiry Check Start 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.
TLS Expiry Check Period 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.

51.5.5 SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

Table 51-26: SSH Parameters

Parameter	Description
Enable SSH Server ssh [SSHServerEnable]	Enables the device's embedded SSH server. ▪ [0] Disable (default) ▪ [1] Enable
Server Port ssh-port [SSHServerPort]	Defines the port number for the embedded SSH server. Range is any valid port number. The default port is 22.
SSH Admin Key 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.
Require Public Key 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.
Max Payload Size 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.
Max Binary Packet Size ssh-max-binary-packet-size [SSHMaxBinaryPacketSize]	Defines the maximum packet size (in bytes) for SSH packets. The valid value is 582 to 35000. The default is 35000.
Maximum SSH Sessions ssh-max-sessions [SSHMaxSessions]	Defines the maximum number of simultaneous SSH sessions. The valid range is 1 to 5. The default is 5 sessions.
Enable Last Login Message 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.
Max Login Attempts 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

Parameter	Description
	connections.

51.5.6 IDS Parameters

The Intrusion Detection System (IDS) parameters are described in the table below.

Table 51-27: IDS Parameters

Parameter	Description
Intrusion Detection System (IDS) enable-ids [EnableIDS]	Enables the IDS feature. [0] Disable (default) [1] Enable Note: For the parameter to take effect, a device reset is required.
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	
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 the table, see "Configuring IDS Policies" on page 176.
IDS Rule Table	
IDS Rule Table [IDSRule]	Defines rules for IDS Policies. The format of the ini file parameter is: [IDSRule] FORMAT IDSRule_Index = IDSRule_Policy, IDSRule_RuleID, IDSRule_Reason, IDSRule_ThresholdScope, IDSRule_ThresholdWindow, IDSRule_MinorAlarmThreshold, IDSRule_MajorAlarmThreshold, IDSRule_CriticalAlarmThreshold, IDSRule_DenyThreshold, IDSRule_DenyPeriod; [\IDSRule] For a detailed description of the table, see "Configuring IDS Policies" on page 176.
IDS Match Table	

Parameter	Description
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 the table, see "Assigning IDS Policies" on page 180.

51.5.7 OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

Table 51-28: OCSP Parameters

Parameter	Description
Enable OCSP Server enable [OCSPEnable]	Enables or disables certificate checking using OCSP. ▪ [0] Disable (default) ▪ [1] Enable For a description of OCSP, see Configuring Certificate Revocation Checking (OCSP).
Primary Server IP server-ip [OCSPServerIP]	Defines the IP address of the OCSP server. The default IP address is 0.0.0.0.
Secondary Server IP secondary-server-ip [OCSPSecondaryServerIP]	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
Server Port server-port [OCSPServerPort]	Defines the OCSP server's TCP port number. The default port number is 2560.
Default Response When Server Unreachable default-response [OCSPDefaultResponse]	Determines whether the device allows or rejects peer certificates when the OCSP server cannot be contacted. ▪ [0] Reject (default) ▪ [1] Allow

51.6 Quality of Experience Parameters

The Quality of Experience (QoE) parameters are described in the table below.

Table 51-29: Quality of Experience Parameters

Parameter	Description
SEM Parameters	

Parameter	Description
Server IP configure voip/qoe configuration/server-ip [QOEServerIP]	Defines the IP address of the primary Session Experience Manager (SEM) server to where the quality experience reports are sent. Note: For the parameter to take effect, a device reset is required.
Redundant Server IP 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 the parameter to take effect, a device reset is required.
Interface Name 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 the 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.
QOE TLS Context Name configure voip/qoe configuration/tls-context-name [QoETLSContextName]	Selects a TLS Context (configured in the TLS Contexts table) for the TLS connection with the SEM server. The valid value is a string representing the name of the TLS Context as configured in the 'Name' field of the TLS Contexts table. The default is the default TLS Context (ID 0).
QoE Report Mode report-mode [QoeReportMode]	Defines at what stage of the call the device sends the QoE data of the call to the SEM server. [0] Report QoE During Call (default) [1] Report QoE At End Call Note: If a QoE traffic overflow between SEM and the device occurs, the device sends the QoE data only at the end of the call, regardless of the settings of the parameter.
Quality of Experience Profile Table	
Quality of Experience Profile configure voip/qoe qoe-profile [QOEProfile]	The table 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 the table, see "Configuring Quality of Experience Profiles" on page 313.
Quality of Experience Color Rules Table	

Parameter	Description
Quality of Experience Color Rules configure voip/qoe qoe-profile qoe-color-rules [QOECOLORRules]	The table defines Quality of Experience Color Rules. The format of the ini file table parameter is as follows: [QOECOLORRules] FORMAT QOECOLORRules_Index = QOECOLORRules_QoeProfile, QOECOLORRules_ColorRuleIndex, QOECOLORRules_monitoredParam, QOECOLORRules_direction, QOECOLORRules_profile, QOECOLORRules_GreenYellowThreshold, QOECOLORRules_GreenYellowHysteresis, QOECOLORRules_YellowRedThreshold, QOECOLORRules_YellowRedHysteresis; [QOECOLORRules] For a detailed description of the table, see "Configuring Quality of Experience Profiles" on page 313.
Bandwidth Profile Table	
Bandwidth Profile configure voip/qoe bw-profile [BWProfile]	The table defines Bandwidth Profiles. The format of the ini file table parameter is as follows: [BWProfile] FORMAT BWProfile_Index = BWProfile_Name, BWProfile_EgressAudioBandwidth, BWProfile_IngressAudioBandwidth, BWProfile_EgressVideoBandwidth, BWProfile_IngressVideoBandwidth, BWProfile_TotalEgressBandwidth, BWProfile_TotalIngressBandwidth, BWProfile_WarningThreshold, BWProfile_hysteresis, BWProfile_GenerateAlarms; [BWProfile] For a detailed description of the table, see "Configuring Bandwidth Profiles" on page 317. Note: For the parameter to take effect, a device reset is required.
Media Enhancement Profile Table	
Media Enhancement Profile configure voip/qoe media-enhancement [MediaEnhancementProfile]	The table defines Media Enhancement Profiles. The format of the ini file table parameter is as follows: [MediaEnhancementProfile] FORMAT MediaEnhancementProfile_Index = MediaEnhancementProfile_ProfileName; [MediaEnhancementProfile] For a detailed description of the table, see "Configuring Media Enhancement Profiles" on page 320.
Media Enhancement Rules Table	
Media Enhancement Rules configure voip/qoe media-enhancement-rules [MediaEnhancementRules]	The table defines Media Enhancement Rules. The format of the ini file table parameter is as follows: [MediaEnhancementRules] FORMAT MediaEnhancementRules_Index = MediaEnhancementRules_MediaEnhancementProfile, MediaEnhancementRules_RuleIndex, MediaEnhancementRules_Trigger, MediaEnhancementRules_Color, MediaEnhancementRules_ActionRule, MediaEnhancementRules_ActionValue; [MediaEnhancementRules] For a detailed description of the table, see "Configuring Media Enhancement Profiles" on page 320.

51.7 Control Network Parameters

51.7.1 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

Table 51-30: Proxy, Registration and Authentication SIP Parameters

Parameter	Description
IP Group Table	
IP Group Table configure voip > voip-network ip- group [IPGroup]	This table configures IP Groups. The format of the ini file table parameter is: <pre>[IPGroup] FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Name, IPGroup_ProxySetName, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_SipReRoutingMode, IPGroup_AlwaysUseRouteTable, IPGroup_SRDName, IPGroup_MediaRealm, IPGroup_ClassifyByProxySet, IPGroup_ProfileName, 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_MsgManUserDef1, IPGroup_MsgManUserDef2, IPGroup_SIPConnect, IPGroup_SBCPSAPMode, IPGroup_DTLSContext, IPGroup_CreatedByRoutingServer, IPGroup_UsedByRoutingServer, IPGroup_SBCOperationMode, IPGroup_SBCRouteUsingRequestURIPort; [/IPGroup]</pre> For a description of the table, see "Configuring IP Groups" on page 348. Note: For the parameter to take effect, a device reset is required.
Account Table	
Account Table configure voip > sip-definition account [Account]	Defines user accounts for registering and/or authenticating (digest) IP Groups (e.g., an IP-PBX) with a Serving IP Group (e.g., a registrar server). The format of the ini file table parameter is as follows: <pre>[Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroupName, Account_ServingIPGroupName, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType; [/Account]</pre> For a detailed description of the table, see "Configuring Registration Accounts" on page 369.
Proxy Registration Parameters	
Proxy Name	Defines the Home Proxy domain name. If specified, this name is used as

Parameter	Description
proxy-name [ProxyName]	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: The parameter functions together with the UseProxyIPasHost parameter.
Use Proxy IP as Host 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 the 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 the 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.
Redundancy Mode 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.
Proxy IP List Refresh Time 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.
Always Use Proxy always-use-proxy [AlwaysSendToProxy]	Determines whether the device sends SIP messages and responses through a Proxy server. [0] Disable = (Default) Use standard SIP routing rules. [1] Enable = All SIP messages and responses are sent to the Proxy server. Note: The parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1).
DNS Query Type 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,

Parameter	Description
	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.
Proxy DNS Query Type proxy-dns-query [ProxyDNSQueryType]	Global parameter that defines the DNS query record type for resolving the Proxy server's configured domain name (FQDN) into an IP address. ▪ [0] A-Record (default) = A-record DNS query. ▪ [1] SRV = If the Proxy IP address parameter contains a domain name without port definition (e.g., ProxyIP = domain.com), an SRV query is performed. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Thus, if the first SRV query returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed. ▪ [2] NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. Notes: ▪ This functionality can be configured per Proxy Set in the Proxy Set table (see "Configuring Proxy Sets" on page 360). ▪ When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Use Gateway Name for OPTIONS 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). To configure the "gateway name", use 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

Parameter	Description
	messages. [2] Server = Device's IP address is used in the From and To headers in keep-alive OPTIONS messages.
Password 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'.
Cnonce 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'.
Mutual Authentication Mode 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.
Challenge Caching Mode challenge-caching [SIPChallengeCachingMode]	Enables local caching of SIP message authorization challenges from Proxy servers. The device sends the first request to the Proxy without authorization. The Proxy sends a 401/407 response with a challenge for credentials. The device saves (caches) the response for further uses. The device sends a new request with the appropriate credentials. Subsequent requests to the Proxy are automatically sent with credentials (calculated from the saved challenge). If the Proxy doesn't accept the new request and sends another challenge, the old challenge is replaced with the new one. One of the benefits of the feature is that it may reduce the number of SIP messages transmitted through the network. [0] None = (Default) Challenges are not cached. Every new request is sent without preliminary authorization. If the request is challenged, a new request with authorization data is sent. [1] INVITE Only = Challenges issued for INVITE requests are cached. This prevents a mixture of REGISTER and INVITE authorizations. [2] Full = Caches all challenges from the proxies. Note: - Challenge caching is used with all proxies and not only with the active one. - The challenge can be cached per Account or per user whose credentials are known through the User Info table.
Proxy Address Table	
Proxy IP Table configure voip > voip-network proxy-ip [ProxyIP]	The table defines proxy addresses per Proxy Set. The format of the ini file table parameter is as follows: [ProxyIP] FORMAT ProxyIp_Index = ProxyIp_ProxySetId, ProxyIp_ProxyIpIndex, ProxyIp_IpAddress, ProxyIp_TransportType; [\ProxyIP] For a description of the table, see "Configuring Proxy Sets" on page 360.
Proxy Sets Table	

Parameter	Description
Proxy Set Table configure voip > voip-network proxy- set [ProxySet]	Defines the Proxy Sets. 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_SRDName, ProxySet_ClassificationInput, ProxySet_TLSContextName, ProxySet_ProxyRedundancyMode, ProxySet_DNSResolveMethod, ProxySet_KeepAliveFailureResp, ProxySet_GWIPv4SIPInterfaceName, ProxySet_SBCIPv4SIPInterfaceName, ProxySet_GWIPv6SIPInterfaceName, ProxySet_SBCIPv6SIPInterfaceName, ProxySet_MinActiveServersLB, ProxySet_SuccessDetectionRetries, ProxySet_SuccessDetectionInterval, ProxySet_FailureDetectionRetransmissions; [\ProxySet]</pre> For a description of the table, see "Configuring Proxy Sets" on page 360.
Registrar Parameters	
Registration Time registration-time [RegistrationTime]	Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. The 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.
Re-registration Timing [%] 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 the 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). Note: The parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0.
Registration Retry Time 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.
Registration Time Threshold registration-time- thres [RegistrationTimeThreshold]	Defines a threshold (in seconds) for re-registration timing. If the 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.
ReRegister On Connection Failure reg-on-conn-failure	Enables the device to perform SIP re-registration upon TCP/TLS connection failure. [0] Disable (default)

Parameter	Description
[ReRegisterOnConnectionFailure]	[1] Enable
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".
Add Empty Authorization Header 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="", response="e56131d19580cd833064787ecc"</pre> Note: This registration header is according to the IMS 3GPP TS24.229 and PKT-SP-24.220 specifications.
Add initial Route Header 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>

Parameter	Description
[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 between 160 to 200 seconds. This prevents an "avalanche" of keep-alive by multiple SIP UAs to a specific server.
Max Generated Register Rate configure voip > sip- definition settings > max- gen-reg-rate [MaxGeneratedRegistersRate]	Defines the maximum number of user register requests (REGISTER messages) that the device sends (to a proxy or registrar server) at a user-defined rate configured by the GeneratedRegistersInterval parameter. The parameter is useful in that it may be used to prevent an overload on the device's CPU caused by sending many registration requests at a given time. The valid value is 30 to 300 register requests per second. The default is 150. For configuration examples, see the description of the GeneratedRegistersInterval parameter.

Parameter	Description
Generated Registers interval gen-reg-int [GeneratedRegistersInterval]	Defines the rate (in seconds) at which the device sends user register requests (REGISTER messages). The parameter is based on the maximum number of REGISTER messages that can be sent at this rate, configured by the MaxGeneratedRegistersRate parameter. The valid value is 1 to 5. The default is 1. Configuration examples: ▪ If you configure the MaxGeneratedRegistersRate parameter to 100 and the GeneratedRegistersInterval to 5, the device sends a maximum of 20 REGISTER messages per second (i.e., 100 messages divided by 5 sec; 100 per 5 seconds). ▪ If you configure the MaxGeneratedRegistersRate parameter to 100 and the GeneratedRegistersInterval to 1, the device sends a maximum of a 100 REGISTER messages per second.

51.7.2 Network Application Parameters

The SIP network application parameters are described in the table below.

Table 51-31: SIP Network Application Parameters

Parameter	Description
SRD Table	
SRD Table configure voip > voip-network srd [SRD]	Defines Signaling Routing Domains (SRD). The format of the ini file table parameter is as follows: [SRD] FORMAT SRD_Index = SRD_Name, SRD_IntraSRDMediaAnchoring, SRD_BlockUnRegUsers, SRD_MaxNumOfRegUsers, SRD_EnableUnAuthenticatedRegistrations, SRD_SharingPolicy, SRD_UsedByRoutingServer, SRD_SBCOperationMode, SRD_SBCRoutingPolicyName; [\SRD] For a detailed description of the table, see "Configuring SRDs" on page 332.
SIP Interface Table	
SIP Interface Table configure voip > voip-network sip- interface [SIPInterface]	Defines SIP Interfaces. The format of the ini file table parameter is as follows: [SIPInterface] FORMAT SIPInterface_Index = SIPInterface_InterfaceName, SIPInterface_NetworkInterface, SIPInterface_ApplicationType, SIPInterface_UDPPort, SIPInterface_TCPPort, SIPInterface_TLSPort, SIPInterface_SRDName, SIPInterface_MessagePolicyName, SIPInterface_TLSContext, SIPInterface_TLSMutualAuthentication, SIPInterface_TCPKeepaliveEnable, SIPInterface_ClassificationFailureResponseType, SIPInterface_PreClassificationManSet, SIPInterface_EncapsulatingProtocol, SIPInterface_MediaRealm, SIPInterface_SBCDirectMedia, SIPInterface_BlockUnRegUsers, SIPInterface_MaxNumOfRegUsers, SIPInterface_EnableUnAuthenticatedRegistrations, SIPInterface_UsedByRoutingServer; [\SIPInterface] For a detailed description of the table, see "Configuring SIP Interfaces" on page 342.
[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	
NAT Translation Table configure voip > voip-network NATTranslation [NATTranslation]	Defines NAT rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses. The format of the ini file table parameter is as follows: <pre>[NATTranslation] FORMAT NATTranslation_Index = NATTranslation_SrcInterfaceName, NATTranslation_TargetIPAddress, NATTranslation_SourceStartPort, NATTranslation_SourceEndPort, NATTranslation_TargetStartPort, NATTranslation_TargetEndPort; [\NATTranslation]</pre> For a detailed description of the table, see "Configuring NAT Translation per IP Interface" on page 158.
Media Realm Table	
Media Realm Table configure voip > voip-network realm [CpMediaRealm]	Defines Media Realms. 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 the table, see "Configuring Media Realms" on page 323.
Remote Media Subnet Table	
Remote Media Subnet configure voip > voip-network realm remote-media-subnet [SubRealm]	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 the table, see "Configuring Remote Media Subnets" on page 326.
Media Realm Extension Table	

Parameter	Description
Media Realm Extension [MediaRealmExtension]	Defines Media Realm Extensions. The format of the ini file table parameter is as follows: <pre>[MediaRealmExtension] FORMAT MediaRealmExtension_Index = MediaRealmExtension_MediaRealmIndex, MediaRealmExtension_ExtensionIndex, MediaRealmExtension_IPv4IF, MediaRealmExtension_IPv6IF, MediaRealmExtension_PortRangeStart, MediaRealmExtension_PortRangeEnd, MediaRealmExtension_MediaSessionLeg; [\MediaRealmExtension]</pre> For a detailed description of the table, see "Configuring Media Realm Extensions" on page 329.

51.8 General SIP Parameters

The general SIP parameters are described in the table below.

Table 51-32: General SIP Parameters

Parameter	Description
Max Call Duration (min) mx-call-duration [MaxCallDuration]	Defines the maximum duration (in minutes) of a call. If this duration is reached, the device terminates the call. This feature is useful for ensuring available resources for new calls, by ensuring calls are properly terminated. The valid range is 0 to 35,791. The default is 0 (i.e., no limitation).
Send reject on overload 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 the parameter is disabled (i.e., 503 is not sent), the device still discards the new SIP dialog-initiating requests when the CPU is overloaded.
SIP 408 Response upon non-INVITE enbl-non-inv-408 [EnableNonInvite408Repl y]	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.
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.
Reject Cancel after Connect reject-cancel- after-connect [RejectCancelAfterConnec t]	Enables or disables the device to accept or reject SIP CANCEL requests received after the receipt of a 200 OK in response to an INVITE (i.e., call established). According to the SIP standard, a CANCEL can be sent only during the INVITE transaction (before 200 OK), and once a 200 OK response is received the call can be rejected only by a BYE request. ▪ [0] Disable = (Default) Accepts a CANCEL request received during the INVITE transaction by sending a 200 OK response and terminates the call session.

Parameter	Description
	[1] Enable = Rejects a CANCEL request received during the INVITE transaction by sending a SIP 481 (Call/Transaction Does Not Exist) response and maintains the call session.
Verify Received RequestURI verify-rcvd-requri [VerifyReceeedRequestU ri]	Enables the device to reject SIP requests (such as ACK, BYE, or re-INVITE) whose user part in the Request-URI is different from the user part received in the Contact header of the last sent SIP request. [0] Disable = (Default) Even if the user is different, the device accepts the SIP request. [1] Enable = If the user is different, the device rejects the SIP request (BYE is responded with 481; re-INVITE is responded with 404; ACK is ignored).
Max Number of Active Calls 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).
Number of Calls Limit [IpProfile_CallLimit,]	Defines the maximum number of concurrent calls per IP Profile (see "Configuring IP Profiles" on page 391).
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; </pre>

Parameter	Description
	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.227.005 Reason: Q.850 ;cause=16 ;text="local" Content-Length: 0
Enable Early Media early-media [EnableEarlyMedia]	Global parameter that enables the Early Media feature for sending media (e.g., ringing) before the call is established. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEarlyMedia). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
Session Expires Disconnect Time 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.
[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 (host part only) is used. The valid range is a 30-character string. By default, this value is not defined.
Fax Signaling Method 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 the parameter and

Parameter	Description
	for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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: The 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).
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: <pre>a=rtpmap:120 NoAudio/8000\r\n</pre> Note: For incoming SDP offers, NoAudio is always supported.
SIP Transport Type app-sip-transport-	Determines the default transport layer for outgoing SIP calls initiated by the device.

Parameter	Description
type [SIPTransportType]	[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 the parameter is also used by the SAS application as the default transport layer for outgoing SIP calls.
Display Default SIP Port 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
Enable SIPS 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 the parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails.
Enable TCP Connection Reuse 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.
Fake TCP alias 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.
Reliable Connection	Enables setting of all TCP/TLS connections as persistent and therefore,

Parameter	Description
Persistent Mode reliable-conn- persistent [ReliableConnectionPersistentMode]	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 the parameter.
TCP Timeout 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.
SIP Destination Port 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.
Use user=phone in SIP URL 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.
Use user=phone in From Header 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.
Use Tel URI for Asserted Identity 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:'
Enable GRUU 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

Parameter	Description
	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. The 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.
User-Agent Information 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.227.005</pre> If not configured, the default string, <AudioCodes product-name>/software version' is used, for example: <pre>User-Agent : Audiocodes-Sip-Gateway-Mediant 2600 SBC/v.6.80A.227.005</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

Parameter	Description
	the SIP header, which may cause problems).
SDP Session Owner 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>
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.
Subject 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.
Multiple Packetization Time Format 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 the 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.
3xx Behavior 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 INVITE message. ▪ [1] Redirect = Use the same call identifiers.
Retry-After Time	Defines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device.

Parameter	Description
retry-afttr-time [RetryAfterTime]	The time range is 0 to 3,600. The default is 0.
Fake Retry After 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 the 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.
Enable P-Associated-URI Header p-associated-uri-hdr [EnablePAssociatedURIHeader]	Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From/P-Asserted-Identity headers value. ▪ [0] Disable (default) ▪ [1] Enable Note: P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).
Source Number Preference 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: ▪ 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

Parameter	Description
	is set to 'id', the calling number is assumed restricted.
Enable Reason Header reason-header [EnableReasonHeader]	Enables the usage of the SIP Reason header. ▪ [0] Disable ▪ [1] Enable (default)
Gateway Name 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. ▪ The 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.
Enable Delayed Offer delayed-offer [EnableDelayedOffer]	Determines whether the device sends the initial INVITE message with or without an SDP. Sending the first INVITE without SDP is typically done by clients for obtaining the far-end's full list of capabilities before sending their own offer. (An alternative method for obtaining the list of supported capabilities is by using SIP OPTIONS, which is not supported by every SIP agent.) ▪ [0] Disable = (Default) The device sends the initial INVITE message with an SDP. ▪ [1] Enable = The device sends the initial INVITE message without an SDP.
[DisableCryptoLifeTimeInSDP]	Enables the device to send "a=crypto" lines without the lifetime parameter in the SDP. For example, if the SDP contains "a=crypto:12 AES_CM_128_HMAC_SHA1_80 inline:hhQe10yZRcRcpIFPKH5xYY9R1de37ogh9G1MpvNp 2^31", it removes the lifetime parameter "2^31". ▪ [0] Disable (default) ▪ [1] Enable
Enable Contact Restriction 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.
Enable User-Information Usage	Enables the usage of the User Information, which is loaded to the idevice> in the User Information Auxiliary file. For more information on User Information, see "User Information File" on page 591.

Parameter	Description
user-inf-usage [EnableUserInfoUsage]	▪ [0] Disable (default) ▪ [1] Enable Note: For the 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 'silecesupp' attribute. ▪ [1] = Handle incoming Re-INVITE messages that include the 'silencesupp:off' attribute in the SDP as a request to switch to the Voice-Band-Data (VBD) mode. In addition, the device includes the attribute 'a=silencesupp:off' in its SDP offer. Note: The 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.
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
prog-ind-2ip	Note: If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
[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").

Parameter	Description
	The re-INVITE is sent immediately upon receipt of the 200 OK (when the call is answered). ▪ [0] = Disable (default) ▪ [1] = Enable Note: The parameter is applicable only if SRTP is used.
[NumberOfActiveDialogs]	Defines the maximum number of concurrent, outgoing SIP REGISTER dialogs. The 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. ▪ The 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'.
Enable Microsoft Extension 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

Parameter	Description
	present in the SDP received from the remote side. [0] Disable (default) [1] Enable
sdp-ecan-frmt [SDPEcanFormat]	Defines the echo canceller format in the outgoing SDP. The 'ecan' attribute is used in the SDP to indicate the use of echo cancellation. [0] = (Default) The 'ecan' attribute appears on the 'a=gpmid' 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: The 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.
First Call Ringback Tone ID 1st-call-rbt-id [FirstCallRBTId]	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 the 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 the 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).
Media IP Version Preference 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 the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 391.
WebSocket Keep-Alive Period configure voip > sip-definition general-settings > websocket-keepalive [WebSocketProtocolKeepAlivePeriod]	Defines how often (in seconds) the device sends ping messages (keep alive) to check whether the WebSocket session with the Web client is still connected. The valid value is 5 to 2000000. The default is 0 (i.e., ping messages are not sent). For more information on WebSocket, see SIP over WebSocket on page 526. Note: The device always replies to WebSocket ping control messages with pong messages.
Retransmission Parameters	
SIP T1 Retransmission Timer t1-re-tx-time [SipT1Rtx]	Defines the time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message. The default is 500. Note: The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx. For INVITE requests, it is multiplied by two for each new retransmitted message. For all other SIP

Parameter	Description
	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.
SIP T2 Retransmission Timer 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.
SIP Maximum RTX 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.
Number of RTX Before Hot-Swap 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. For example, if configured to 3 and no response is received from an IP destination, the device attempts another three times to send the call to the IP destination. If still unsuccessful, it attempts to redirect the call to another IP destination. Note: The parameter is also used for alternative routing (see Alternative Routing Based on IP Connectivity).
SIP Message Manipulations Table	
Message Manipulations configure voip > sbc manipulations message-manipulations [MessageManipulations]	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 the table, see Configuring SIP Message Manipulation on page 377.
Message Policy Table	
Message Policy Table configure voip > sbc message-policy	Defines SIP message policy rules for blocking (blacklist) unwanted incoming SIP messages or allowing (whitelist) receipt of desired messages.

Parameter	Description
[MessagePolicy]	The format of the ini file table parameter is as follows: <pre>[MessagePolicy] FORMAT MessagePolicy_Index = MessagePolicy_Name, MessagePolicy_MaxMessageLength, MessagePolicy_MaxHeaderLength, MessagePolicy_MaxBodyLength, MessagePolicy_MaxNumHeaders, MessagePolicy_MaxNumBodies, MessagePolicy_SendRejection, MessagePolicy_MethodList, MessagePolicy_MethodListType, MessagePolicy_BodyList, MessagePolicy_BodyListType; [/MessagePolicy]</pre> For a detailed description of the table, see Configuring SIP Message Policy Rules.

51.9 Coders and Profile Parameters

The profile parameters are described in the table below.

Table 51-33: Profile Parameters

Parameter	Description
Coders Table / Coders Groups Table	
Coders Table/Coder Group Settings configure voip > coders-and-profiles coders-group [CodersGroup0] [CodersGroup1] [CodersGroup2] [CodersGroup3] [CodersGroup4] [CodersGroup5] [CodersGroup6] [CodersGroup7] [CodersGroup8] [CodersGroup9]	Defines the device's coders. Each group can consist of up to 10 coders. The first Coders Group is the default coder list and the default Coders Group. The format of the ini file table parameter is as follows: <pre>[CodersGroup<0-9>] FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce, CodersGroup0_CoderSpecific; [\CodersGroup<0-9>]</pre> Notes: - For a list of supported coders and a description of the table, see Configuring Default Coders on page 385. - For configuring Coders Groups, see "Configuring Coders Groups" on page 389. - The coder name is case-sensitive.
IP Profile Settings Table	
IP Profile Settings configure voip > coders-and-profiles ip-profile [IPProfile]	Defines the IP Profile table. The format of the ini file table parameter is as follows: <pre>[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,</pre>

Parameter	Description
	IpProfile_SBCDiversioMode, IpProfile_SBCHistoryInfoMode, IpProfile_EnableQSIGTunneling, IpProfile_SBCFaxCodersGroupID, IpProfile_SBCFaxBehavior, IpProfile_SBCFaxOfferMode, IpProfile_SBCFaxAnswerMode, IpProfile_SbcPrackMode, IpProfile_SBCSessionExpiresMode, IpProfile_SBCRemoteUpdateSupport, IpProfile_SBCRemoteReinviteSupport, IpProfile_SBCRemoteDelayedOfferSupport, IpProfile_SBCRemoteReferBehavior, IpProfile_SBCRemote3xxBehavior, IpProfile_SBCRemoteMultiple18xSupport, IpProfile_SBCRemoteEarlyMediaResponseType, IpProfile_SBCRemoteEarlyMediaSupport, IpProfile_EnableSymmetricMKI, IpProfile_MKISize, IpProfile_SBCEnforceMKISize, IpProfile_SBCRemoteEarlyMediaRTP, IpProfile_SBCRemoteSupportsRFC3960, IpProfile_SBCRemoteCanPlayRingback, IpProfile_EnableEarly183, IpProfile_EarlyAnswerTimeout, IpProfile_SBC2833DTMFPayloadType, IpProfile_SBCUserRegistrationTime, IpProfile_ResetSRTPStateUponRekey, IpProfile_AmdMode, IpProfile_SBCReliableHeldToneSource, IpProfile_GenerateSRTPKeys, IpProfile_SBCPlayHeldTone, IpProfile_SBCRemoteHoldFormat, IpProfile_SBCRemoteReplacesBehavior, IpProfile_SBCSDPTimeAnswer, IpProfile_SBCPreferredPTIME, IpProfile_SBCUseSilenceSupp, IpProfile_SBCRTPRedundancyBehavior, IpProfile_SBCPlayRBTToTransferee, IpProfile_SBCRTCPMode, IpProfile_SBCJitterCompensation, IpProfile_SBCRemoteRenegotiateOnFaxDetection, IpProfile_JitterBufMaxDelay, IpProfile_SBCUserBehindUdpNATRegistrationTime, IpProfile_SBCUserBehindTcpNATRegistrationTime, IpProfile_SBCSDPHandleRTCPAttribute, IpProfile_SBCRemoveCryptoLifetimeInSDP, IpProfile_SBCIceMode, IpProfile_SBCRTCPMux, IpProfile_SBCMediaSecurityMethod, IpProfile_SBCHandleXDetect, IpProfile_SBCRTCPFeedback, IpProfile_SBCRemoteRepresentationMode, IpProfile_SBCKeepVIAHeaders, IpProfile_SBCKeepRoutingHeaders, IpProfile_SBCKeepUserAgentHeader, IpProfile_SBCRemoteMultipleEarlyDialogs, IpProfile_SBCRemoteMultipleAnswersMode, IpProfile_SBCDirectMediaTag; [IPProfile] For a description of the table, see "Configuring IP Profiles" on page 391.

51.10 Channel Parameters

This subsection describes the device's channel parameters.

51.10.1 Voice Parameters

The voice parameters are described in the table below.

Table 51-34: Voice Parameters

Parameter	Description
Input Gain input-gain [InputGain]	Global parameter that defines the pulse-code modulation (PCM) input (received) gain control level (in decibels). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_InputGain). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. Note: If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
Voice Volume voice-volume [VoiceVolume]	Global parameter that defines the voice gain control (in decibels). This defines the level of the transmitted signal. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VoiceVolume). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. Note: If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
G726-voice-payload-format [VoicePayloadFormat]	Determines the bit ordering of the G.726 voice payload format. ▪ [0] = (Default) Little Endian ▪ [1] = Big Endian Note: To ensure high voice quality when using G.726, both communicating ends should use the same endianness format. Therefore, when the device communicates with a third-party entity that uses the G.726 voice coder and voice quality is poor, change the settings of the parameter (between Big Endian and Little Endian).
MF Transport Type MF-transport-type [MFTransportType]	Currently, not supported.
Echo Canceller echo-canceller-enable [EnableEchoCanceller]	Global parameter that enables echo cancellation (i.e., echo from voice calls is removed). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEchoCanceller). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. Note: If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.

Parameter	Description
Network Echo Suppressor Enable acoustic-echo-suppressor-enable [AcousticEchoSuppressorSupport]	Enables the network Acoustic Echo Suppressor feature on SBC calls. This feature removes echoes and sends only the near-end's desired speech signal to the network (i.e., to the far-end party). [0] Disable (default) [1] Enable Note: For the parameter to take effect, a device reset is required.
Echo Canceller Type echo-canceller-type [EchoCancellerType]	Defines the echo canceller type. [0] Line echo canceller = (Default) Echo canceller for Tel side. [1] Acoustic Echo suppressor - netw = Echo canceller for IP side.
Attenuation Intensity acoustic-echo-suppressor-attenuation-intensity [AcousticEchoSuppAttenuationIntensity]	Defines the acoustic echo suppressor signals identified as echo attenuation intensity. The valid range is 0 to 3. The default is 0.
Max ERL Threshold - DB acoustic-echo-suppressor-max-ERL [AcousticEchoSuppMaxERLThreshold]	Defines the acoustic echo suppressor maximum ratio between signal level and returned echo from the phone (in decibels). The valid range is 0 to 60. The default is 10.
Min Reference Delay x10 msec acoustic-echo-suppressor-min-reference-delay [AcousticEchoSuppMinRefDelayx10ms]	Defines the acoustic echo suppressor minimum reference delay (in 10-ms units). The valid range is 0 to 40. The default is 0.
Max Reference Delay x10 msec acoustic-echo-suppressor-max-reference-delay [AcousticEchoSuppMaxRefDelayx10ms]	Defines the acoustic echo suppressor maximum reference delay (in 10-ms units). The valid range is 0 to 40. The default is 40 (i.e., 40 x 10 = 400 ms).
echo-canceller-hybrid-loss [ECHybridLoss]	Defines the four-wire to two-wire worst-case Hybrid loss, the ratio between the signal level sent to the hybrid and the echo level returning from the hybrid. [0] = (Default) 6 dB [1] = N/A [2] = 0 dB [3] = 3 dB
echo-canceller-NLP-mode [ECNLPMode]	Enables Non-Linear Processing (NLP) mode for echo cancellation. [0] = (Default) NLP adapts according to echo changes [1] = Disables NLP
echo-canceller-aggressive-NLP [EchoCancellerAggressiveNLP]	Enables the Aggressive NLP at the first 0.5 second of the call. [0] = Disable [1] = (Default) Enable. The echo is removed only in the first half of a second of the incoming IP signal.

Parameter	Description
	Note: For the parameter to take effect, a device reset is required.
number-of-SID-coefficients [RTPSIDCoeffNum]	Defines the number of spectral coefficients added to an SID packet being sent according to RFC 3389. The valid values are [0] (default), [4], [6], [8] and [10].
Answer Detector (AD) Parameters	
Enable Answer Detector [EnableAnswerDetector]	Currently, not supported.
Answer Detector Activity Delay answer-detector-activativity-delay [AnswerDetectorActivityDelay]	Defines the time (in 100-msec resolution) between activating the Answer Detector and the time that the detector actually starts to operate. The valid range is 0 to 1023. The default is 0.
Answer Detector Silence Time [AnswerDetectorSilenceTime]	Currently, not supported.
Answer Detector Redirection [AnswerDetectorRedirection]	Currently, not supported.
Answer Detector Sensitivity answer-detector-sensitivity [AnswerDetectorSensitivity]	Defines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0.

51.10.2 Coder Parameters

The coder parameters are described in the table below.

Table 51-35: Coder Parameters

Parameter	Description
Silk Tx Inband FEC silk-tx-inband-fec [SilkTxInbandFEC]	Enables forward error correction (FEC) for the SILK coder. [0] Disable (default) [1] Enable
Silk Max Average Bit Rate silk-max-average-bitrate [SilkMaxAverageBitRate]	Defines the maximum average bit rate for the SILK coder. The valid value range is 5000 to 30000. The default is 16000. The SILK coder is Skype's default audio codec used for Skype-to-Skype calls.
vbr-coder-header-format [VBRCoderHeaderFormat]	Determines the format of the RTP header for VBR coders. [0] = (Default) Payload only (no header, TOC, or m-factor) - similar to RFC 3558 Header Free format. [1] = Supports RFC 2658 - 1 byte for interleaving header (always 0), TOC, no m-factor. [2] = Payload including TOC only, allow m-factor. [3] = RFC 3558 Interleave/Bundled format.
vbr-coder-hangover [VBRCoderHangover]	Defines the required number of silence frames at the beginning of each silence period when using the VBR coder silence suppression. The range is 0 to 255. The default is 1.
AMR Payload Format [AmrOctetAlignedEnable]	Defines the AMR payload format type. [0] Bandwidth Efficient [1] Octet Aligned (default) Note: The AMR payload type can also be configured per Coder Group (see Configuring Coder Groups on page 389). The Coder Group configuration overrides the parameter.
[AMRCoderHeaderFormat]	Determines the payload format of the AMR header. [0] = Non-standard multiple frames packing in a single RTP frame. Each frame has a CMR and TOC header. [1] = AMR frame according to RFC 3267 bundling. [2] = AMR frame according to RFC 3267 interleaving. [3] = AMR is passed using the AMR IF2 format. Note: Bandwidth Efficient mode is not supported; the mode is always Octet-aligned.

51.10.3 DTMF Parameters

The dual-tone multi-frequency (DTMF) parameters are described in the table below.

Table 51-36: DTMF Parameters

Parameter	Description
DTMF Transport Type DTMF-transport-type [DTMFTransportType]	Determines the DTMF transport type. ▪ [0] Mute DTMF = DTMF digits are removed from the voice stream and are not relayed to remote side. ▪ [2] Transparent DTMF = DTMF digits remain in the voice stream. ▪ [3] RFC 2833 Relay DTMF = (Default) DTMF digits are removed from the voice stream and are relayed to remote side according to RFC 2833. ▪ [7] RFC 2833 Relay Decoder Mute = DTMF digits are sent according to RFC 2833 and muted when received. Note: The parameter is automatically updated if the parameters FirstTxDTMFOption or RxDTMFOption are configured.
DTMF Volume (-31 to 0 dB) DTMF-volume [DTMFVolume]	Defines the DTMF gain control value (in decibels). The valid range is -31 to 0 dB. The default is -11 dB.
DTMF Generation Twist DTMF-generation-twist [DTMFGenerationTwist]	Defines the range (in decibels) between the high and low frequency components in the DTMF signal. Positive decibel values cause the higher frequency component to be stronger than the lower one. Negative values cause the opposite effect. For any parameter value, both components change so that their average is constant. The valid range is -10 to 10 dB. The default is 0 dB. Note: For the parameter to take effect, a device reset is required.
inter-digit-interval [DTMFInterDigitInterval]	Defines the time (in msec) between generated DTMF digits (if FirstTxDTMFOption = 1, 2 or 3). The valid range is 0 to 32767. The default is 100.
[DTMFDigitLength]	Defines the time (in msec) for generating DTMF tones (if FirstTxDTMFOption = 1, 2 or 3). It also configures the duration that is sent in INFO (Cisco) messages. The valid range is 0 to 32767. The default is 100.
default-dtmf-signal-duration [RxDTMFHangOverTime]	Defines the Voice Silence time (in msec) after playing DTMF or MF digits that arrive as Relay. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
digit-hangover-time-tx [TxDTMFHangOverTime]	Defines the Voice Silence time (in msec) after detecting the end of DTMF or MF digits when the DTMF Transport Type is either Relay or Mute. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
NTE Max Duration telephony-events-max-duration [NTEMaxDuration]	Defines the maximum time for sending Named Telephony Events / NTEs (RFC 4733/2833 DTMF relay), regardless of the DTMF signal duration on the other side. The range is -1 to 200,000,000 msec. The default is -1 (i.e., NTE stops only upon detection of an End event).

51.10.4 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

Table 51-37: RTP/RTCP and T.38 Parameters

Parameter	Description
Dynamic Jitter Buffer Minimum Delay jitter-buffer-minimum-delay [DJBufMinDelay]	Global parameter that defines the minimum delay (in msec) of the device's dynamic Jitter Buffer. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_JitterBufMinDelay). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 391. Note: If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
Dynamic Jitter Buffer Optimization Factor jitter-buffer-optimization-factor [DJBufOptFactor]	Global parameter that defines the Dynamic Jitter Buffer frame error/delay optimization factor. You can also configure this functionality per specific calls, using IP Profiles. For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 391. Note: If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
RTP Redundancy Depth RTP-redundancy-depth [RTPRedundancyDepth]	Global parameter that enables the device to generate RFC 2198 redundant packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_RTPRedundancyDepth). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
RFC 2198 Payload Type RTP-redundancy-payload-type [RFC2198PayloadType]	Defines the RTP redundancy packet payload type (according to RFC 2198). The valid value is 96 to 127. The default is 104. Note: The parameter is applicable only if the RTPRedundancyDepth parameter is set to 1.
Packing Factor [RTPPackingFactor]	N/A. Controlled internally by the device according to the selected coder.
RFC 2833 TX Payload Type telephony-events-payload-type-tx [RFC2833TxPayloadType]	Defines the Tx RFC 2833 DTMF relay dynamic payload type for outbound calls. The valid range is 96 to 127. The default is 96. Note: When RFC 2833 payload type negotiation is used (i.e., the parameter FirstTxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.
RFC 2833 RX Payload Type telephony-events-payload-type-rx	Defines the Rx RFC 2833 DTMF relay dynamic payload type for inbound calls. The valid range is 96 to 127. The default is 96.

Parameter	Description
[RFC2833RxBPayloadType]	Note: When RFC 2833 payload type negotiation is used (i.e., the parameter FirstTxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.
[EnableDetectRemoteMACChange]	Determines whether the device changes the RTP packets according to the MAC address of received RTP packets and according to Gratuitous Address Resolution Protocol (GARP) messages. ▪ [0] = Nothing is changed. ▪ [1] = If the device receives RTP packets with a different source MAC address (than the MAC address of the transmitted RTP packets), then it sends RTP packets to this MAC address and removes this IP entry from the device's ARP cache table. ▪ [2] = (Default) The device uses the received GARP packets to change the MAC address of the transmitted RTP packets. ▪ [3] = Options 1 and 2 are used. Notes: ▪ For the parameter to take effect, a device reset is required. ▪ If the device is located in a network subnet which is connected to other gateways using a router that uses Virtual Router Redundancy Protocol (VRRP) for redundancy, then set the parameter to 0 or 2.
RTP 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 more information on configuring the UDP port range, see "Configuring RTP Base UDP Port" on page 201. The range of possible UDP ports is 6,000 to 65,535. The default base UDP port is 6000. Note: For the 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 the parameter to take effect. ▪ For more information on configuring the UDP port range, see Configuring RTP Base UDP Port on page 201.
no-operation-enable [NoOpEnable]	Enables the transmission of RTP or T.38 No-Op packets. ▪ [0] = Disable (default) ▪ [1] = Enable This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.
[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.

Parameter	Description
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 the parameter, ensure that it doesn't cause collision with other payload types.
RTP Control Protocol Extended Reports (RTCP XR) Parameters	
Enable RTCP XR 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 the parameter to take effect, a device reset is required.
Minimum Gap Size [VQMonGMin]	Defines the voice quality monitoring - minimum gap size (number of frames). The default is 16.
Burst Threshold [VQMonBurstHR]	Defines the voice quality monitoring - excessive burst alert threshold. The default is -1 (i.e., no alerts are issued).
Delay Threshold [VQMonDelayTHR]	Defines the voice quality monitoring - excessive delay alert threshold. The default is -1 (i.e., no alerts are issued).
R-Value 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).
RTCP XR Packet Interval 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.
Disable RTCP XR Interval Randomization 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
SBC RTCP XR Report Mode 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

Parameter	Description
publication-ip-group-id [PublicationIPGroupID]	Defines the IP Group to where the RTCP XR is sent.

51.11 SBC Parameters

The SBC and CRP parameters are described in the table below.

Table 51-38: SBC and CRP Parameters

Parameter	Description
CRP-specific Parameters	
CRP Application enable-crp [EnableCRPApplication]	Enables the CRP application. [0] Disable (default) [1] Enable Note: For the parameter to take effect, a device reset is required.
CRP Survivability Mode crp-survivability-mode [CRPSurvivabilityMode]	Defines the CRP mode. [0] Standard Mode (default) [1] Always Emergency Mode [2] Auto-answer REGISTER
crp-gw-fallback [CRPGatewayFallback]	Enables fallback routing from the proxy server to the Gateway (PSTN). [0] = Disable (default) [1] = Enable
SBC-specific Parameters	
Enable SBC enable-sbc [EnableSBCApplication]	Enables the Session Border Control (SBC) application. [0] Disable (default) [1] Enable Notes: - For the parameter to take effect, a device reset is required. - In addition to enabling the parameter, the number of maximum SBC/IP-to-IP sessions must be included in the Software License Key.
SBC and CRP Parameters	
Unclassified Calls 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.
SBC No Answer Timeout sbc-no-answer-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.

Parameter	Description
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 the 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.
configure voip/sbc general- setting/sbc-dialog- subsc-route-mode [SBCInDialogSubscribeRouteMode]	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 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 ('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).
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. The 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 the 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.
SBC Session-Expires 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.
Minimum Session-Expires 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.

Parameter	Description
	The valid range is 0 (default) to 1,000,000, where 0 means that the device does not limit Session-Expires.
configure voip/sbc general- setting/sbc- session-refresh- policy [SBCSessionRefreshingP olicy]	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, the 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.
User Registration Grace Time configure voip/sbc general- setting/sbc-usr- reg-grace-time [SBCUserRegistrationGra ceTime]	Defines additional time (in seconds) to add to the registration expiry time users that are registered in the device's Users Registration database. The valid value is 0 to 2,000,000. The default is 0. For more information, see Registration Refreshes on page 432.
SBC DB Routing Search Mode configure voip > sbc general-setting > set sbc- db-route-mode [SBCDBRoutingSearchMo de]	Defines the method for searching a registered user in the device's User Registration database when a SIP INVITE message is received for routing to a user. If the registered user is found (i.e., destination URI in INVITE), the device routes the call to the user's corresponding contact address specified in the database. ▪ [0] All permutations = (Default) Device searches for the user in the database using the entire Request-URI (user@host). If not found, it searches for the user part of the Request-URI. For example, it first searches for "4709@joe.company.com" and if not found, it searches for "4709". ▪ [1] Dest URI dependant = Device searches for the user in the database using the entire Request-URI (user@host) only. For

Parameter	Description
	example, it searches for "4709@joe.company.com". Note: If the Request-URI contains the "tel:" URI or "user=phone" parameter, the device searches only for the user part.
Handle P-Asserted-Identity 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
Keep original user in Register [SBCKeepContactUserinRegister]	Defines the device's handling of the SIP Contact header in REGISTER requests which it forwards as the outgoing message. ▪ [0] Do not keep user; override with unique identifier = (Default) The device replaces the user part of the Contact header with a unique value, for example: ✓ Incoming Contact Header: <sip:123@domain.com> ✓ Outgoing Contact Header: <sip:FEU1-7-1-3@SBC> ▪ [1] keep user without unique identifier = The device retains the original user part value of the Contact header in the outgoing REGISTER request. ▪ [2] Keep user; add unique identifier as URI parameter = The device retains the original user part value of the Contact header in the outgoing REGISTER request. In addition, it adds the special URI parameter "ac-feu=<identifier>" to the Contact header, which is used to differentiate between two SIP entities with the same user part. The identifier value is generated by the device. ✓ Incoming Contact Header: <sip:123@domain.com> ✓ Outgoing Contact Header: <sip:123@SBC;ac-feu=1-7-1-3> Note: ▪ The parameter is applicable only to REGISTER messages received from User-type IP Groups which are sent to Server-type IP Groups. ▪ Depending on the 'Remote Representation Mode' parameter of the IP Profiles table (IpProfile_SBCRemoteRepresentationMode), the host part in the SIP Contact header can be replaced by the device's IP address or by the value of the 'SIP Group Name' parameter (configured in the IP Groups table).
SBC Remote Refer Behavior 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
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

Parameter	Description
	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.
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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. Note: If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
[SBCEnforceMediaOrder]	Enables the device to include all previously negotiated media lines within the current session ('m=' line) in the SDP offer-answer exchange (RFC 3264). ▪ [0] Disable (default) ▪ [1] Enable 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 the 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 udptl t38</pre> If the parameter is disabled, the only 'm=' line included in the SDP is the newly negotiated media (i.e., image).
SBC Diversion URI Type 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.
SBC Server Auth Mode 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.

Parameter	Description
Lifetime of the nonce in seconds 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. The 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.
Authentication Challenge Method 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.
Authentication Quality of Protection auth-qop [AuthQOP]	Defines the authentication and integrity level of quality of protection (QoP) for digest authentication offered to the client. When the device challenges a SIP request (e.g., INVITE), it sends a SIP 401 response with the Proxy-Authenticate header or WWW-Authenticate header containing the 'qop' parameter. The QoP offered in the 401 response can be 'auth', 'auth-int', both 'auth' and 'auth-int', or the 'qop' parameter can be omitted from the 401 response. In response to the 401, the client needs to send the device another INVITE with the MD5 hash of the INVITE message and indicate the selected auth type. [0] 0 = The device sends 'qop=auth' in the SIP response, requesting authentication (i.e., validates user by checking user name and password). This option does not authenticate the message body (i.e., SDP). [1] 1 = The device sends 'qop=auth-int' in the SIP response, indicating required authentication and authentication with integrity (e.g., checksum). This option restricts the client to authenticating the entire SIP message, including the body, if present. [2] 2 = (Default) The device sends 'qop=auth, auth-int' in the SIP response, indicating either authentication or integrity. This enables the client to choose 'auth' or 'auth-int'. If the client chooses 'auth-int', then the body is included in the authentication. If the client chooses 'auth', then the body is not authenticated. [3] 3 = No 'qop' parameter is offered in the SIP 401 challenge message.
SBC User Registration Time 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
SBC Proxy Registration Time 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). This value is sent in the Expires header. 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.

Parameter	Description
<code>config-voip>sb general-setting sbcrand-expire</code> [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 the parameter, and then subtracts this random number from the original expiry time value. For example, assume that the original expiry time is 120 and the 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 the 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.
SBC Survivability Registration Time <code>sbcsurv-rgrstr-time</code> [SBCSurvivabilityRegistrat ionTime]	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.
[SBCEnableSurvivabilityN otice]	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

Parameter	Description
	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>
SBC Dialog-Info Interworking 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 460.
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 the parameter's settings.
SBC GRUU Mode 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.) The 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

Parameter	Description
	and its clients (endpoints). Public-GRUU: sip:userA@domain.com;gr=unique-id
Bye Authentication sbc-bye-auth [SBCEnableByeAuthentic ation]	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.
SBC Enable Subscribe Trying configure voip > sbc general-setting > set sbc-subs-try [SBCSendTryingToSubscr ibe]	Enables the device to send SIP 100 Trying responses upon receipt of SUBSCRIBE or NOTIFY messages. ▪ [0] Disable (Default) ▪ [1] Enable
[SBCExtensionsProvisioni ngMode]	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 534.
SBC Direct Media sbc-direct-media [SBCDirectMedia]	Enables the Direct Media feature (i.e., no Media Anchoring) 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). ▪ [1] Enable = Direct media flow between endpoints for all SBC calls. Notes: ▪ The setting of direct media in the SIP Interface table overrides this global parameter. In other words, even if the parameter is disabled for direct media (i.e., Media Anchoring is enabled), if direct media is enabled for a SIP Interface (in the SIP Interface table), calls between endpoints belonging to the SIP Interface employ direct media. ▪ For more information on No Media Anchoring, see "Direct Media" on page 436.
Transcoding Mode transcoding-mode	Global parameter that defines the voice transcoding mode (media negotiation). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_TranscodingMode). For a detailed description

Parameter	Description
[TranscodingMode]	of the parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 391. 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.
Preferences Mode [SBCPreferencesMode]	Determines the order of the Extension coders (coders added if there are no common coders between SDP offered coders and Allowed coders) and Allowed coders (configured in the Allowed Coders Group table) in the outgoing SIP message (in the SDP). ▪ [0] Doesn't Include Extensions = (Default) Extension coders are added at the end of the coder list. ▪ [1] Include Extensions = Extension coders and Allowed coders are arranged according to their order of appearance in the Allowed Coders Group table. Note: The parameter is applicable only if an Extension Coders Group is assigned to the IP Profile (IP Profile table's parameter, SBCExtensionCodersGroupID).
SBC RTCP Mode 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 the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
SBC Send Invite To All Contacts 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 531.
SBC Shared Line Registration Mode sbc-shared-line-reg-mode [SBCSharedLineRegMode]	Enables the termination on the device of SIP REGISTER messages from secondary lines that belong 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.
SBC Forking Handling Mode sbc-forking-handling-mode [SBCForkingHandlingMode]	Defines the handling of SIP 18x responses that are 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.

Parameter	Description
	[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.
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.
configure voip > sbc settings > sbc- remove-sips-non- sec-transp [SBCRemoveSIPsFromNonSecuredTransport]	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.
SBC Fax Detection Timeout configure voip/sbc general- setting/sbc-fax- detection-timeout [SBCFaxDetectionTimeout]	Defines the duration (in seconds) for which the device attempts to detect fax (CNG tone) immediately upon the establishment of a voice session. The interval starts from the establishment of the voice call. The valid value is 1 to any integer. The default is 10. The feature applies to faxes that are sent immediately after the voice channel is established (i.e., after 200 OK). You can configure the handling of fax negotiation by the device for specific calls, using IP Profiles configured in the IP Profile table (see the IpProfile_SBCRemoteRenegotiateOnFaxDetection parameter in Configuring IP Profiles on page 391).
Admission Control Table	
Admission Control configure voip > sbc sbc-admission- control [SBCAdmissionControl]	Defines Call Admission Control (CAC) rules. The format of the ini file table parameter is as follows: <pre>[SBCAdmissionControl] FORMAT SBCAdmissionControl_Index = SBCAdmissionControl_AdmissionControlName, SBCAdmissionControl_LimitType, SBCAdmissionControl_IPGroupName, SBCAdmissionControl_SRDName, SBCAdmissionControl_SIPInterfaceName, SBCAdmissionControl_RequestType,</pre>

Parameter	Description
	SBCAdmissionControl_RequestDirection, SBCAdmissionControl_Limit, SBCAdmissionControl_LimitPerUser, SBCAdmissionControl_Rate, SBCAdmissionControl_MaxBurst, SBCAdmissionControl_Reservation; [\SBCAdmissionControl] For a description of the table, see "Configuring Admission Control" on page 463.
Allowed Audio Coders Table	
Allowed Audio Coders <pre>configure voip > sbcs allowed-coders-group allowedcodersgroup0 [AllowedCodersGroupX]</pre>	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> Where X represents the index number. For a description of the table, see "Configuring Allowed Audio Coder Groups" on page 467.
Allowed Video Coders Table	
<pre>configure voip/sbc allowed-video-coders-group group- X [AllowedVideoCodersGroupX]</pre>	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: <pre>[AllowedVideoCodersGroup0] FORMAT AllowedVideoCodersGroup_Index = AllowedVideoCodersGroup_Name; [\AllowedVideoCodersGroup]</pre> Where X represents the index number. For a description of the table, see "Configuring Allowed Video Coder Groups" on page 469.
Classification Table	
Classification Table <pre>configure voip > sbcs routing classification [Classification]</pre>	Defines call Classification rules. The format of the ini file table parameter is as follows: <pre>[Classification] FORMAT Classification_Index = Classification_ClassificationName, Classification_MessageConditionName, Classification_SRDName, Classification_SrcSIPInterfaceName, Classification_SrcAddress, Classification_SrcPort, Classification_SrcTransportType, Classification_SrcUsernamePrefix, Classification_SrcHost, Classification_DestUsernamePrefix, Classification_DestHost, Classification_ActionType, Classification_SrcIPGroupName, Classification_DestRoutingPolicy, Classification_IpProfileName; [\Classification]</pre> For a description of the table, see "Configuring Classification Rules" on page 471.
Condition Table	
Condition Table <pre>configure voip > sbcs routing condition-table</pre>	Defines SIP Message Condition rules. <pre>[ConditionTable] FORMAT ConditionTable_Index = ConditionTable_Condition, ConditionTable_Description;</pre>

Parameter	Description
[ConditionTable]	[\ConditionTable] For a description of the table, see "Configuring Message Condition Rules" on page 478.
SBC IP-to-IP Routing Table	
IP-to-IP Routing Table configure voip > sbc routing ip2ip- routing [IP2IPRouting]	Defines SBC IP-to-IP routing rules. The format of the ini file table parameter is as follows: [IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_RouteName, IP2IPRouting_RoutingPolicyName, IP2IPRouting_SrcIPGroupName, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_RequestType, IP2IPRouting_MessageConditionName, IP2IPRouting_ReRouteIPGroupName, IP2IPRouting_Trigger, IP2IPRouting_CallSetupRulesSetId, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupName, IP2IPRouting_DestSIPInterfaceName, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions, IP2IPRouting_GroupPolicy, IP2IPRouting_CostGroup, IP2IPRouting_DestTags, IP2IPRouting_SrcTags; [\IP2IPRouting] For a description of the table, see "Configuring SBC IP-to-IP Routing Rules" on page 480.
SBC Alternative Routing Reasons Table	
SBC Alternative Routing Reasons configure voip > sbc routing sbc- alternative- routing-reasons [SBCAlternativeRoutingReasons]	Defines SBC alternative routing reason rules. The format of the ini file table parameter is as follows: [SBCAlternativeRoutingReasons] FORMAT SBCAlternativeRoutingReasons_Index = SBCAlternativeRoutingReasons_ReleaseCause; [\SBCAlternativeRoutingReasons] For a description of the table, see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 491.
IP to IP Inbound Manipulation Table	
IP to IP Inbound Manipulation configure voip > sbc manipulations ip-inbound- manipulation [IPInboundManipulation]	Defines IP-to-IP inbound manipulation rules. The format of the ini file table parameter is as follows: [IPInboundManipulation] FORMAT IPInboundManipulation_Index = IPInboundManipulation_ManipulationName IPInboundManipulation_IsAdditionalManipulation, IPInboundManipulation_ManipulatedURI, IPInboundManipulation_ManipulationPurpose, IPInboundManipulation_SrcIPGroupName, IPInboundManipulation_SrcUsernamePrefix, IPInboundManipulation_SrcHost, IPInboundManipulation_DestUsernamePrefix, IPInboundManipulation_DestHost, IPInboundManipulation_RequestType, IPInboundManipulation_RemoveFromLeft, IPInboundManipulation_RemoveFromRight, IPInboundManipulation_LeaveFromRight,

Parameter	Description
	IPInboundManipulation_Prefix2Add, IPInboundManipulation_Suffix2Add; [IPInboundManipulation] For a description of the table, see "Configuring IP-to-IP Inbound Manipulations" on page 499.
IP to IP Outbound Manipulation Table	
IP to IP Outbound Manipulation configure voip > sbc manipulations ip-outbound-manipulation [IPOutboundManipulation]	Defines IP-to-IP outbound manipulation rules. The format of the ini file table parameter is as follows: [IPOutboundManipulation] FORMAT IPOutboundManipulation_Index = IPOutboundManipulation_ManipulationName, IPOutboundManipulation_RoutingPolicyName, IPOutboundManipulation_IsAdditionalManipulation, IPOutboundManipulation_SrcIPGroupName, IPOutboundManipulation_DestIPGroupName, IPOutboundManipulation_SrcUsernamePrefix, IPOutboundManipulation_SrcHost, IPOutboundManipulation_DestUsernamePrefix, IPOutboundManipulation_DestHost, IPOutboundManipulation_CallingNamePrefix, IPOutboundManipulation_MessageConditionName, IPOutboundManipulation_RequestType, IPOutboundManipulation_ReRouteIPGroupName, IPOutboundManipulation_Trigger, IPOutboundManipulation_ManipulatedURI, IPOutboundManipulation_RemoveFromLeft, IPOutboundManipulation_RemoveFromRight, IPOutboundManipulation_LeaveFromRight, IPOutboundManipulation_Prefix2Add, IPOutboundManipulation_Suffix2Add, IPOutboundManipulation_PrivacyRestrictionMode, IPOutboundManipulation_DestTags, IPOutboundManipulation_SrcTags; [IPOutboundManipulation] For a description of the table, see "Configuring IP-to-IP Outbound Manipulations" on page 504.
SBC Routing Policy Table	
SBC Routing Policy configure voip > sbc routing sbc-routing-policy [SBCRoutingPolicy]	Defines SBC Routing Policies. The format of the ini file table parameter is as follows: [SBCRoutingPolicy] FORMAT SBCRoutingPolicy_Index = SBCRoutingPolicy_Name, SBCRoutingPolicy_LCREnable, SBCRoutingPolicy_LCRAverageCallLength, SBCRoutingPolicy_LCRDefaultCost, SBCRoutingPolicy_LdapServerGroupName; [SBCRoutingPolicy] For a description of the table, see "Configuring SBC Routing Policy Rules" on page 493.
Dial Plan Table	
Dial Plan configure voip >	Defines the name of the Dial Plan. The format of the ini file table parameter is as follows:

Parameter	Description
sbc dial-plan [DialPlans]	[DialPlan] FORMAT DialPlan_Index = DialPlan_Name; [\DialPlan] For a description of the table, see Configuring Dial Plans on page 511.
Dial Plan Rule Table	
Dial Plan Rule configure voip > sbc dial-plan-rule [DialPlanRule]	Defines the dial plan rules per Dial Plan. For a description of the table, see Configuring Dial Plans. Note: - The table is hidden in the ini file. - To configure Dial Plan rules from a file, see Importing and Exporting Dial Plans on page 515.

51.11.1 Supplementary Services

The SBC and CRP supplementary services parameters are described in the table below.

Table 51-39: SBC and CRP Supplementary Services Parameters

Parameter	Description
Emergency Call Preemption Parameters For more information on SBC emergency call preemption, "Configuring Call Preemption for SBC Emergency Calls" on page 521.	
SBC Preemption Mode configure voip > sbc general-setting > sbc-preemption- mode [SBCPreemptionMode]	Enables SBC emergency call preemption. [0] Disable (default) [1] Enable
SBC Emergency Message Condition configure voip > sbc general-setting > sbc-emerg- condition [SBCEmergencyCondition]	Defines the index of the Message Condition rule in the Message Condition table that is used to identify emergency calls. Note: The device applies the rule only after call classification (but before inbound manipulation).
SBC Emergency RTP DiffServ configure voip > sbc general-setting > sbc-emerg-rtp- diffserv [SBCEmergencyRTPDiffserv]	Defines DiffServ bits sent in the RTP for SBC emergency calls. The valid value is 0 to 63. The default is 46.

Parameter	Description
SBC Emergency Signaling DiffServ configure voip > sbc general-setting > sbc-emerg-sig-diffserv [SBCEmergencySignaling DiffServ]	Defines DiffServ bits sent in SIP signaling messages for SBC emergency calls. This is included in the SIP Resource-Priority header. The valid value is 0 to 63. The default is 40.

51.12 IP Media Parameters

The IP media parameters are described in the table below.

Table 51-40: IP Media Parameters

Parameter	Description
IPMedia Detectors IPM-detectors-enable [EnabledDSPIPMDetectors]	Enables the device's DSP detectors for detection features such as AMD. ▪ [0] Disable (default) ▪ [1] Enable Notes: ▪ For the parameter to take effect, a device reset is required. ▪ The DSP Detectors 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 596.
Number of Media Channels media-channels [MediaChannels]	Defines the maximum number of DSP channels allocated for various functionalities such as transcoding, . The default is 0. Notes: ▪ For the parameter to take effect, a device reset is required. ▪ The SBC application does not require DSP channels. The SBC application uses DSP channels only if media transcoding is needed, where two DSP channels are used per transcoding session.
Automatic Gain Control (AGC) Parameters	
Enable AGC AGC-enable [EnableAGC]	Enables the AGC mechanism. The AGC mechanism adjusts the level of the received signal to maintain a steady (configurable) volume level. ▪ [0] Disable (default) ▪ [1] Enable Note: For a description of AGC, see Automatic Gain Control (AGC) on page 210.
AGC Slope AGC-gain-slope [AGCGainSlope]	Determines the AGC convergence rate: ▪ [0] 0 = 0.25 dB/sec ▪ [1] 1 = 0.50 dB/sec ▪ [2] 2 = 0.75 dB/sec ▪ [3] 3 = 1.00 dB/sec (default) ▪ [4] 4 = 1.25 dB/sec ▪ [5] 5 = 1.50 dB/sec ▪ [6] 6 = 1.75 dB/sec ▪ [7] 7 = 2.00 dB/sec ▪ [8] 8 = 2.50 dB/sec ▪ [9] 9 = 3.00 dB/sec ▪ [10] 10 = 3.50 dB/sec ▪ [11] 11 = 4.00 dB/sec ▪ [12] 12 = 4.50 dB/sec ▪ [13] 13 = 5.00 dB/sec ▪ [14] 14 = 5.50 dB/sec ▪ [15] 15 = 6.00 dB/sec

Parameter	Description
	[16] 16 = 7.00 dB/sec [17] 17 = 8.00 dB/sec [18] 18 = 9.00 dB/sec [19] 19 = 10.00 dB/sec [20] 20 = 11.00 dB/sec [21] 21 = 12.00 dB/sec [22] 22 = 13.00 dB/sec [23] 23 = 14.00 dB/sec [24] 24 = 15.00 dB/sec [25] 25 = 20.00 dB/sec [26] 26 = 25.00 dB/sec [27] 27 = 30.00 dB/sec [28] 28 = 35.00 dB/sec [29] 29 = 40.00 dB/sec [30] 30 = 50.00 dB/sec [31] 31 = 70.00 dB/sec
AGC Redirection AGC-redirection [AGCRedirection]	Determines the AGC direction. [0] 0 = (Default) AGC works on signals from the TDM side. [1] 1 = AGC works on signals from the IP side.
AGC Target Energy AGC-target-energy [AGCTargetEnergy]	Defines the signal energy value (dBm) that the AGC attempts to attain. The valid range is 0 to -63 dBm. The default is -19 dBm.
AGC Minimum Gain AGC-min-gain [AGCMinGain]	Defines the minimum gain (in dB) by the AGC when activated. The range is 0 to -31. The default is -20. Note: For the parameter to take effect, a device reset is required.
AGC Maximum Gain AGC-max-gain [AGCMaxGain]	Defines the maximum gain (in dB) by the AGC when activated. The range is 0 to 18. The default is 15. Note: For the parameter to take effect, a device reset is required.
Disable Fast Adaptation AGC-disable-fast-adaptation [AGCDisableFastAdaptation]	Enables the AGC Fast Adaptation mode. [0] = Disable (default) [1] = Enable Note: For the parameter to take effect, a device reset is required.
Answering Machine Detector (AMD) Parameters For more information on AMD, see "Answering Machine Detection (AMD)" on page 206.	
Answer Machine Detector Sensitivity Parameter Suit amd-sensitivity-parameter-suit [AMDSensitivityParameterSuit]	Global parameter that defines the AMD Parameter Suite to use. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDSensitivityParameterSuit). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
Answer Machine Detector Sensitivity Level	Global parameter that defines the AMD detection sensitivity level of the selected AMD Parameter Suite. You can also configure this

Parameter	Description
amd-sensitivity-level [AMDSensitivityLevel]	functionality per specific calls, using IP Profiles (IpProfile_AMDSensitivityLevel). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
AMD Sensitivity File [AMDSensitivityFileName]	Defines the name of the AMD Sensitivity file that contains the AMD Parameter Suites. Notes: - This file must be in binary format (.dat). You can use the DConvert utility to convert the original file format from XML to .dat. - You can load this file using the Web interface (see "Loading Auxiliary Files" on page 583).
[AMDSensitivityFileUrl]	Defines the URL path to the AMD Sensitivity file for downloading from a remote server.
[AMDMinimumVoiceLength]	Defines the AMD minimum voice activity detection duration (in 5-ms units). Voice activity duration below this threshold is ignored and considered as non-voice. The valid value range is 10 to 100. The default is 42 (i.e., 210 ms).
[AMDMaxGreetingTime]	Global parameter that defines the maximum duration that the device can take to detect a greeting message. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDMaxGreetingTime). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
[AMDMaxPostGreetingSilenceTime]	Global parameter that defines the maximum duration of silence from after the greeting time is over (defined by AMDMaxGreetingTime) until the device's AMD decision. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDMaxPostSilenceGreetingTime). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.
[AMDTimeout]	Defines the timeout (in msec) between receiving Connect messages from the ISDN and sending AMD results. The valid range is 1 to 30,000. The default is 2,000 (i.e., 2 seconds).
AMD Beep Detection Mode amd-beep-detection [AMDBeepDetectionMode]	Determines the AMD beep detection mode. This mode detects the beeps played at the end of an answering machine message, by using the X-Detect header extension. The device sends a SIP INFO message containing the field values Type=AMD and SubType=Beep. This feature allows users of certain third-party, Application server to leave a voice message after an answering machine plays the "beep". [0] Disabled (default) [1] Start After AMD

Parameter	Description
	▪ [2] Start Immediately
Answer Machine Detector Beep Detection Timeout amd-beep-detection-timeout [AMDBeepDetectionTimeout]	Defines the AMD beep detection timeout (i.e., the duration that the beep detector functions from when detection is initiated). This is used for detecting beeps at the end of an answering machine message. The valid value is in units of 100 milliseconds, from 0 to 1638. The default is 200 (i.e., 20 seconds).
Answer Machine Detector Beep Detection Sensitivity amd-beep-detection-sensitivity [AMDBeepDetectionSensitivity]	Defines the AMD beep detection sensitivity for detecting beeps at the end of an answering machine message. The valid value is 0 to 3, where 0 (default) is the least sensitive.
early-amd [EnableEarlyAMD]	Enables AMD detection to be activated upon receipt of an ISDN Alerting or Connect message. ▪ [0] = (Default) Disable - AMD is activated upon receipt of ISDN Connect message. ▪ [1] = Enable - AMD is activated upon receipt of ISDN Alerting message.
AMD mode amd-mode [AMDmode]	Global parameter that enables the device to disconnect the IP-to-Tel call upon detection of an answering machine on the Tel side. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AmdMode). For a detailed description of the parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 391. 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.

51.13 Services

51.13.1 SIP-based Media Recording Parameters

The SIP-based media recording parameters are described in the table below.

Table 51-41: SIP-based Media Recording Parameters

Parameter	Description
SIP Recording Application 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 the parameter to take effect, a device reset is required.
Recording Server (SRS) Destination Username configure voip/services sip- recording general- setting/siprec- server-dest-username [SIPRecServerDestUserna me]	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 Time Stamp Format configure voip/services sip- recording general- setting/siprec-time- stamp [SIPRecTimeStamp]	Defines the format of the device's time (timestamp) in SIP messages that are sent to the SIP Recording Server (SRS). ▪ [0] Local Time = (Default) The device's local time (without the UTC time zone) is used for the timestamp. ▪ [1] UTC = The device's UTC time is used for the timestamp. Note: The timestamp is contained in the XML body of the SIP message. If the timestamp uses the UTC time, the time is suffixed with the letter "Z", for example: <associate-time>2017-09-07T06:33:38Z</associate-time>
SIP Recording Table	
SIP Recording table configure voip/services sip- recording sip-rec- routing [SIPRecRouting]	Defines SIP Recording Routing rules (for siprec). The format of the ini file table parameter is as follows: [SIPRecRouting] FORMAT SIPRecRouting_Index = SIPRecRouting_RecordedIPGroupName, SIPRecRouting_RecordedSourcePrefix, SIPRecRouting_RecordedDestinationPrefix, SIPRecRouting_PeerIPGroupName, SIPRecRouting_PeerTrunkGroupID, SIPRecRouting_Caller, SIPRecRouting_SRSIPGroupName; [\SIPRecRouting] For a description of the table, see "Configuring SIP Recording Rules" on page 230.

51.13.2 RADIUS and LDAP Parameters

51.13.2.1 General Parameters

The general RADIUS and LDAP parameters are described in the table below.

Table 51-42: General RADIUS and LDAP Parameters

Parameter	Description
Use Local Users Database configure system > mgmt-auth > use- local-users-db [MgmtUseLocalUsersData base]	Defines when the device uses its local management-users database (Web Users table) or an LDAP/RADIUS server for authenticating the login credentials (username-password) of users when logging into the device's management interface (e.g., Web or CLI). ▪ [0] When No Auth Server Defined = (Default) The device authenticates the users using the Web Users table in the following scenarios: ✓ If no LDAP/RADIUS server is configured. ✓ If an LDAP/RADIUS server is configured, but connectivity with the server is down. If there is connectivity with the server, the device uses the server to authenticate the user. ▪ [1] Always = The device first attempts to authenticate the user using the Web Users table. If no user is found (based on the username-password combination), it attempts to authenticate the user using the LDAP/RADIUS server.
Behavior upon Authentication Server Timeout 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- and RADIUS-based management-user login authentication.
Default Access Level 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.

51.13.2.2 RADIUS Parameters

The RADIUS parameters are described in the table below.

Table 51-43: RADIUS Parameters

Parameter	Description
General RADIUS Parameters	
Enable RADIUS Access Control enable [EnableRADIUS]	Enables the RADIUS application. [0] Disable (Default) [1] Enable Note: For the parameter to take effect, a device reset is required.
[RadiusTrafficType]	Defines the device's network interface for communicating (RADIUS traffic) with the RADIUS server(s). [0] OAMP (default) [1] Control Note: If set to Control, only one Control interface must be configured in the Interface table; otherwise, RADIUS communication will fail.
RADIUS VSA Vendor ID configure system > radius > 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.
[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.
RADIUS Packets Retransmission [RADIUSRetransmission]	Defines the number of RADIUS retransmission retries when no response is received from the RADIUS server. See also the RadiusTo parameter. The valid range is 1 to 10. The default is 1.
RADIUS Response Time Out [RadiusTO]	Defines the time interval (in seconds) that the device waits for a response before it performs a RADIUS retransmission. See also the RADIUSRetransmission parameter. The valid range is 1 to 30. The default is 2.
RADIUS Accounting Parameters	
RADIUS Accounting Type 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.
AAA Indications aaa-indications [AAAIndications]	Determines the Authentication, Authorization and Accounting (AAA) indications. [0] None = (Default) No indications. [3] Accounting Only = Only accounting indications are used.
RADIUS User Authentication Parameters	
Use RADIUS for Web/Telnet Login enable-mgmt-login	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-

Parameter	Description
[WebRADIUSLogin]	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 must also configure the EnableRADIUS parameter to 1 (Enable). ▪ 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 HTTPOnly parameter to 1 to force the use of HTTPS, since the transport is encrypted.
Password Local Cache Mode 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).
Password Local Cache Timeout 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.
RADIUS VSA Access Level Attribute 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.

51.13.2.3 LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below.

Table 51-44: LDAP Parameters

Parameter	Description
LDAP Service configure voip/ldap/enable [LDAPServiceEnable]	Enables the LDAP feature. ▪ [0] Disable (default) ▪ [1] Enable Note: For the parameter to take effect, a device reset is required.
LDAP Authentication Filter 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 the parameter is set to "(sAMAccountName=\$)" and the user logs in with the username "SueM", the LDAP query is run for sAMAccountName=SueM.
Use LDAP for Web/Telnet Login configure voip > ldap > enable-mgmt-login [MgmtLDAPLogin]	Enables LDAP-based management-user login authentication and authorization. ▪ [0] Disable (default) ▪ [1] Enable Note: For the 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.
LDAP Numeric Attribute configure voip > sip-definition advanced-settings > ldap-numeric-attr [LDAPNumericAttributes]	Defines up to five LDAP Attributes (separated by commas) for which the device employs LDAP query searches in the AD for numbers that may have characters between the digits. For more information, see Enabling LDAP Searches for Numbers with Characters on page 263.
MS LDAP OCS Number attribute name ldap-ocs-nm-attr [MSLDAPOCSNumAttributeName]	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-Line".
MS LDAP PBX Number attribute name ldap-pbx-nm-attr [MSLDAPPBXNumAttributeName]	Defines the name of the attribute that represents the user PBX number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "telephoneNumber".
MS LDAP MOBILE Number attribute name ldap-mobile-nm-attr [MSLDAPMobileNumAttributeName]	Defines the name of the attribute that represents the user Mobile number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "mobile".

Parameter	Description
ldap-private-nm-attr [MSLDAPPrivateNumAttribute Name]	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, the parameter is not used as a search key. The default is "msRTCSIP-PrivateLine".
MS LDAP DISPLAY Name Attribute Name ldap-display-nm-attr [MSLDAPDisplayNameAttribute Name]	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".
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.
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 cache [LDAPCacheEnable]	Enables the LDAP cache service. ▪ [0] Disable (default) ▪ [1] Enable Notes: ▪ For the parameter to take effect, a device reset is required. ▪ For more information on LDAP caching, see "Configuring the Device's LDAP Cache" on page 253.
LDAP Configuration Table	
LDAP Configuration Table configure voip > ldap > ldap-configuration [LdapConfiguration]	Defines LDAP servers. The format of the ini file table parameter is as follows: [LdapConfiguration] FORMAT LdapConfiguration_Index = LdapConfiguration_Group, LdapConfiguration_LdapConfServerIp, LdapConfiguration_LdapConfServerPort, LdapConfiguration_LdapConfServerMaxRespondTime, LdapConfiguration_LdapConfServerDomainName, LdapConfiguration_LdapConfPassword, LdapConfiguration_LdapConfBindDn, LdapConfiguration_Interface, LdapConfiguration_MngmAuthAtt, LdapConfiguration_useTLS, LdapConfiguration_ConnectionStatus; [\LdapConfiguration] For a description of the table, see "Configuring LDAP Servers" on page 245.
LDAP Server Search Base DN Table	
LDAP Server Search Base DN Table configure voip > ldap > ldap-servers-	Defines the full base path (i.e., distinguished name / DN) to the objects in the AD where the query is done, per LDAP server. The format of the ini file table parameter is as follows: [LdapServersSearchDNs]

Parameter	Description
search-dns [LdapServersSearchDNs]	FORMAT LdapServersSearchDNs_Index = LdapServersSearchDNs_Base_Path, LdapServersSearchDNs_LdapConfigurationIndex, LdapServersSearchDNs_SearchDnInternalIndex; [\LdapServersSearchDNs] For a detailed description of the table, see "Configuring LDAP DNS (Base Paths) per LDAP Server" on page 250.
Management LDAP Groups Table	
Management LDAP Groups Table configure voip > ldap > mgmt-ldap-groups [MgmntLDAPGroups]	Defines the users group attribute in the AD and corresponding management access level. The format of the ini file table parameter is as follows: [MgmntLDAPGroups] FORMAT MgmntLDAPGroups_Index = MgmntLDAPGroups_LdapConfigurationIndex, MgmntLDAPGroups_GroupIndex, MgmntLDAPGroups_Level, MgmntLDAPGroups_Group; [\MgmntLDAPGroups] For a description of the table, see "Configuring Access Level per Management Groups Attributes" on page 251.
LDAP Server Groups Table	
LDAP Server Groups Table config-voip > ldap > ldap-servers-group [LDAPServersGroup]	Defines LDAP Server Groups. The format of the ini file table parameter is as follows: [LdapServersGroup] FORMAT LdapServersGroup_Index = LdapServersGroup_Name, LdapServersGroup_ServerType, LdapServersGroup_SearchMethod, LdapServersGroup_CacheEntryTimeout, LdapServersGroup_CacheEntryRemovalTimeout, LdapServersGroup_SearchDnsMethod; [\LdapServersGroup] For a description of the table, see "Configuring LDAP Server Groups" on page 243.

51.13.3 Least Cost Routing Parameters

The Least Cost Routing (LCR) parameters are described in the table below.

Table 51-45: LCR Parameters

Parameter	Description
Cost Group Table configure voip > services least-cost-routing cost-group [CostGroupTable]	Defines 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;

Parameter	Description
	For a description of the table, see "Configuring Cost Groups" on page 271.
Cost Group > Time Band Table configure voip > services least-cost-routing cost-group-time-bands [CostGroupTimebands]	Defines time bands and associates them with Cost Groups. [CostGroupTimebands] FORMAT CostGroupTimebands_TimebandIndex = CostGroupTimebands_StartTime, CostGroupTimebands_EndTime, CostGroupTimebands_ConnectionCost, CostGroupTimebands_MinuteCost; [\CostGroupTimebands] For a description of the table, see "Configuring Time Bands for Cost Groups" on page 272.

51.13.4 Call Setup Rules Parameters

The Call Setup Rules parameters are described in the table below.

Table 51-46: Call Setup Rules Parameters

Parameter	Description
Call Setup Rules configure voip/services call-setup-rules [CallSetupRules]	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_QueryTarget, CallSetupRules_AttributesToQuery, CallSetupRules_AttributesToGet, CallSetupRules_RowRole, CallSetupRules_Condition, CallSetupRules_ActionSubject, CallSetupRules_ActionType, CallSetupRules_ActionValue; [\CallSetupRules] For a description of the table, see "Configuring Call Setup Rules" on page 292.

51.13.5 HTTP-based Services

The HTTP-based service parameters are described in the table below.

Table 51-47: HTTP-based Service Parameters

Parameter	Description
HTTP Remote Services [HTTPRemoteServices]	Defines HTTP-based services. The format of the ini file table parameter is as follows: [HTTPRemoteServices] FORMAT HTTPRemoteServices_Index = HTTPRemoteServices_Name, HTTPRemoteServices_Path, HTTPRemoteServices_HTTPType, HTTPRemoteServices_Policy, HTTPRemoteServices_LoginNeeded, HTTPRemoteServices_PersistentConnection, HTTPRemoteServices_NumOfSockets, HTTPRemoteServices_AuthUserName,

Parameter	Description
	HTTPRemoteServices_AuthPassword, HTTPRemoteServices_TLSTContext, HTTPRemoteServices_VerifyCertificate, HTTPRemoteServices_TimeOut, HTTPRemoteServices_KeepAliveTimeOut, HTTPRemoteServices_ServiceStatus; [HTTPRemoteServices] For a description of the table, see "Configuring HTTP Services" on page 275.
HTTP Remote Hosts [HTTPRemoteHosts]	Defines remote HTTP hosts per HTTP-based service. The format of the ini file table parameter is as follows: [HTTPRemoteHosts] FORMAT HTTPRemoteHosts_Index = HTTPRemoteHosts_HTTPRemoteServiceIndex, HTTPRemoteHosts_RemoteHostIndex, HTTPRemoteHosts_Name, HTTPRemoteHosts_Address, HTTPRemoteHosts_Port, HTTPRemoteHosts_Interface, HTTPRemoteHosts_HTTPTransportType, HTTPRemoteHosts_HostStatus; [HTTPRemoteHosts] For a description of the table, see "Configuring Remote HTTP Hosts" on page 278.
Topology Status [RoutingServerGroupStatus]	Enables the reporting of the device's topology status (using the REST TopologyStatus API command) to HTTP remote hosts. [0] Disable (default) [1] Enable For more information, see "Configuring HTTP Services" on page 275.

51.13.6 HTTP Proxy Parameters

The HTTP Proxy service parameters are described in the table below.

Table 51-48: HTTP Proxy Service Parameters

Parameter	Description
HTTP Proxy Application configure system > http-proxy > http- proxy-app [HTTPProxyApplication]	Enables the HTTP Proxy application. [0] Disable (default) [1] Enable Note: For the parameter to take effect, a device reset is required.

Parameter	Description
HTTP Interfaces Table configure system > http-proxy > http- interface [HTTPInterface]	Defines local listening interfaces for receiving HTTP/S requests from Web clients for HTTP/S-based services. The format of the ini file table parameter is as follows: [HTTPInterface] FORMAT HTTPInterface_Index = HTTPInterface_InterfaceName, HTTPInterface_NetworkInterface, HTTPInterface_Protocol, HTTPInterface_Port, HTTPInterface_TLSContext, HTTPInterface_VerifyCert; [\HTTPInterface] For a description of the table, see 'Configuring HTTP Interfaces' on page 284.
HTTP Proxy Services Table configure system > http-proxy > http- proxy-serv [HTTPProxyService]	Defines HTTP Proxy based services. The format of the ini file table parameter is as follows: [HTTPProxyService] FORMAT HTTPProxyService_Index = HTTPProxyService_ServiceName, HTTPProxyService_ListeningInterface, HTTPProxyService_URLPrefix, HTTPProxyService_KeepAliveMode; [\HTTPProxyService] For a description of the table, see 'Configuring HTTP Proxy Services' on page 286.
HTTP Proxy Hosts Table configure system > http-proxy > http- proxy-host [HTTPProxyHost]	Defines HTTP Proxy hosts. The table is a "child" of the HTTP Proxy Services table (HTTPProxyService). An HTTP Proxy Host represents the HTTP-based managed equipment (e.g., IP Phone). The format of the ini file table parameter is as follows: [HTTPProxyHost] FORMAT HTTPProxyHost_Index = HTTPProxyHost_HTTPProxyServiceId, HTTPProxyHost_HTTPProxyHostId, HTTPProxyHost_NetworkInterface, HTTPProxyHost_IpAddress, HTTPProxyHost_Protocol, HTTPProxyHost_Port, HTTPProxyHost_TLSContext, HTTPProxyHost_VerifyCert; [\HTTPProxyHost] For a description of the table, see 'Configuring HTTP Proxy Hosts' on page 287.
EMS Services Table configure system > http-proxy > ems- serv [EMSService]	Defines an HTTP-based EMS Service so that the device can act as an HTTP Proxy that enables AudioCodes EMS to manage AudioCodes equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and EMS is located in a public domain (e.g., in the WAN). The format of the ini file table parameter is as follows: [EMSService] FORMAT EMSService_Index = EMSService_ServiceName, EMSService_PrimaryServer, EMSService_SecondaryServer, EMSService_DeviceLoginInterface, EMSService_EMSServiceInterface; [\EMSService] For a description of the table, see 'Configuring an HTTP-based EMS Service' on page 289.

This page is intentionally left blank.

52 SBC and DSP Channel Capacity

This chapter lists the supported DSP firmware templates and channel capacity.



Notes:

- Installation and use of voice coders is subject to obtaining the appropriate license and royalty payments.
- The number of channels refers to the maximum channel capacity of the device.
- For additional DSP templates, contact your AudioCodes sales representative.

52.1 Signaling-Media Sessions & User Registrations

The table below lists the maximum capacity figures for SIP signaling, media sessions, and registered users.

Table 52-1: Maximum Signaling, Media Sessions and Registered Users

Product	Signaling Sessions	Media Sessions			Registered Users
		RTP-to-RTP	SRTP-RTP	Codec Transcoding	
Mediant 2600 SBC	600	600	600	See Table 52-2	8,000



Notes:

- The figures listed in the table 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 *RTP-to-RTP* column represents maximum media sessions when all media sessions are RTP-to-RTP only. The same applies to the *SRTP-RTP* column.
- *Registered Users* is the maximum number of users that can be registered with the device. This applies to the supported application (SBC or CRP).
- 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, for products with a greater signaling session capacity 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 G.729, one signaling resource, one media session resource, and one transcoding session resource is used.

52.2 Channel Capacity and Capabilities

The maximum number of supported SBC sessions is listed in "Signaling-Media Sessions & User Registrations" on page 835. The SBC sessions also support SRTP and RTCP XR. When DSP capabilities are required, the number of sessions that can use DSP capabilities is reduced, as shown in the table below.

Table 52-2: Channel Capacity per Coder-Capability Profile

Session Coders		Max. Sessions	
From Coder Profile	To Coder	Without MPM4	With MPM4
1	Profile 1	400	600
2	Profile 1	300	600
2	Profile 2	250	600
1	Profile 2 + AMR-NB / G.722	275	600
2	Profile 2 + AMR-NB / G.722	225	600
1	Profile 2 + iLBC	175	575
2	Profile 2 + iLBC	150	500
1	Profile 2 + AMR-WB	200	600
2	Profile 2 + AMR-WB	175	525
1	Profile 2 + SILK-NB	200	600
2	Profile 2 + SILK-NB	175	525
1	Profile 2 + SILK-WB	100	350
2	Profile 2 + SILK-WB	100	350
1	Profile 2 + Opus-NB	125	425
2	Profile 2 + Opus-NB	125	375
1	Profile 2 + Opus-WB	100	300
2	Profile 2 + Opus-WB	75	275



Notes:

- Profile 1: G.711 at 20ms only, with in-band signaling (in voice channel) and Silence Suppression (no fax detection or T.38 support).
- Profile 2: G.711, G.726, G.729, G.723.1, T.38 with fax detection, in-band signaling (in voice channel), and Silence Compression.
- Acoustic Echo Suppressor reduces performance. For more information, contact your AudioCodes sales representative.
- MPM is the optional, Media Processing Module that provides additional DSPs, allowing greater capacity.

53 Technical Specifications

The device's technical specifications are listed in the table below.


Notes:

- All specifications in this document are subject to change without prior notice.
- The compliance and regulatory information can be downloaded from AudioCodes Web site at <http://www.audiocodes.com/library>.

Table 53-1: Technical Specifications

Function	Specification
Networking Interfaces	
Ethernet	8 Redundant 100/1000Base-T Ethernet ports for physical separation between multiple LAN and WAN, between Media, Control and OAMP
Security	
Encryption and Authentication	TLS, DTLS, SRTP, HTTPS, SSH, client/server SIP Digest authentication, RADIUS Digest
Access Control	DoS/DDoS line rate protection, bandwidth throttling, dynamic blacklisting
VoIP Firewall	RTP pinhole management, rogue RTP detection and prevention, SIP message policy, advanced RTP latching
Privacy	Topology hiding, user privacy
Traffic Separation	VLAN/physical interface separation for multiple media, control and OAMP interfaces
Intrusion Detection System	Detection and prevention of VoIP attacks, 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 and Authentication	User registration restriction control, registration and authentication on behalf of users, SIP authentication server for SBC users
Transport Mediation	SIP over UDP/TCP/TLS/WebSocket, IPv4 to IPv6, RTP to SRTP (SDS/DTLS)
Message Manipulation	Ability to add/modify/delete SIP headers and message body using advanced regular expressions (regex)
URI and Number Manipulations	URI user and host name manipulations, ingress and egress digit manipulation
Transcoding and Vcoders	Coder normalization including transcoding, coder enforcement and re-prioritization, extensive vocoder support: G.711, G.723.1, G.726, G.729, GSM-FR, AMR-NB, AMR-WB (G.722.2) SILK-NB/WB, Opus-NB/WB

Function	Specification
NAT	Local and far-end NAT traversal for support of remote workers
Signal Conversion	DTMF/RFC 2833/SIP, T.38 fax
WebRTC Controller	Interworking between WebRTC devices and SIP networks Supports WebSocket, Opus, VP8 video coder, lite ICE, DTLS, RTP multiplexing, secure RTCP with feedback
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.
Impairment Mitigation	Packet Loss Concealment, Dynamic Programmable Jitter Buffer, Silence Suppression/Comfort Noise Generation, RTP redundancy, broken connection detection
Voice Enhancement	Transrating, RTCP-XR, acoustic echo cancellation, replacing voice profile due to impairment detection, fixed & dynamic voice gain control
Direct Media (No Media Anchoring)	Hair-pinning of local calls to avoid unnecessary media delays and bandwidth consumption
Voice Quality Monitoring	RTCP-XR, AudioCodes Session Experience Manager (SEM)
High Availability (Redundancy)	SBC high availability with two-box redundancy, active calls preserved
Quality of Experience	Access control and media quality enhancements based on QoE and bandwidth utilization
Test Agent	Ability to remotely verify connectivity, voice quality and SIP message flow between SIP UAs
SIP Routing	
Routing Methods	Request URL, IP Address, FQDN, ENUM, advanced LDAP, third-party routing control through REST API
Advanced Routing Criteria	QoE, bandwidth, SIP message (SIP request, coder type, etc.), Layer-3 parameters
Redundancy	Detection of proxy failures and subsequent routing to alternative proxies
Routing Features	Least-cost routing, call forking, load balancing, E911 gateway support, emergency call detection and prioritization
Multiple LANs	Support for up to 48 separate LANs
SIPRec	IETF standard SIP recording interface
Management	
OAM&P	Browser-based GUI, CLI, SNMP, EMS, INI Configuration file, REST API
Multi Tenancy	Advanced multi-tenant SBC partitioning
Physical / Environmental	

Function	Specification
Dimensions (HxWxD)	1U x 19" (444 mm) x 14" (355 mm)
Weight	Approx. 11.7 lbs (5.3 kg)
Mounting	Desktop or 19" rack mount
Power	100–240 VAC redundant dual feed
Environmental	▪ Operational: 0 to 40°C (32 to 104°F) ▪ Storage: -20 to 70°C (-4 to 158°F) ▪ Relative Humidity: 10 to 85% non-condensing

International Headquarters

1 Hayarden Street,
Airport City
Lod 7019900, Israel
Tel: +972-3-976-4000
Fax: +972-3-976-4040

AudioCodes Inc.

27 World's Fair Drive,
Somerset, NJ 08873
Tel: +1-732-469-0880
Fax: +1-732-469-2298

Contact us: www.audiocodes.com/contact

Website: 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-41796

