Mediant™ 800B MSBR

Multi-Service Business Router

Session Border Controller

User's Manual

Version 6.8

September 2014

Document # LTRT-12813

Table of Contents

1 Overview 23

2 Introduction 27

3 Default OAMP IP Address...29

4 Configuring VoIP LAN Interface for OAMP 31

4.1 Web Interface..31
4.2 CLI..33

5 Configuring Data-Router's LAN and WAN 35

5.1 Configuring Data-Router's LAN Interface.......................................35
5.2 Configuring the Device's DHCP Server 36
5.3 Configuring the WAN Interface 36

6 Enabling Remote Management from WAN...............................39

6.1 Remote Web-based (HTTP/S) Management.................................39
6.2 Remote Telnet-based Management...40

7 Introduction 43

8 Web-Based Management...45

8.1 Getting Acquainted with the Web Interface....................................45
8.1.1 Computer Requirements...45
8.1.2 Accessing the Web Interface...46
8.1.3 Areas of the GUI...47
8.1.4 Toolbar Description...48
8.1.5 Navigation Tree 48
8.1.5.1 Displaying Navigation Tree in Basic and Full View..............49
8.1.5.2 Showing / Hiding the Navigation Pane.................................50
8.1.6 Working with Configuration Pages 51
8.1.6.1 Accessing Pages..51
8.1.6.2 Viewing Parameters 51
8.1.6.3 Modifying and Saving Parameters 53
8.1.6.4 Working with Tables...54
8.1.7 Searching for Configuration Parameters 55
8.1.8 Creating a Login Welcome Message.......................................57
8.1.9 Getting Help..58
8.1.10 Logging Off the Web Interface...59
8.2 Viewing the Home Page..60
8.2.1 Assigning a Port Name..63
8.3 Configuring Web User Accounts..64
8.3.1 Basic User Accounts Configuration...65
8.3.2 Advanced User Accounts Configuration...................................67
8.4 Displaying Login Information upon Login......................................70
8.5 Configuring Web Security Settings 71
8.6 Limiting OAMP Access to a Specific WAN Interface.......................72
8.7 Web Login Authentication using Smart Cards 72
8.8 Configuring Web and Telnet Access List 73

9 CLI-Based Management...75

9.1 Getting Familiar with CLI...75

9.1.1 Understanding Configuration Modes 75
9.1.2 Using CLI Shortcuts 76
9.1.3 Common CLI Commands 77
9.1.4 Configuring Tables in CLI 78
9.1.5 Understanding CLI Error Messages 79
9.2 Enabling CLI 79
9.2.1 Enabling Telnet for CLI 79
9.2.2 Enabling SSH with RSA Public Key for CLI 80
9.3 Establishing a CLI Session 82
9.4 Configuring Maximum Telnet/SSH Sessions 82
9.5 Viewing Current CLI Sessions 83
9.6 Terminating a User's CLI Session 83
9.7 Configuring Displayed Output Lines in CLI Terminal Window 84
9.8 Configuring TACACS+ for CLI Login 84

10 SNMP-Based Management 87

10.1 Enabling SNMP and Configuring SNMP Community Strings 87
10.2 Configuring SNMP Trap Destinations 89
10.3 Configuring SNMP Trusted Managers 90
10.4 Configuring SNMP V3 Users 91

11 TR-069 Based Management 93

11.1 TR-069 93
11.2 TR-104 98
11.3 Configuring TR-069 99

12 INI File-Based Management 101

12.1 INI File Format 101
12.1.1 Configuring Individual ini File Parameters 101
12.1.2 Configuring Table ini File Parameters 101
12.1.3 General ini File Formatting Rules 103
12.2 Configuring an ini File 103
12.3 Loading an ini File to the Device 104
12.4 Secured Encoded ini File 104
12.5 Configuring Password Display in ini File 105
12.6 INI Viewer and Editor Utility 106

13 Configuring SSL/TLS Certificates 109

13.1 Configuring TLS Certificate Contexts 109
13.2 Assigning CSR-based Certificates to TLS Contexts 113
13.3 Assigning Externally Created Private Keys to TLS Contexts 114
13.4 Generating Private Keys for TLS Contexts 115
13.5 Creating Self-Signed Certificates for TLS Contexts 116
13.6 Importing Certificates and Certificate Chain into Trusted Certificate Store 117
13.7 Configuring Mutual TLS Authentication 118
13.7.1 TLS for SIP Clients 118
13.7.2 TLS for Remote Device Management 119
13.8 Configuring TLS Server Certificate Expiry Check 120

14 Date and Time 121

14.1 Configuring Date and Time Manually 121

14.2 Configuring Automatic Date and Time using SNTP 121
14.3 Configuring Daylight Saving Time 123

15 Configuring Power over Ethernet 125

16 Network 129

16.1 Configuring Underlying Ethernet Devices 129
16.2 Configuring IP Network Interfaces 130
16.2.1 Assigning NTP Services to Application Types 135
16.2.2 Multiple Interface Table Configuration Summary and Guidelines 135
16.2.3 Networking Configuration Examples 136
16.2.3.1 One VoIP Interface for All Applications 136
16.2.3.2 VoIP Interface per Application Type.... 136
16.2.3.3 VoIP Interfaces for Combined Application Types 137
16.2.3.4 VoIP Interfaces with Multiple Default Gateways 138
16.3 Configuring Static IP Routes 139
16.3.1 Configuration Example of Static IP Routes 140
16.3.2 Troubleshooting the Routing Table 141
16.4 Configuring Quality of Service....142
16.5 DNS....144
16.5.1 Configuring the Internal DNS Table....144
16.5.2 Configuring the Internal SRV Table....145
16.6 Open Solution Network (OSN) Server 147
16.6.1 Configuring Native VLAN for OSN Server....147
16.6.2 Disabling Internal Switch Port for OSN....148
16.7 Configuring NFS Settings....148
16.8 Network Address Translation Support 149
16.8.1 Device Located behind NAT 150
16.8.1.1 Configuring a Static NAT IP Address for All Interfaces.... 151
16.8.1.2 Configuring NAT Translation per IP Interface 151
16.8.2 Remote UA behind NAT 153
16.8.2.1 SIP Signaling Messages 153
16.8.2.2 Media (RTP/RTCP/T.38).... 153
16.9 Robust Receipt of Media Streams by Media Latching 156
16.10 Multiple Routers Support....158

17 Security 159

17.1 Configuring Firewall Settings 159
17.2 Configuring General Security Settings....163
17.3 Intrusion Detection System 164
17.3.1 Enabling IDS....165
17.3.2 Configuring IDS Policies....165
17.3.3 Assigning IDS Policies....169
17.3.4 Viewing IDS Alarms 171

18 Media 173

18.1 Configuring Voice Settings....173
18.1.1 Configuring Voice Gain (Volume) Control 173
18.1.2 Silence Suppression (Compression) 174
18.1.3 Echo Cancellation....174
18.2 Fax and Modem Capabilities....176
18.2.1 Fax/Modem Operating Modes 177
18.2.2 Fax/Modem Transport Modes 177

18.2.2.1 T.38 Fax Relay Mode 177
18.2.2.2 G.711 Fax / Modem Transport Mode 180
18.2.2.3 Fax Fallback 180
18.2.2.4 Fax/Modem Bypass Mode 181
18.2.2.5 Fax / Modem NSE Mode 182
18.2.2.6 Fax / Modem Transparent with Events Mode 183
18.2.2.7 Fax / Modem Transparent Mode 183
18.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection 184
18.2.3 V.34 Fax Support 184
18.2.3.1 Bypass Mechanism for V.34 Fax Transmission 185
18.2.3.2 Relay Mode for T.30 and V.34 Faxes 186
18.2.3.3 V.34 Fax Relay for SG3 Fax Machines 186
18.2.4 V.150.1 Modem Relay 187
18.2.5 Simultaneous Negotiation of Fax (T.38) and Modem (V.150.1) Relay 188
18.2.6 V.152 Support 189
18.2.7 Fax Transmission behind NAT 190
18.3 Configuring RTP/RTCP Settings 190
18.3.1 Configuring the Dynamic Jitter Buffer 190
18.3.2 Comfort Noise Generation 191
18.3.3 Dual-Tone Multi-Frequency Signaling 192
18.3.3.1 Configuring DTMF Transport Types 192
18.3.3.2 Configuring RFC 2833 Payload 193
18.3.4 Configuring RTP Base UDP Port 194
18.4 Configuring IP Media Settings 195
18.4.1 Automatic Gain Control (AGC) 195
18.5 Configuring Various Codec Attributes 196
18.6 Configuring Analog Settings 196
18.7 Configuring Media (SRTP) Security 197
19 Services 199
19.1 SIP-based Media Recording 199
19.1.1 Enabling SIP-based Media Recording 202
19.1.2 Configuring SIP Recording Routing Rules 203
19.1.3 Configuring SIP User Part for SRS 204
19.1.4 Interworking SIP-based Media Recording with Third-Party Vendors 205
19.1.4.1 Genesys 205
19.1.4.2 Avaya UCID 205
19.2 RADIUS Authentication 206
19.2.1 Setting Up a Third-Party RADIUS Server 206
19.2.2 Configuring RADIUS Authentication 208
19.2.3 Securing RADIUS Communication 209
19.2.4 Authenticating RADIUS in the URL 210
19.3 LDAP-based Management and SIP Services 210
19.3.1 Enabling the LDAP Service 212
19.3.2 Enabling LDAP-based Web/CLI User Login Authentication and Authorization 212
19.3.3 Configuring LDAP Servers 212
19.3.4 Configuring LDAP DNs (Base Paths) per LDAP Server 216
19.3.5 Configuring the LDAP Search Filter Attribute 217
19.3.6 Configuring Access Level per Management Groups Attributes 217
19.3.7 Configuring LDAP Search Methods 220
19.3.8 Configuring the Device's LDAP Cache 220
19.3.9 Configuring Local Database for Management User Authentication 223
19.3.10 LDAP-based Login Authentication Example 224
19.3.11 Active Directory-based Routing for Microsoft Lync 228
19.3.11.1 Querying the AD and Routing Priority 228
19.3.11.2 Configuring AD-Based Routing Rules 231

19.3.11.3 Querying the AD for Calling Name 233
19.4 Least Cost Routing 234
19.4.1 Overview 234
19.4.2 Configuring LCR 236
19.4.2.1 Enabling the LCR Feature 236
19.4.2.2 Configuring Cost Groups 238
19.4.2.3 Configuring Time Bands for Cost Groups 239
19.4.2.4 Assigning Cost Groups to Routing Rules 240
19.5 Configuring Call Setup Rules 241
19.5.1 Call Setup Rule Examples 245
20 Quality of Experience 247
20.1 Reporting Voice Quality of Experience to SEM 247
20.1.1 Configuring the SEM Server 247
20.1.2 Configuring Clock Synchronization between Device and SEM 248
20.1.3 Enabling RTCP XR Reporting to SEM 248
20.2 Configuring Quality of Experience Profiles 248
20.3 Configuring Bandwidth Profiles 252
20.4 Configuring Media Enhancement Profiles 255
21 Control Network 259
21.1 Configuring Media Realms 259
21.2 Configuring Remote Media Subnets 262
21.3 Configuring SRDs 264
21.4 Configuring SIP Interfaces 267
21.5 Configuring IP Groups 270
21.6 Configuring Proxy Sets 282
21.7 Assign WAN Interface to VoIP Traffic 289
22 SIP Definitions 291
22.1 Configuring SIP Parameters 291
22.2 Configuring Registration Accounts 291
22.2.1 Regular Registration Mode 294
22.2.2 Single Registration for Multiple Phone Numbers using GIN 294
22.3 Configuring Proxy and Registration Parameters 296
22.3.1 SIP Message Authentication Example 297
22.4 Configuring SIP Message Manipulation 299
22.5 Configuring SIP Message Policy Rules 304
23 Coders and Profiles 307
23.1 Configuring Default Coders 307
23.2 Configuring Coder Groups 310
23.3 Configuring Tel Profile 311
23.4 Configuring IP Profiles 314
24 Introduction 345
25 Digital PSTN 347
25.1 Configuring Trunk Settings 347
25.2 TDM and Timing 350
25.2.1 Configuring TDM Bus Settings 350

25.2.2 Clock Settings....350
25.2.2.1 Recovering Clock from PSTN Line Interface 351
25.2.2.2 Configuring Internal Clock as Clock Source 351
25.3 Configuring CAS State Machines....352
25.4 Configuring Digital Gateway Parameters....354
25.5 Tunneling Applications....355
25.5.1 TDM Tunneling....355
25.5.1.1 DSP Pattern Detector.... 358
25.5.2 QSIG Tunneling....358
25.6 ISDN Non-Facility Associated Signaling (NFAS) 360
25.6.1 NFAS Interface ID....360
25.6.2 Working with DMS-100 Switches 361
25.6.3 Creating an NFAS-Related Trunk Configuration 362
25.6.4 Performing Manual D-Channel Switchover in NFAS Group....362
25.7 ISDN Overlap Dialing....362
25.7.1 Collecting ISDN Digits and Sending Complete Number in SIP....363
25.7.2 Interworking ISDN Overlap Dialing with SIP According to RFC 3578....364
25.8 Redirect Number and Calling Name (Display) 364
26 Trunk Group 365
26.1 Configuring Trunk Group Table....365
26.2 Configuring Hunt Group Settings 367
27 Manipulation 373
27.1 Configuring General Settings....373
27.2 Configuring Source/Destination Number Manipulation Rules 373
27.3 Manipulating Number Prefix....379
27.4 SIP Calling Name Manipulations....380
27.5 Configuring Redirect Number IP to Tel 383
27.6 Manipulating Redirected and Diverted Numbers for Call Diversion 387
27.7 Mapping NPI/TON to SIP Phone-Context....388
27.8 Configuring Release Cause Mapping 390
27.8.1 Fixed Mapping of SIP Response to ISDN Release Reason....391
27.8.2 Fixed Mapping of ISDN Release Reason to SIP Response....392
27.8.3 Reason Header....394
27.8.4 Mapping PSTN Release Cause to SIP Response 395
27.9 Numbering Plans and Type of Number....395
28 Routing....397
28.1 Configuring General Routing Parameters 397
28.2 Configuring Outbound IP Routing Table 397
28.3 Configuring Inbound IP Routing Table....406
28.4 IP Destinations Connectivity Feature....410
28.5 Alternative Routing for Tel-to-IP Calls....412
28.5.1 Alternative Routing Based on IP Connectivity....412
28.5.2 Alternative Routing Based on SIP Responses....413
28.5.3 Alternative Routing upon SIP 3xx with Multiple Contacts....415
28.5.4 PSTN Fallback....416
28.6 Alternative Routing for IP-to-Tel Calls....417
28.6.1 Alternative Routing to Trunk upon Q.931 Call Release Cause Code 417
28.6.2 Alternative Routing to an IP Destination upon a Busy Trunk 418
28.6.3 Alternative Routing upon ISDN Disconnect....420

29 Configuring DTMF and Dialing....421

29.1 Dialing Plan Features....422

29.1.1 Digit Mapping....422

29.1.2 External Dial Plan File....424

30 Configuring Supplementary Services....425

30.1 Call Hold and Retrieve....427

30.2 Call Pickup....429

30.3 BRI Suspend and Resume....429

30.4 Consultation Feature....430

30.5 Call Transfer....430

30.5.1 Consultation Call Transfer....430

30.5.2 Consultation Transfer for QSIG Path Replacement....431

30.5.3 Blind Call Transfer....431

30.6 Call Forward....432

30.6.1 Call Forward Reminder Ring....433

30.6.2 Call Forward Reminder (Off-Hook) Special Dial Tone....433

30.6.3 Call Forward Reminder Dial Tone (Off-Hook) upon Spanish SIP Alert-Info....434

30.6.4 BRI Call Forwarding....434

30.7 Call Waiting....435

30.8 Message Waiting Indication....436

30.9 Caller ID....438

30.9.1 Caller ID Detection / Generation on the Tel Side....438

30.9.2 Debugging a Caller ID Detection on FXO....438

30.9.3 Caller ID on the IP Side....439

30.10 Three-Way Conferencing....440

30.11 Emergency E911 Phone Number Services....442

30.11.1 FXS Device Emulating PSAP using DID Loop-Start Lines....442

30.11.2 FXO Device Interworking SIP E911 Calls from Service Provider's IP Network to PSAP DID Lines....445

30.11.3 Pre-empting Existing Calls for E911 IP-to-Tel Calls....448

30.11.4 Enhanced 9-1-1 Support for Lync Server 2010....449

30.11.4.1 About E9-1-1 Services....449

30.11.4.2 Microsoft Lync Server 2010 and E9-1-1....450

30.11.4.3 AudioCodes ELIN Gateway for Lync Server 2010 E9-1-1 Calls to PSTN 454

30.11.4.4 Configuring AudioCodes ELIN Gateway....459

30.12 Multilevel Precedence and Preemption....461

30.12.1 MLPP Preemption Events in SIP Reason Header....464

30.12.2 Precedence Ring Tone....465

30.13 Denial of Collect Calls....465

30.14 Configuring Multi-Line Extensions and Supplementary Services....465

30.15 Detecting Collect Calls....468

30.16 Advice of Charge Services for Euro ISDN....468

30.17 Configuring Charge Codes....470

30.18 Configuring Voice Mail....472

31 Analog Gateway....473

31.1 Configuring Keypad Features....473

31.2 Configuring Metering Tones....474

31.3 Configuring FXO Settings 476
31.4 Configuring Authentication 477
31.5 Configuring Automatic Dialing..478
31.6 Configuring Caller Display Information..480
31.7 Configuring Call Forward 481
31.8 Configuring Caller ID Permissions 483
31.9 Configuring Call Waiting..484
31.10 Rejecting Anonymous Calls 485
31.11 Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number 485
31.12 FXS/FXO Coefficient Types 487
31.13 FXO Operating Modes 487
31.13.1 FXO Operations for IP-to-Tel Calls..488
31.13.1.1 One-Stage Dialing 488
31.13.1.2 Two-Stage Dialing 489
31.13.1.3 DID Wink 489
31.13.2 FXO Operations for Tel-to-IP Calls..490
31.13.2.1 Automatic Dialing 490
31.13.2.2 Collecting Digits Mode.. 491
31.13.2.3 FXO Supplementary Services.. 491
31.13.3 Call Termination on FXO Devices 492
31.13.3.1 Calls Termination by PBX 492
31.13.3.2 Call Termination before Call Establishment.. 493
31.13.3.3 Ring Detection Timeout.. 493
31.14 Remote PBX Extension between FXO and FXS Devices..493
31.14.1 Dialing from Remote Extension (Phone at FXS) 494
31.14.2 Dialing from PBX Line or PSTN..494
31.14.3 Message Waiting Indication for Remote Extensions 495
31.14.4 Call Waiting for Remote Extensions 495
31.14.5 FXS Gateway Configuration 496
31.14.6 FXO Gateway Configuration..497
32 SBC Overview..501
32.1 SIP Network Definitions 502
32.2 SIP Dialog Initiation Process..502
32.3 User Registration 504
32.3.1 Initial Registration Request Processing..505
32.3.2 SBC Users Registration Database 505
32.3.3 Routing using Users Registration Database..506
32.3.4 Registration Refreshes 506
32.3.5 Registration Restriction Control..507
32.4 SBC Media Handling..507
32.4.1 Media Anchoring without Transcoding (Transparent) 508
32.4.2 Media Anchoring with Transcoding 509
32.4.3 No Media Anchoring 511
32.4.4 Transcoding Modes 512
32.4.5 Restricting Coders 513
32.4.6 Coder Transcoding 513
32.4.7 Prioritizing Coder List in SDP Offer 515
32.4.8 SRTP-RTP and SRTP-SRTP Transcoding 515
32.4.9 Multiple RTP Media Streams per Call Session 516
32.4.10 Interworking DTMF Methods 516
32.5 Fax Negotiation and Transcoding 516
32.6 Limiting SBC Call Duration..517

32.7 SBC Authentication 517
32.7.1 SIP Authentication Server Functionality 517
32.7.2 User Authentication based on RADIUS....518
32.8 Interworking SIP Signaling 518
32.8.1 Interworking SIP 3xx Redirect Responses 519
32.8.1.1 Resultant INVITE Traversing Device 519
32.8.1.2 Local Handling of SIP 3xx 520
32.8.2 Interworking SIP Diversion and History-Info Headers 521
32.8.3 Interworking SIP REFER Messages....521
32.8.4 Interworking SIP PRACK Messages 522
32.8.5 Interworking SIP Session Timer 522
32.8.6 Interworking SIP Early Media 522
32.8.7 Interworking SIP re-INVITE Messages....524
32.8.8 Interworking SIP UPDATE Messages 525
32.8.9 Interworking SIP re-INVITE to UPDATE....525
32.8.10 Interworking Delayed Offer 525
32.8.11 Interworking Call Hold....525
32.9 Call Survivability....526
32.9.1 Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server for Survivability....526
32.9.2 BroadSoft's Shared Phone Line Call Appearance for SBC Survivability....526
32.9.3 Call Survivability for Call Centers 528
32.9.4 Survivability Mode Display on Aastra IP Phones 530
32.10 Call Forking....531
32.10.1 Initiating SIP Call Forking 531
32.10.2 SIP Forking Initiated by SIP Proxy Server....531
32.10.3 Call Forking-based IP-to-IP Routing Rules....532
32.11 Alternative Routing on Detection of Failed SIP Response 532
33 Enabling the SBC Application....533
34 Configuring General Settings 535
34.1 Interworking Dialog Information in SIP NOTIFY Messages 535
35 Configuring Admission Control....539
36 Configuring Coder Groups....543
36.1 Configuring Allowed Audio Coder Groups 543
36.2 Configuring Allowed Video Coder Groups 544
37 Routing SBC 545
37.1 Configuring Classification Rules 545
37.1.1 Classification Based on URI of Selected Header Example....550
37.2 Configuring Message Condition Rules....551
37.3 Configuring SBC IP-to-IP Routing....552
37.4 Configuring SIP Response Codes for Alternative Routing Reasons....561
38 SBC Manipulations....563
38.1 Configuring IP-to-IP Inbound Manipulations 565
38.2 Configuring IP-to-IP Outbound Manipulations....568

39 CRP Overview....575

40 CRP Configuration....577

40.1 Enabling the CRP Application....577
40.2 Configuring Call Survivability Mode....578
40.3 Pre-Configured IP Groups....579
40.4 Pre-Configured IP-to-IP Routing Rules....580
40.4.1 Normal Mode....580
40.4.2 Emergency Mode....581
40.4.3 Auto Answer to Registrations....581
40.5 Configuring PSTN Fallback....582

41 Introduction....585

42 Basic Maintenance....589

42.1 Resetting the Device....589
42.2 Remotely Resetting Device using SIP NOTIFY....590
42.3 Locking and Unlocking the Device....591
42.4 Saving Configuration....592

43 Disconnecting Active Calls....593

44 Resetting Channels....595

44.1 Resetting an Analog Channel....595
44.2 Restarting a B-Channel....595

45 Disabling Analog Ports....597

46 Software Upgrade....599

46.1 Loading Auxiliary Files....599
46.1.1 Call Progress Tones File....600
46.1.1.1 Distinctive Ringing....603
46.1.2 Prerecorded Tones File....604
46.1.3 CAS Files....605
46.1.4 Dial Plan File....606
46.1.4.1 Creating a Dial Plan File....606
46.1.4.2 External Dial Plan File....606
46.1.4.3 Dial Plan Prefix Tags for Routing....608
46.1.4.4 Obtaining IP Destination from Dial Plan File....612
46.1.4.5 Modifying ISDN-to-IP Calling Party Number....613
46.1.5 User Information File....614
46.1.5.1 Enabling the User Info Table....614
46.1.5.2 Gateway User Information for PBX Extensions and "Global" Numbers 614
46.1.5.3 User Information File for SBC User Database....618
46.2 Software License Key....621
46.2.1 Obtaining the Software License Key File....621
46.2.2 Installing the Software License Key....622
46.2.2.1 Installing Software License Key using Web Interface....623
46.2.2.2 Installing Software License Key using CLI....624
46.3 Software Upgrade Wizard....625
46.4 Backing Up and Loading Configuration File....629

47 Automatic Provisioning Mechanisms....631

47.1 Automatic Configuration Methods....631
47.1.1 DHCP-based Provisioning....631

47.1.2 HTTP-based Provisioning....................632
47.1.3 FTP- or NFS-based Provisioning....................633
47.1.4 Provisioning using AudioCodes EMS....................633

47.2 HTTP/S-Based Provisioning using the Automatic Update Feature....................634

47.2.1 Files Provisioned by Automatic Update....................634
47.2.2 File Location for Automatic Update....................635
47.2.3 Triggers for Automatic Update....................636
47.2.4 Access Authentication with HTTP Server....................636
47.2.5 Querying Provisioning Server for Updated Files....................637
47.2.6 File Download Sequence....................639
47.2.7 Cyclic Redundancy Check on Downloaded Configuration Files....................640
47.2.8 MAC Address Automatically Inserted in Configuration File Name....................640
47.2.9 Automatic Update Configuration Examples....................641
47.2.9.1 Automatic Update for Single Device.................... 641
47.2.9.2 Automatic Update from NFS, FTP and HTTP Servers.................... 642
47.2.9.3 Automatic Update for Mass Deployment.................... 644

47.3 Zero Configuration....................646

47.3.1 Zero Configuration Process....................646
47.3.2 Configuring Zero Configuration....................648
47.3.3 Using Zero Configuration with Automatic Update....................649

47.4 Automatic Provisioning using USB Flash Drive....................651

48 Restoring Factory Defaults....................653

48.1 Restoring Defaults using CLI....................653
48.2 Restoring Defaults using Hardware Reset Button....................654
48.3 Restoring Defaults using an ini File....................654

49 Automatic Archiving of Configuration File....................655

50 USB Storage Capabilities....................657

51 System Status....................661

51.1 Viewing Device Information....................661
51.2 Viewing Ethernet Port Information....................662

52 Carrier-Grade Alarms....................665

52.1 Viewing Active Alarms....................665
52.2 Viewing Alarm History....................666

53 Performance Monitoring....................667

53.1 Viewing MOS per Media Realm....................667
53.2 Viewing Trunk Utilization....................668
53.3 Viewing Quality of Experience....................669
53.4 Viewing Average Call Duration....................671
53.5 Network Monitoring (Probing) Two Devices....................672

54 VoIP Status....................677

54.1 Viewing Trunks & Channels Status....................677
54.2 Viewing Analog Port Information....................679
54.3 Viewing NFAS Groups and D-Channel Status....................679
54.4 Viewing Active IP Interfaces....................680
54.5 Viewing Ethernet Device Status....................680
54.6 Viewing Static Routes Status....................681

54.7 Viewing Performance Statistics....681
54.8 Viewing Call Counters....682
54.9 Viewing Registered Users....684
54.10 Viewing Registration Status....685
54.11 Viewing Call Routing Status....686
54.12 Viewing IP Connectivity....687

55 Reporting Information to External Party....689

55.1 Configuring RTCP XR....689
55.2 Generating Call Detail Records....692
55.2.1 Configuring CDR Reporting....693
55.2.2 CDR Field Description....693
55.2.2.1 CDR Fields for SBC Signaling....693
55.2.2.2 CDR Fields for SBC Media....696
55.2.2.3 CDR Fields for Gateway/IP-to-IP Application....697
55.2.2.4 Release Reasons in CDR....701
55.3 Configuring RADIUS Accounting....704
55.4 Event Notification using X-Detect Header....707
55.5 Querying Device Channel Resources using SIP OPTIONS....710

56 Obtaining Status and Performance using a USB Flash Drive....711

57 Syslog and Debug Recordings....715

57.1 Syslog Message Format....715
57.1.1 Event Representation in Syslog Messages....716
57.1.2 Unique Device Identification in Syslog Messages....718
57.1.3 Identifying AudioCodes Syslog Messages using Facility Levels....718
57.1.4 SNMP Alarms in Syslog Messages....719
57.2 Enabling Syslog....719
57.3 Configuring Web Operations to Report to Syslog....720
57.4 Configuring Debug Recording....721
57.5 Filtering Syslog Messages and Debug Recordings....722
57.5.1 Filtering IP Network Traces....724
57.6 Viewing Syslog Messages....725
57.7 Collecting Debug Recording Messages....726
57.8 Debug Capturing VoIP and Data-Router Traffic....728
57.9 Debug Capturing on Physical VoIP Interfaces....729
57.10 Configuring Termination of Debug Capture Upon Event....730

58 Self-Testing....731

59 Creating Core Dump and Debug Files upon Device Crash....733

60 Re-initializing Device with "Purified" Configuration....735

61 Analog Line Testing....737

61.1 FXO Line Testing....737
61.2 FXS Line Testing....737

62 Testing SIP Signaling Calls....739

62.1 Configuring Test Call Endpoints....739
62.2 Starting and Stopping Test Calls....744
62.3 Viewing Test Call Statistics....744

62.4 Configuring DTMF Tones for Test Calls....746
62.5 Configuring Basic Test Call....747
62.6 Configuring SBC Test Call with External Proxy....748
62.7 Test Call Configuration Examples....749

63 Data-Router Debugging....753

63.1 Loopback on WAN Interface Debugging....753
63.2 Performing a Traceroute....754

64 Pinging a Remote Host or IP Address....755

65 Troubleshooting using a USB Flash Drive....757

66 Dialing Plan Notation for Routing and Manipulation....761

67 Configuration Parameters Reference....763

67.1 Management Parameters....763
67.1.1 General Parameters....763
67.1.2 Web Parameters....764
67.1.3 Telnet Parameters....767
67.1.4 ini File Parameters....767
67.1.5 SNMP Parameters....768
67.1.6 TR-069 Parameters....771
67.1.7 Serial Parameters....772
67.1.8 Auxiliary and Configuration File Name Parameters....773
67.1.9 Automatic Update Parameters....774
67.2 Networking Parameters....778
67.2.1 Multiple VoIP Network Interfaces and VLAN Parameters....778
67.2.2 Routing Parameters....778
67.2.3 Open Solution Network (OSN) Parameters....779
67.2.4 Quality of Service Parameters....779
67.2.5 NAT and STUN Parameters....780
67.2.6 NFS Parameters....781
67.2.7 DNS Parameters....782
67.2.8 DHCP Parameters....782
67.2.9 NTP and Daylight Saving Time Parameters....783
67.2.10 Power over Ethernet Parameters....785
67.3 Debugging and Diagnostics Parameters....785
67.3.1 General Parameters....785
67.3.2 SIP Test Call Parameters....787
67.3.3 Syslog, CDR and Debug Parameters....788
67.3.4 Resource Allocation Indication Parameters....793
67.4 Security Parameters....794
67.4.1 General Security Parameters....794
67.4.2 HTTPS Parameters....796
67.4.3 SRTP Parameters....797
67.4.4 TLS Parameters....800
67.4.5 SSH Parameters....802
67.4.6 TAACS+ Parameters....803
67.4.7 IDS Parameters....803
67.4.8 OCSP Parameters....804
67.5 Quality of Experience Parameters....805
67.6 Control Network Parameters....807
67.6.1 IP Group, Proxy, Registration and Authentication Parameters....807
67.6.2 Network Application Parameters....819

67.7 General SIP Parameters 821
67.8 Coders and Profile Parameters 849
67.9 Channel Parameters 851
67.9.1 Voice Parameters 851
67.9.2 Coder Parameters 854
67.9.3 DTMF Parameters 855
67.9.4 RTP, RTCP and T.38 Parameters 856
67.10 Gateway and IP-to-IP Parameters 861
67.10.1 Fax and Modem Parameters 861
67.10.2 DTMF and Hook-Flash Parameters 867
67.10.3 Digit Collection and Dial Plan Parameters 872
67.10.4 Voice Mail Parameters 874
67.10.5 Supplementary Services Parameters 880
67.10.5.1 Caller ID Parameters 880
67.10.5.2 Call Waiting Parameters 885
67.10.5.3 Call Forwarding Parameters 887
67.10.5.4 Message Waiting Indication Parameters 888
67.10.5.5 Call Hold Parameters 890
67.10.5.6 Call Transfer Parameters 891
67.10.5.7 Multi-Line Extensions and Supplementary Services Parameters 894
67.10.5.8 Three-Way Conferencing Parameters 894
67.10.5.9 MLPP and Emergency Call Parameters 896
67.10.5.10 Call Cut-Through Parameters 902
67.10.5.11 Automatic Dialing Parameters 903
67.10.5.12 Direct Inward Dialing Parameters 904
67.10.5.13 ISDN BRI Parameters 906
67.10.6 PSTN Parameters 907
67.10.6.1 General Parameters 907
67.10.6.2 TDM Bus and Clock Timing Parameters 912
67.10.6.3 CAS Parameters 914
67.10.6.4 ISDN Parameters 917
67.10.7 ISDN and CAS Interworking Parameters 925
67.10.8 Answer and Disconnect Supervision Parameters 941
67.10.9 Tone Parameters 946
67.10.9.1 Telephony Tone Parameters 946
67.10.9.2 Tone Detection Parameters 953
67.10.9.3 Metering Tone Parameters 954
67.10.10 Telephone Keypad Sequence Parameters 956
67.10.11 FXO and FXS Parameters 959
67.10.12 Trunk Groups and Routing Parameters 963
67.10.13 IP Connectivity Parameters 970
67.10.14 Alternative Routing Parameters 971
67.10.15 Number Manipulation Parameters 974
67.11 SBC Parameters 983
67.12 Standalone Survivability Parameters 997
67.13 IP Media Parameters 1002
67.14 Services 1005
67.14.1 SIP-based Media Recording Parameters 1005
67.14.2 RADIUS and LDAP Parameters 1007
67.14.2.1 General Parameters 1007
67.14.2.2 RADIUS Parameters 1007
67.14.2.3 LDAP Parameters 1009
67.14.3 Least Cost Routing Parameters 1012
67.14.4 Call Setup Rules Parameters 1014
68 SBC and DSP Channel Capacity 1015
68.1 Signaling-Media Sessions & User Registrations 1015

68.2 Channel Capacity and Capabilities 1017

69 Technical Specifications 1021

This page is intentionallt left blank.

Notice

This document describes AudioCodes Mediant 800 Multi-Service Business Router (MSBR).

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

Date Published: September-07-2014

Trademarks

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

WEEE EU Directive

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

Customer Support

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

Abbreviations and Terminology

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

Related Documentation

Manual Name
SIP CPE Release Notes
Mediant 800 MSBR Hardware Installation Manual
Complementary Guides
CLI Reference Guide
CPE Configuration Guide for IP Voice Mail
SNMP User's Guide
CWMP TR-069 & TR-104 Reference Guide
SBC Design Guide
Recommended Security Guidelines Configuration Note
SIP Message Manipulations Quick Reference Guide
SAS Application Configuration Guide
CAS Protocol Table Configuration Note
IP-to-IP Application Configuration Guide Ver. 6 8
Utility Guides
INI Viewer & Editor Utility User's Guide
DConvert 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, FXS and 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 products.

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

Note: The device's installed Software License Key does not include the MSFT feature key, which enables the device to operate in a Microsoft Lync Server environment. If necessary, you can order this feature key separately from your AudioCodes sales representative.

Notes:

- For data-router configuration, refer to the CLI Reference Guide.
- Web-based management for data-router functionality is not supported. Instead, CLI is used to configure this functionality. However, AudioCodes recommends using CLI scripting to configure all other functionality as well (i.e., VoIP and System) through the CLI.

Notes:

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

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

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

Documentation Feedback

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

1 Overview

The Mediant 800B Multi-Service Business Router (MSBR) is a networking device that combines multiple service functions such as a Media Gateway, Session Border Controller (SBC), Data Router and Firewall, LAN switch, WAN access, Stand Alone Survivability (SAS) and an integrated general-purpose server. The device offers enhanced dialing plans and voice routing capabilities along with SIP-to-SIP mediation, allowing enterprises to implement SIP Trunking services (IP-to-IP call routing) and IP-based Unified Communications, as well as flexible PSTN and legacy PBX connectivity.

The device is designed as a secured Voice-over-IP (VoIP) and data platform. Enhanced media gateway security features include, for example, SRTP for media, TLS for SIP control, and IPSec for management. Data security functions include integrated Stateful Firewall, IDS/IPS, SSL for remote user access, and site-to-site VPN. A fully featured enterprise class SBC provides a secured voice network deployment based on a Back-to-Back User Agent (B2BUA) implementation.

The device's SAS functionality offers service continuity to enterprises served by a centralized SIP-based IP-Centrex server or branch offices of distributed enterprises. SAS enables internal office communication between SIP clients, along with PSTN fallback in the case of disconnection from the centralized SIP IP-Centrex server or IP-PBX.

The device provides Foreign Exchange Station (FXS) and/or Foreign Exchange Office (FXO) telephony module interfaces, depending on ordered hardware configuration. The device supports either a combination of FXS and FXO port interfaces, or only FXS or only FXO interfaces. The device can support up to 12 simultaneous VoIP calls. Each FXS or FXO module provides four analog RJ-11 ports. The FXO module can be used to connect analog lines of an enterprise's PBX or the PSTN, to the IP network. The FXS module can be used to connect legacy telephones, fax machines, and modems to the IP network. Optionally, the FXS module can be connected to the external trunk lines of a PBX. When deployed with a combination of FXO and FXS modules, the device can be used as a PBX for Small Office Home Office (SOHO) users, and businesses not equipped with a PBX. The FXS modules also support the Analog Lifeline feature, enabling an FXS port to connect directly to the PSTN upon power and/or network failure.

The device supports up to 8 ISDN Basic Rate Interface (BRI) S/T interfaces (RJ-45 ports), supporting up to 16 voice channels. These connect ISDN terminal equipment such as ISDN telephones. The device also provides an optional, single or dual E1/T1 interface port(s), supporting Transparent, CAS and ISDN protocols. The device supports various ISDN PRI protocols such as Euro ISDN, North American NI-2, Lucent™ 4/5ESS, Nortel™ DMS-100 and others. It also supports various ISDN BRI protocols such as ETSI 5ESS and QSIG over BRI. It also supports different variants of CAS protocols for E1 and T1 spans, including MFC R2, E&M immediate start, E&M delay dial / start, loop start and ground start.

Note: For maximum call capacity figures, see "SBC and DSP Channel Capacity" on page 1015.

The device provides two USB ports that can be used for an optional, 3G cellular WAN modem and/or USB storage services.

The device's data routing capabilities support static and dynamic routing protocols such as RIP/OSPF and BGP, Virtual Routing and Forwarding (VRF-Lite) where interfaces can be clustered into a VRF to provide segregated routing domains. The device supports various optional WAN interfaces, providing flexibility in connecting to Service Providers:

- 1000Base-T Gigabit Ethernet copper.
- Symmetric High-Speed Digital Subscriber Line (SHDSL) - supports up to four copper wire pairs according to G.991.2, acting as a remote-terminal CPE device. Both ATM and EFM modes are supported. In the ATM mode, a variety of protocols are

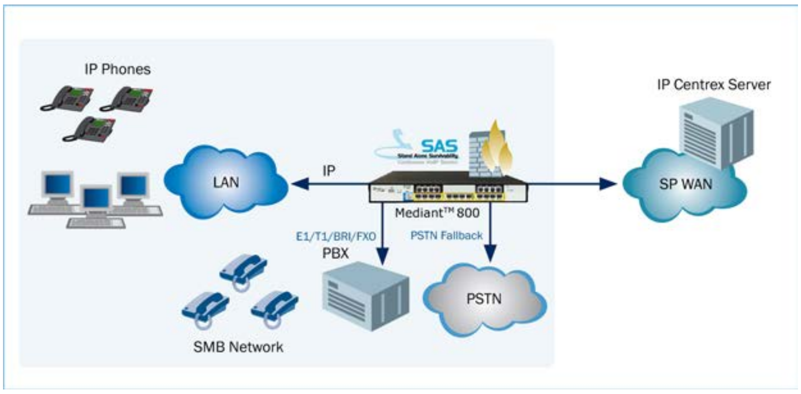
supported, including PPPoE, PPPoA, and RFC 2684 in both bridged (Ethernet-over-ATM) and routed (IP-over-ATM) variants. In the EFM mode, the SHDSL port functions as a logical Ethernet device.

- ADSL2+ / VDSL2 (RJ-11 port interfaces)
- Optical Fiber, supporting 100 and 1000 Mbps Ethernet
- Four E1/T1 WANs
- 3G Cellular modem using a USB connection - this can be used as the primary WAN interface or as a WAN backup in case of failure in the WAN connection (provided by any of the above).

The device is optimized for wire-speed delivery of data, providing up to 12 Ethernet LAN ports for connecting equipment such as computers and IP phones. These ports are divided into Gigabit Ethernet and Fast Ethernet interfaces (the number depends on the ordered configuration), and provide power-over-Ethernet (PoE) capabilities. The device also supports an optional, Wi-Fi interface, providing wireless LAN 802.11n access point at 2.4 GHz, 3Tx/3Rx enabling data rates of up to 300 Mbps. The Wi-Fi interface also supports 802.11b/802.11g backward compatibility, allowing interoperability of multiple devices with different types of Wi-Fi

The device also provides an integrated Open Solution Network (OSN) Server module. The OSN can host a variety of third-party applications such as IP-PBX, Call Center, and Conferencing.

Figure 1-1: Typical Application



The device allows full management through its command line interface (CLI) as well as its HTTP/S-based embedded Web server. The user-friendly Web interface allows remote configuration using any standard Web browser (such as Microsoft™ Internet Explorer™).

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. Device management can be done though the VoIP-LAN OAMP, WAN, and/or LAN interface.

Note: By default, the device's embedded DHCP server is enabled. For more information, see Configuring the Device's DHCP Server on page 36.

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
Prefix Length	255.255.255.0 (24)
Default Gateway	192.168.0.1
Underlying Device	1
Interface Name	"Voice"

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:

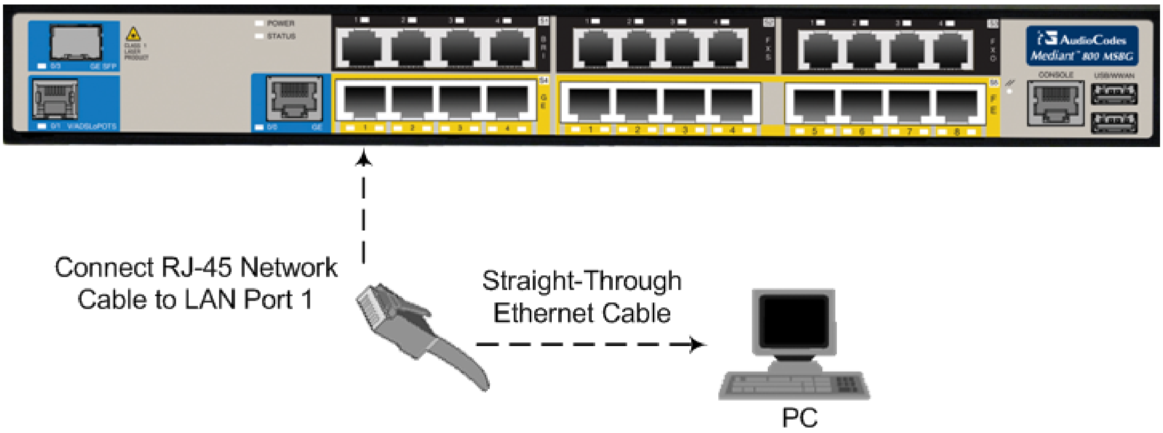
- Embedded HTTP/S-based Web server - see "Web Interface" on page 31
- Embedded command line interface (CLI) - see "CLI" on page 33

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.

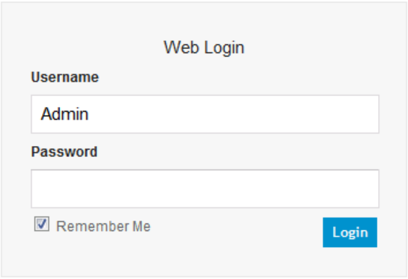
➢ **To configure the VoIP-LAN IP Address for OAMP, using the Web interface:**

1. Connect Port 1 (left-most LAN port) located on the front panel directly to the network interface of your computer, using a straight-through Ethernet cable.


AudioCodes Mediant 800 MSBG
Connect RJ-45 Network Cable to LAN Port 1
Straight-Through Ethernet Cable
PC

2. Make sure that your computer is configured to automatically obtain an IP address. The device has an embedded DHCP server, which by default allocates IP addresses to connected computers.
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



Web Login

Username

Admin

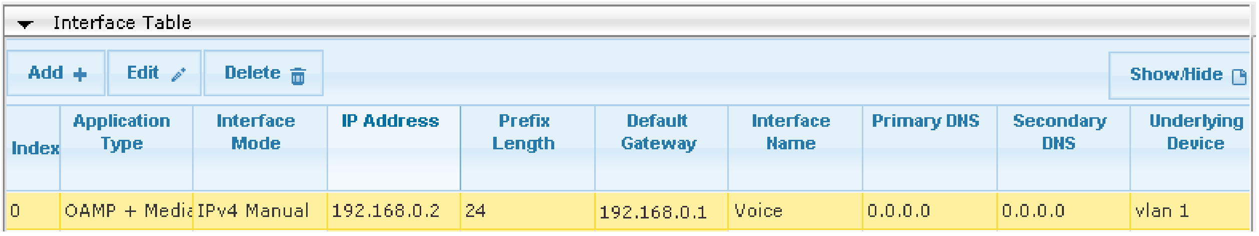
Password

☑ Remember Me

Login

   b. In the 'Username' and 'Password' fields, enter the case-sensitive, default login username ("Admin") and password ("Admin").
   c. Click **Login**.

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



Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Interface Name	Primary DNS	Secondary DNS	Underlying Device
0	OAMP + Medi	IPv4 Manual	192.168.0.2	24	192.168.0.1	Voice	0.0.0.0	0.0.0.0	vlan 1

5. Select the 'Index' radio button corresponding to the **OAMP + Media + Control** application type, and then click **Edit**.

6. 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

7. Click **Submit**.

8. Save your settings by resetting the device with a flash burn (see "Resetting the Device" on page 589).

9. 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.

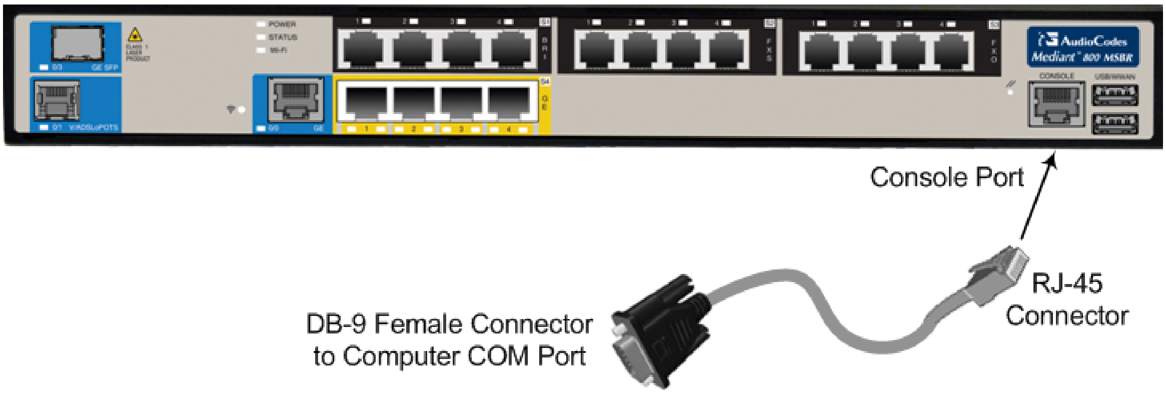
Note: When you complete the above procedure, change your PC's IP address to correspond with your network requirements.

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. Exit the Interface table:

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

13. Exit the VoIP configuration mode:

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

14. Reset the device with a flash burn:

```
# reload now
```

15. Cable the device to your network. You can now access the device's management interface using this new OAMP IP address.

5 Configuring Data-Router's LAN and WAN

This section describes how to configure the device's data-router LAN and/or WAN interfaces.

Notes:

- Make sure that you configure the LAN IP address of the data-router in the same subnet as the VoIP-LAN IP address for OAMP.
- After you access the device through the default VoIP-LAN interface, you can configure Web management access from one of the following interfaces:
  - √ **Any of the configured data-router LAN interfaces:** The default LAN data interface is 192.168.0.1. This interface can be in a different subnet to the VoIP-LAN IP address and with a different VLAN ID. This is useful, for example, if you want to separate management from the VoIP traffic.
  - √ **WAN port interface:** In this setup, you need to enable remote access to the WAN port interface, as described in "Enabling Remote Management from WAN" on page 39.

5.1 Configuring Data-Router's LAN Interface

The device's default LAN IP address of the data-router is listed below:

- **IP Address:** 192.168.0.1
- **Subnet Mask:** 255.255.255.0
- **Default Gateway:** 0.0.0.0

➢ **To configure LAN IP address of data-router:**

1. Establish serial communication with the device.
2. At the prompt, type the following command to access the Data-router configuration mode:

```
# configure data
```

3. Access the VLAN 1 LAN switch interface:

```
(config-data)# interface vlan 1
```

4. Configure the IP address and subnet:

```
(conf-if-VLAN 1)# ip address <IP address> <subnet>
```

For example:

```
(conf-if-VLAN 1)# ip address 10.8.6.85 255.255.255.0
```

5. Save your settings with a flash burn:

```
(conf-if-VLAN 1)# do write
```

5.2 Configuring the Device's DHCP Server

By default, the device's embedded DHCP server is enabled for the LAN, and with default IP pool addresses relating to the default subnet LAN. You can disable the DHCP server, or modify the IP address pool. The DHCP server allocates this spool of IP addresses to the computers connected to its LAN interface.

➢ **To enable / disable the device's DHCP server:**

1. Establish serial communication with the device.
2. At the prompt, type the following command to access the Data-router configuration mode:

```
# configure data
```

3. Access the data LAN switch interface:

```
(config-data)# interface vlan 1
```

4. To disable the DHCP server:

```
(conf-if-VLAN 1)# no service dhcp
```

5. To enable DHCP server:
   a. Configure the pool of IP addresses:

```
(conf-if-VLAN 1)# ip dhcp-server network 10.8.6.84 10.8.6.89
255.255.255.0
```

   b. Enable DHCP server functionality:

```
(conf-if-VLAN 1)# service dhcp
```

6. Save your settings with a flash burn:

```
(conf-if-VLAN 1)# do write
```

5.3 Configuring the WAN Interface

This procedure describes how to configure the WAN interface and uses Gigabit Ethernet as an example. If you are using a different WAN interface, refer to the *MSBR Series CLI Reference Guide for Data*.

Note: Before you configure the WAN interface, make sure that you have all the required information from your Internet Telephony Service Provider (ITSP).

➢ **To configure a WAN IP address:**

1. Connect the WAN port to the WAN network. For information on cabling the WAN port, refer to the *Hardware Installation Manual*.
2. Establish serial communication with the device.
3. At the prompt, type the following command to access the Data-router configuration mode:

```
# configure data
```

4. Access the WAN interface:

```
(config-data)# interface GigabitEthernet 0/0
```

5. Configure the IP address and subnet mask:

```
(config-if-GE 0/0)# ip address 100.33.2.105  255.255.255.0
```

6. Enable Network Address Port Translation (NAPT) on the WAN interface:

   ```
   (config-if-GE 0/0)# napt
   ```

7. Enable the WAN interface:

   ```
   (config-if-GE 0/0)# no shutdown
   ```

8. Exit the interface:

   ```
   (config-if-GE 0/0)# exit
   ```

9. Configure the default route:

   ```
   (config-data)# ip route 0.0.0.0 0.0.0.0 100.33.2.106
   GigabitEthernet 0/0
   ```

10. Exit the data-router configuration mode:

    ```
    (config-data)# exit
    ```

11. Save the configuration to flash:

    ```
    # write
    ```

This page is intentionally left blank.

6 Enabling Remote Management from WAN

This section describes how to configure remote device management from the WAN.

6.1 Remote Web-based (HTTP/S) Management

This procedure describes how to enable remote Web-based management (HTTP/S) from the WAN.

➢ **To enable remote Web (HTTP/S) management from WAN:**

- **CLI:**

  1. Access the System configuration mode:

     ```
     # configure system
     ```

  2. Enable HTTP management from the WAN:

     ```
     <config-system># web
     <web># wan-http on
     ```

  3. Reset the device with a burn to flash:

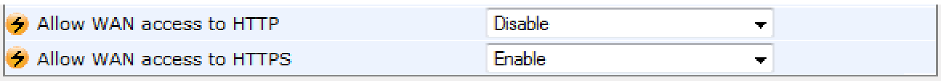
     ```
     <web># do reload now
     ```

- **Web:**

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

Table 6-1: Enabling Web Management from WAN

Figure 6-1: Defining WAN HTTP Port



Allow WAN access to HTTP	Disable
Allow WAN access to HTTPS	Enable

  2. From the 'Allow WAN access to HTTPS' or 'Allow WAN access to HTTP' drop-down list, select **Enable**.

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

6.2 Remote Telnet-based Management

This procedure describes how to enable remote Telnet-based management from the WAN.

➢ **To enable remote Telnet management from WAN:**

- **CLI:**

1. Access the System configuration mode:

```
# configure system
```

2. Type the following command:

```
<config-system># cli-terminal
```

3. Enable Telnet:

```
<cli-terminal># telnet
```

4. Enable Telnet from WAN:

```
<cli-terminal># wan-telnet-allow on
```

5. Reset the device with a burn to flash:

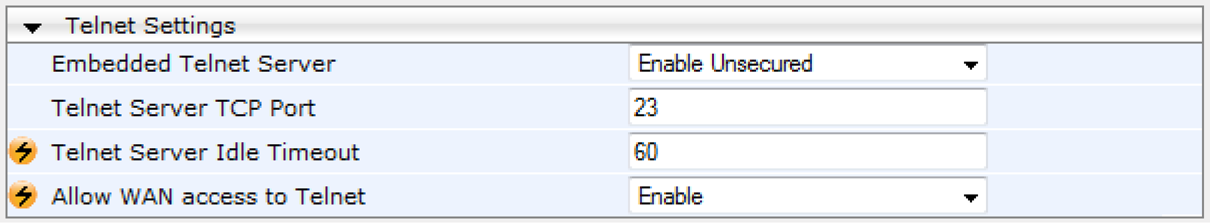
```
<cli-terminal># do reload now
```

- **Web:**

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

Table 6-2: Enabling Telnet Management from WAN

Figure 6-2: Telnet Settings on Telnet/SSH Settings Page



Telnet Settings

Embedded Telnet Server	Enable Unsecured
Telnet Server TCP Port	23
Telnet Server Idle Timeout	60
Allow WAN access to Telnet	Enable

2. From the 'Embedded Telnet Server' drop-down list, select **Enable Secured**.
3. From the 'Allow WAN access to Telnet' drop-down list, select **Enable**.
4. Click **Submit**.
5. Save your settings with a flash burn.

Part II

Management Tools

7 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 45
- Command Line Interface (CLI) - see "CLI-Based Management" on page 75
- Simple Network Management Protocol (SNMP) - see "SNMP-Based Management" on page 87
- TR-069 - see TR-069 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 763.

This page is intentionally left blank.

8 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, data routing, 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 621).

8.1 Getting Acquainted with the Web Interface

This section provides a description of the Web interface.

8.1.1 Computer Requirements

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

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

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

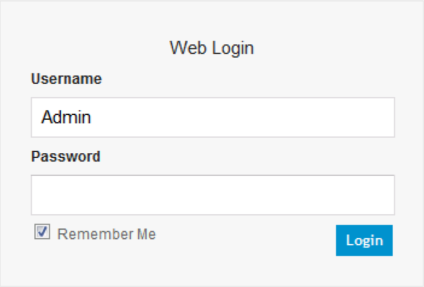
8.1.2 Accessing the Web Interface

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 45).
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 8-1: Web Login Screen



Web Login

Username

Admin

Password

☑ Remember Me

3. In the 'Username' and 'Password' fields, enter the case-sensitive, user name and password respectively.
4. Click **Login**; the Web interface is accessed, displaying the Home page. For a detailed description of the Home page, see "Viewing the Home Page" on page 60.

Notes:

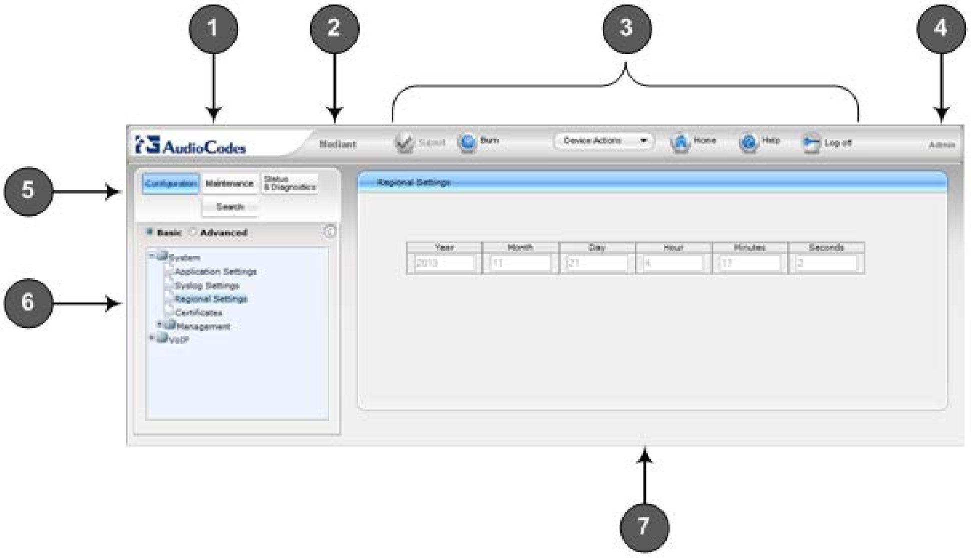
- The default login username and password is "Admin". To change the login credentials, see "Configuring the Web User Accounts" on page 64.
- If you want the Web browser to remember your password, select the 'Remember Me' check box and then agree to the browser's prompt (depending on your browser) to save the password for future logins. On your next login attempt, simply press the Tab or Enter keys to auto-fill the 'Username' and 'Password' fields, and then click **Login**.
- Depending on your Web browser's settings, a security warning box may be displayed. The reason for this is that the device's certificate is not trusted by your PC. The browser may allow you to install the certificate, thus skipping the warning box the next time you connect to the device. If you are using Windows Internet Explorer, click **View Certificate**, and then **Install Certificate**. The browser also warns you if the host name used in the URL is not identical to the one listed in the certificate. To resolve this, add the IP address and host name (ACL_nnnnnn, where *nnnnnn* is the serial number of the device) to your hosts file, located at /etc/hosts on UNIX or C:\Windows\System32\Drivers\ETC\hosts on Windows; then use the host name in the URL (e.g., https://ACL_280152). Below is an example of a host file:

```
127.0.0.1    localhost
10.31.4.47   ACL_280152
```

8.1.3 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 8-2: Main Areas of the Web Interface GUI



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

Table 8-1: 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 48.
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 51) ▪ **Search** tab: Enables a search engine for searching configuration parameters (see "Searching for Configuration Parameters" on page 55)
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 48.
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 51.

8.1.4 Toolbar Description

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

Table 8-2: Description of Toolbar Buttons

Icon	Button Name	Description
✓	**Submit**	Applies parameter settings to the device (see "Saving Configuration" on page 592). **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 592).
Device Actions ▼	**Device Actions**	Opens a drop-down list with frequently needed commands: ▪ **Load Configuration File:** Opens the Configuration File page for loading an *ini* file to the device (see "Backing Up and Loading Configuration File" on page 629). ▪ **Save Configuration File:** Opens the Configuration File page for saving the *ini* file to a folder on your PC (see "Backing Up and Loading Configuration File" on page 629). ▪ **Reset:** Opens the Maintenance Actions page for performing various maintenance procedures such as resetting the device (see "Resetting the Device" on page 589). ▪ **Software Upgrade Wizard:** Starts the Software Upgrade Wizard for upgrading the device's software (see "Software Upgrade Wizard" on page 625).
⌂	**Home**	Opens the Home page (see "Viewing the Home Page" on page 60).
?	**Help**	Opens the Online Help topic of the currently opened configuration page (see "Getting Help" on page 58).
🔑	**Log off**	Logs off a session with the Web interface (see "Logging Off the Web Interface" on page 59).
-	**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.

8.1.5 Navigation Tree

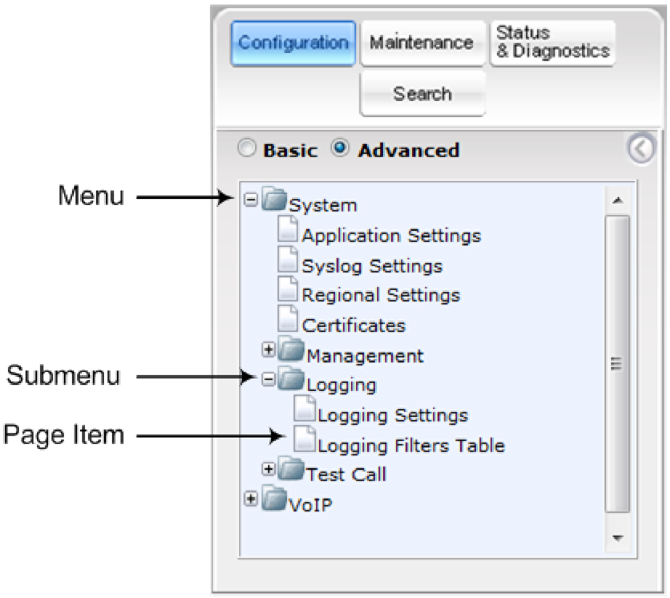
The Navigation tree is located in the Navigation pane and displays a tree-like structure of menus pertaining to the selected tab on the Navigation bar. You can drill-down to the required page item level to open its corresponding page in the Work pane.

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 8-3: 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.

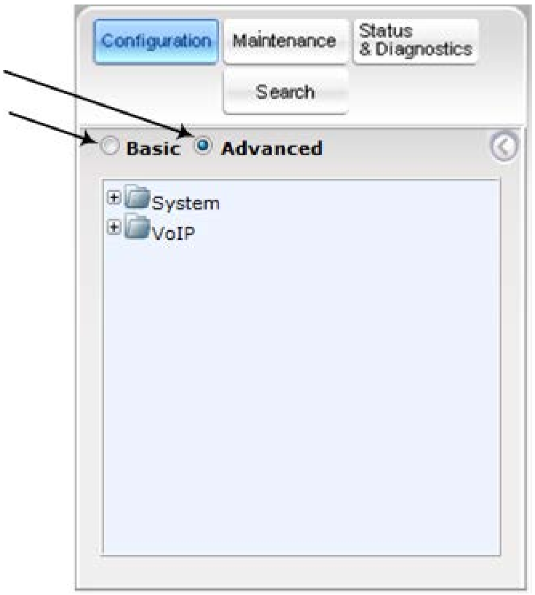
8.1.5.1 Displaying Navigation Tree in Basic and Full View

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 8-4: Basic and Full View Options



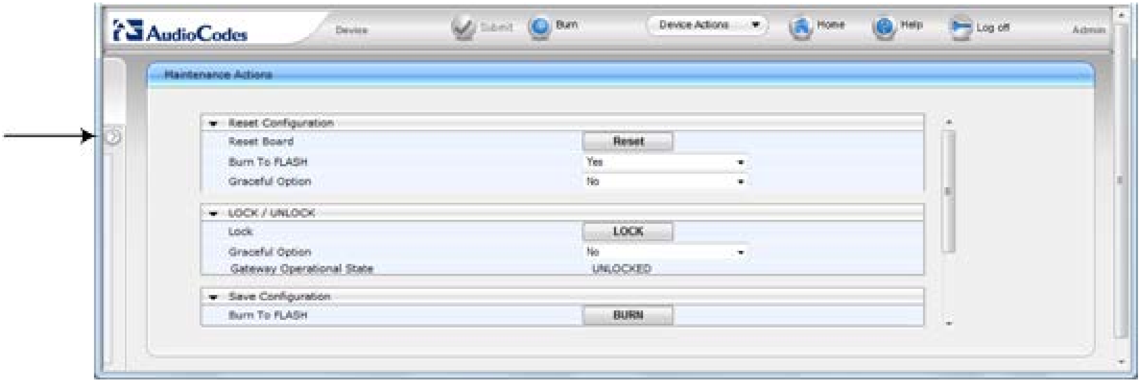
Note: After you reset the device, the Web GUI is displayed in **Basic** view.

8.1.5.2 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 8-5: Show and Hide Button (Navigation Pane in Hide View)



Reset Configuration
- Reset Board: Reset
- Burn To FLASH: Yes
- Graceful Option: No

LOCK / UNLOCK
- Lock: LOCK
- Graceful Option: No
- Gateway Operational State: UNLOCKED

Save Configuration
- Burn To FLASH: BURN

8.1.6 Working with Configuration Pages

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

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

8.1.6.2 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 51
- Displaying parameter groups - see "Showing / Hiding Parameter Groups" on page 52

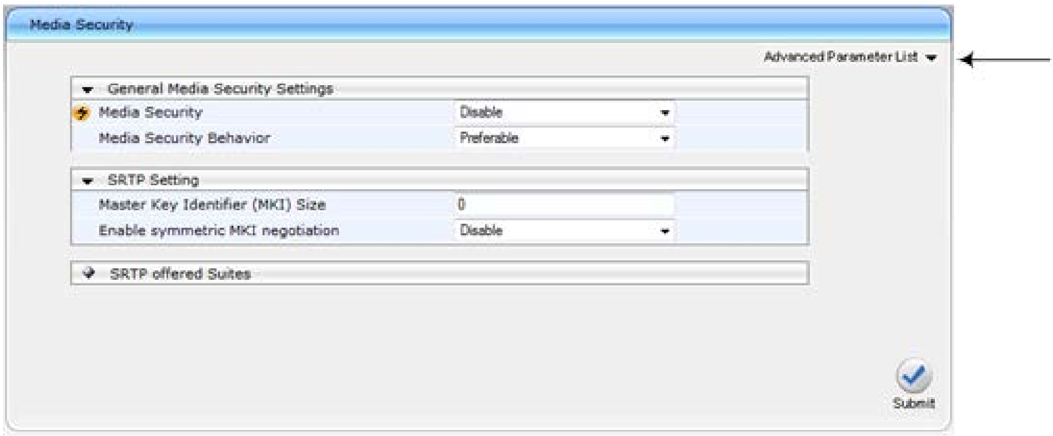
8.1.6.2.1 Displaying Basic and Advanced Parameters

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 8-6: Toggling between Basic and Advanced View



Media Security

Advanced Parameter List

General Media Security Settings

Media Security: Disable

Media Security Behavior: Preferable

SRTP Setting

Master Key Identifier (MKI) Size: 0

Enable symmetric MKI negotiation: Disable

SRTP offered Suites

Submit

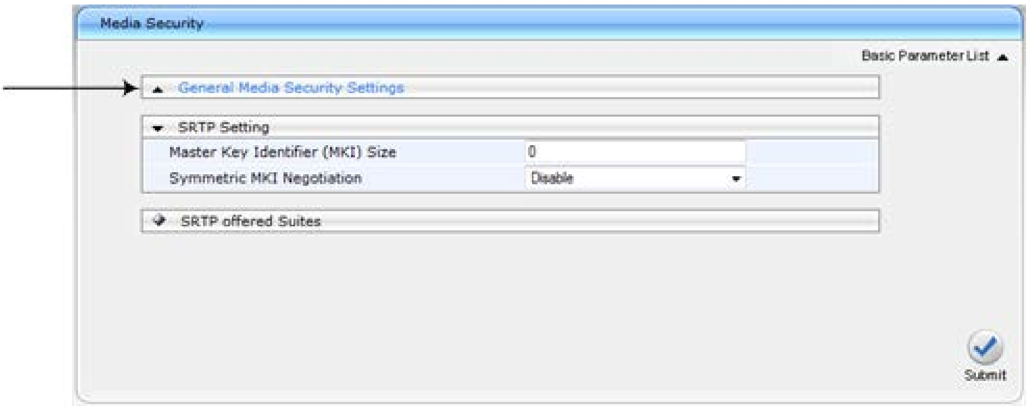
Notes:

- When the Navigation tree is in **Advanced** display mode (see "Navigation Tree" on page 48), 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.

8.1.6.2.2 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 8-7: Expanding and Collapsing Parameter Groups



Media Security

Basic Parameter List

General Media Security Settings

SRTP Setting

Master Key Identifier (MKI) Size: 0

Symmetric MKI Negotiation: Disable

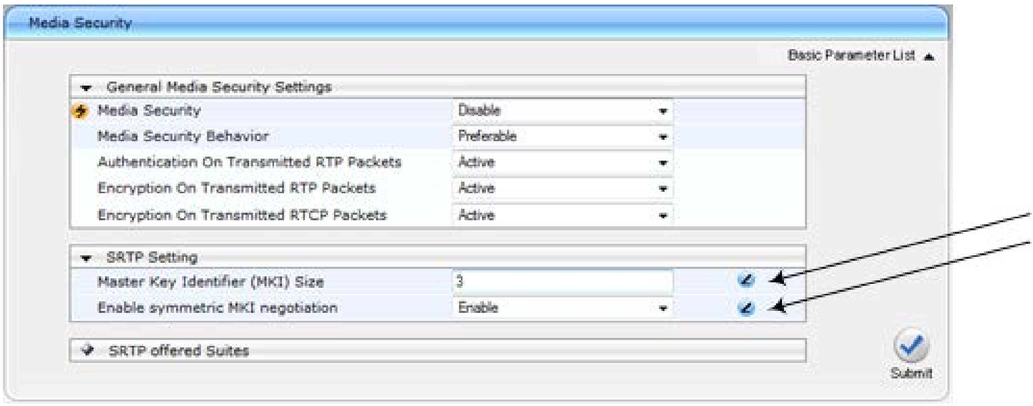
SRTP offered Suites

Submit

8.1.6.3 Modifying and Saving Parameters

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

Figure 8-8: Edit Symbol after Modifying Parameter Value



Media Security

Basic Parameter List ▲

▾ General Media Security Settings

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

▾ SRTP Setting

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

▸ SRTP offered Suites

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

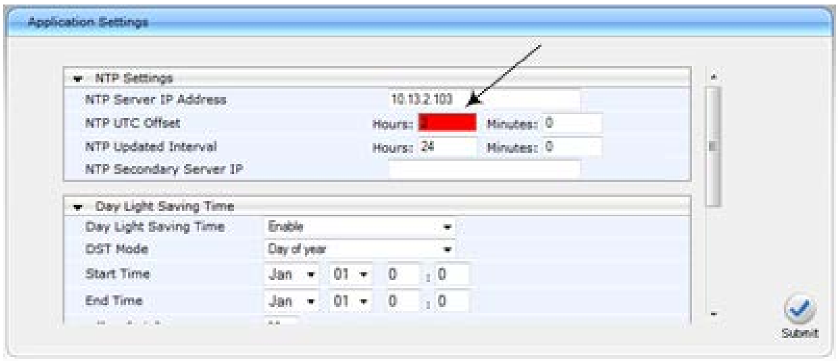
- On the toolbar, click the **Submit** ✓ button.
- At the bottom of the page, click the **Submit** ✓ button.

When you click **Submit**, modifications to parameters with on-the-fly capabilities are immediately applied to the device and take effect. Parameters displayed on the page with the lightning ⚡ icon take effect only after a device reset. For resetting the device, see "Resetting the Device" on page 589.

> **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 592).

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

Figure 8-9: Value Reverts to Previous Valid Value



Application Settings

NTP Settings

NTP Server IP Address: 10.13.2.103
NTP UTC Offset: Hours: [highlighted in red] Minutes: 0
NTP Updated Interval: Hours: 24 Minutes: 0
NTP Secondary Server IP:

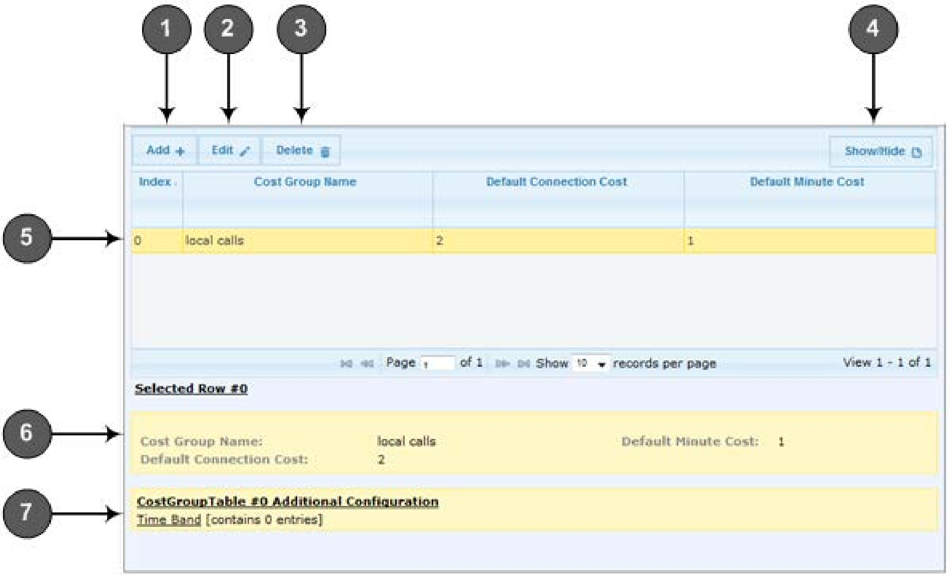
Day Light Saving Time

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

8.1.6.4 Working with Tables

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

Figure 8-10: Displayed Details Pane



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

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

Selected Row #0

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

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

Table 8-3: Enhanced Table Design Description

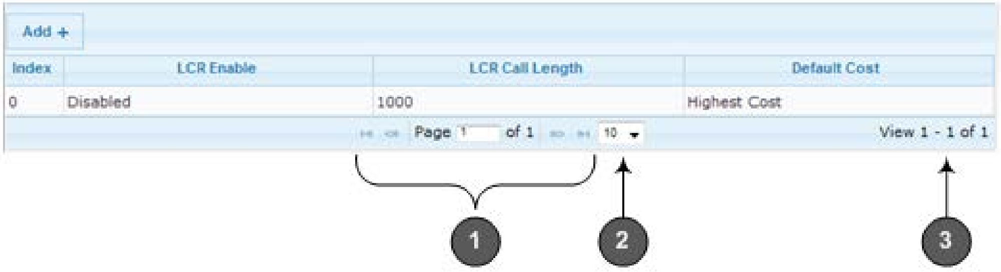
Item #	Button	
1	**Add**	Adds a new index entry row to the table. When you click this button, a dialog box appears with parameters for configuring the new entry. When you have completed configuration, click the **Submit** button in the dialog box to add it to the table.
2	**Edit**	Edits the selected row.
3	**Delete**	Removes the selected row from the table. When you click this button, a confirmation box appears requesting you to confirm deletion. Click **Delete** to accept deletion.

Item #	Button	
4	**Show/Hide**	Toggles between displaying and hiding the full configuration of a selected row. This configuration is displayed below the table (see Item #6) and is useful for large tables that cannot display all its columns in the work pane.
5	-	Selected index row entry for editing, deleting and showing configuration.
6	-	Displays the full configuration of the selected row when you click the **Show/Hide** button.
7	-	Links to access additional configuration tables related to the current configuration.

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

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

Figure 8-11: Viewing Table Rows per Page



Add +

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

Page 1 of 1 10 View 1 - 1 of 1

Table 8-4: Row Display and Page Navigation

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

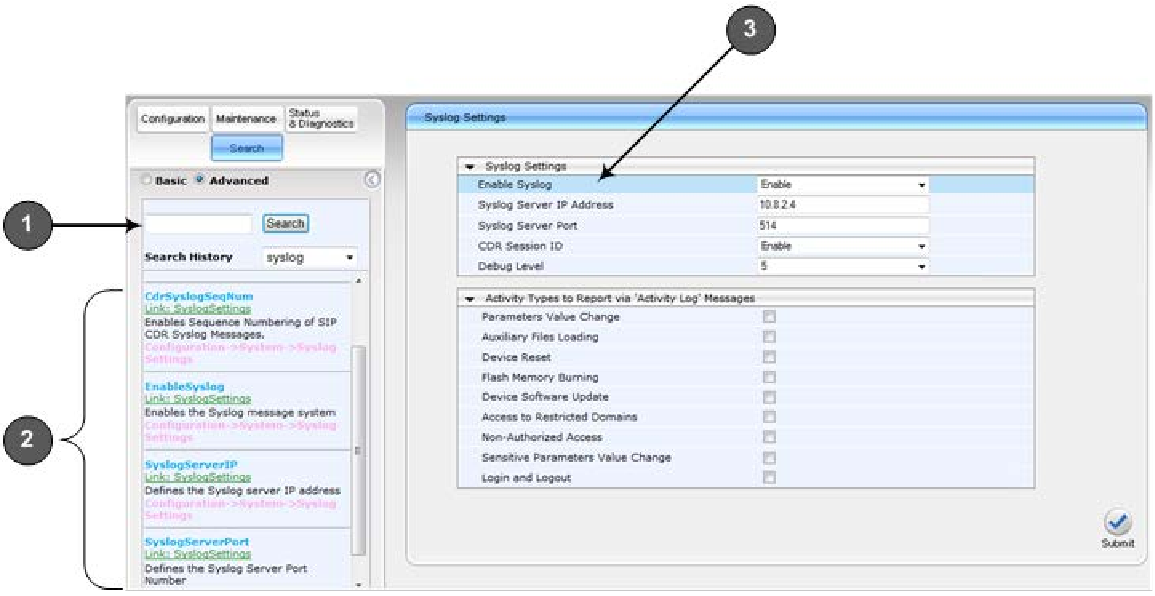
8.1.7 Searching for Configuration Parameters

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

➢ To search for a parameter:

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

Figure 8-12: Searched Result Screen



Search History: syslog

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

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

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

SyslogServerPort
Link: SyslogSettings
Defines the Syslog Server Port Number

Syslog Settings

▾ Syslog Settings

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

▾ Activity Types to Report via 'Activity Log' Messages

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

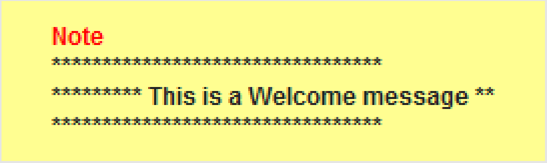
Table 8-5: Search Description

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

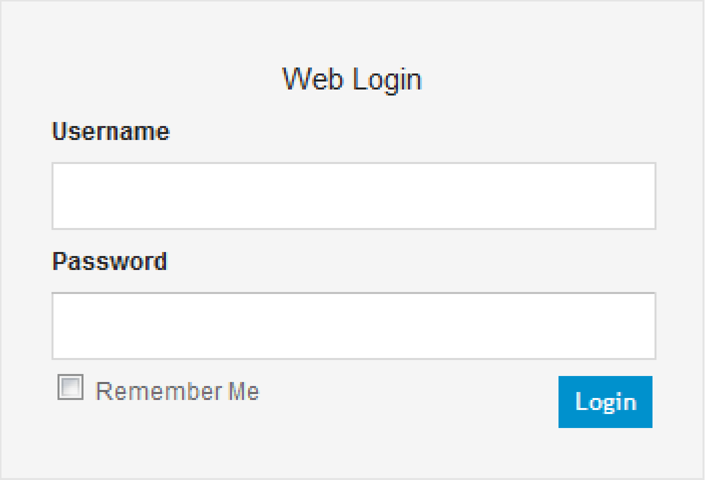
8.1.8 Creating a Login Welcome Message

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

Figure 8-13: User-Defined Web Welcome Message after Login



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



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

Table 8-6: *ini* File Parameter for Welcome Login Message

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

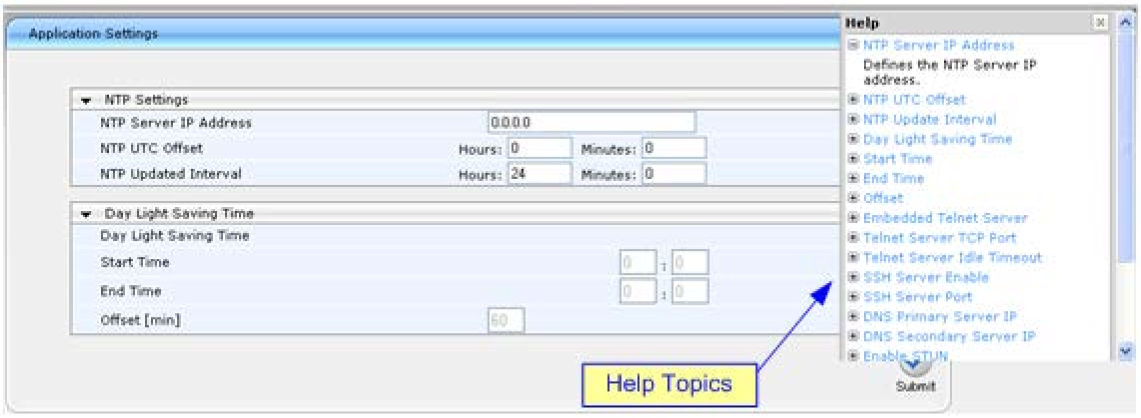
8.1.9 Getting Help

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

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

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

Figure 8-14: Help Topic for Current Page



Application Settings

▾ NTP Settings

NTP Server IP Address: 0.0.0.0

NTP UTC Offset: Hours: 0 Minutes: 0

NTP Updated Interval: Hours: 24 Minutes: 0

▾ Day Light Saving Time

Day Light Saving Time

Start Time: 0 : 0

End Time: 0 : 0

Offset [min]: 60

Help

⊟ NTP Server IP Address

Defines the NTP Server IP address.

⊞ NTP UTC Offset
⊞ NTP Update Interval
⊞ Day Light Saving Time
⊞ Start Time
⊞ End Time
⊞ Offset
⊞ Embedded Telnet Server
⊞ Telnet Server TCP Port
⊞ Telnet Server Idle Timeout
⊞ SSH Server Enable
⊞ SSH Server Port
⊞ DNS Primary Server IP
⊞ DNS Secondary Server IP
⊞ Enable STUN

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.

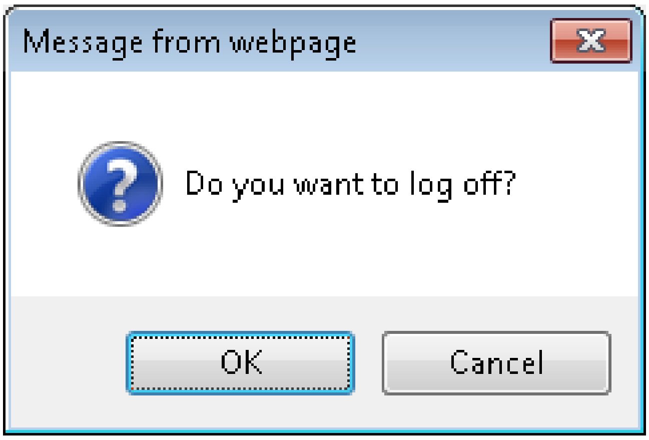
8.1.10 Logging Off the Web Interface

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

➢ **To log off the Web interface:**

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

Figure 8-15: Log Off Confirmation Box



Message from webpage

Do you want to log off?

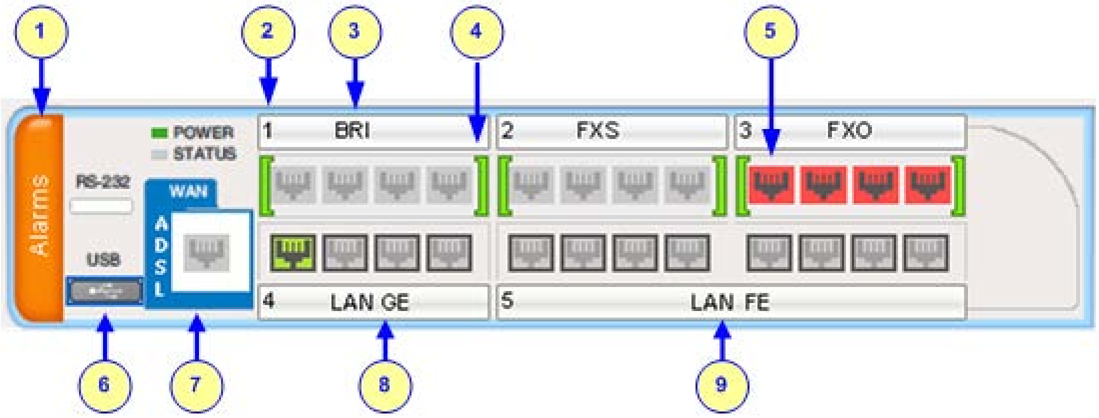
2. Click **OK**; you are logged off the Web session and the Web Login dialog box appears enabling you to re-login, if required.

8.2 Viewing the Home Page

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

➢ **To access the Home page:**

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



Note: The displayed number and type of telephony interfaces, LAN interfaces and WAN interfaces depends on the ordered hardware configuration.

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:

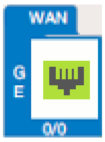
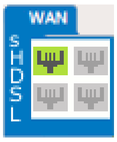
- **IP Address:** IP address of the device
- **Subnet Mask:** Subnet mask address of the device
- **Default Gateway Address:** Default gateway used by the device
- Digital Port Number: Number of digital PRI ports (depending on ordered hardware configuration)
- BRI Port Number: Number of BRI ports (depending on ordered hardware configuration))
- Analog Port Number: Number of analog (FXS and FXO) ports (depending on ordered hardware configuration)
- **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 589.

The table below describes the areas of the Home page.

Table 8-7: Home Page Description

<table>
<tr><th>Item #</th><th colspan="3">Description</th></tr>
<tr><td>1</td><td colspan="3">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 665).</td></tr>
<tr><td>2</td><td colspan="3">Module slot number.</td></tr>
<tr><td>3</td><td colspan="3">Module interface type (e.g., FXS, FXO, and DIGITAL).</td></tr>
<tr><td>4</td><td colspan="3">Module status icon:
▪ [(green): Module has been inserted or is correctly configured
▪ [(gray): Module was removed and "Reserved" is displayed
▪ [(red): Module failure and "Failure" is displayed</td></tr>
<tr><td>5</td><td colspan="3">Port (trunk or channel) status icon.</td></tr>
<tr><td></td><td>Icon</td><td>Trunk Description
(Digital Module)</td><td>Channel Description
(Analog Modules)</td></tr>
<tr><td></td><td>⬜
(gray)</td><td>Disable: Trunk not configured (not in use)</td><td>Idle: Channel is currently on-hook</td></tr>
<tr><td></td><td>🟩
(green)</td><td>Active - OK: Trunk synchronized</td><td>Call Connected: Active RTP stream</td></tr>
<tr><td></td><td>🟨
(yellow)</td><td>RAI Alarm: Remote Alarm Indication (RAI), also known as the Yellow Alarm</td><td>-</td></tr>
<tr><td></td><td>🟥
(red)</td><td>LOS/LOF Alarm: Loss due to LOS (Loss of Signal) or LOF (Loss of Frame)</td><td>Not Connected: No FXO line is connected to this port or port out of service due to Serial Peripheral Interface (SPI) failure (applicable only to FXO interfaces)</td></tr>
<tr><td></td><td>🟦
(blue)</td><td>AIS Alarm: Alarm Indication Signal (AIS), also known as the Blue Alarm</td><td>Handset Offhook: Channel is off-hook, but there is no active RTP session</td></tr>
<tr><td></td><td>🟧
(orange)</td><td>D-Channel Alarm: D-channel alarm</td><td>-</td></tr>
<tr><td></td><td>🟧
(dark orange)</td><td>NFAS Alarm</td><td>-</td></tr>
</table>

Item #	Description
	If you click a port, a shortcut menu appears with commands allowing you to do the following: ▪ Reset channel (Analog ports only): Resets the analog port (see Resetting an Analog Channel on page 595) ▪ Port Settings: Displays trunk status (see "Viewing Trunk and Channel Status" on page 677) and analog port status (see "Viewing Analog Port Information" on page 679) ▪ Update Port Info: Assigns a name to the port (see "Assigning a Port Name" on page 63)
6	USB port for 3G cellular WAN modem for primary or backup WAN: ▪ Gray - USB 3G cellular modem is not configured. ▪ Blue - USB 3G cellular modem is in standby mode (backup mode). ▪ Green - USB 3G cellular modem is active. ▪ Red - USB 3G cellular modem is not active
7	WAN port status icons: ▪ 🟩 (green): Link is working ▪ ⬜ (gray): Link is not configured ▪ 🟥 (red): Link error Depending on ordered hardware configuration, the WAN port can be Gigabit Ethernet copper, SHDSL, or ADSL2+ / VDSL2: Gigabit Ethernet  One Port SHDSL  Four Ports ADSL2+ / VDSL2 One Port
8	Gigabit Ethernet LAN port status icons: ▪ 🟩 (green): Link is working ▪ ⬜ (gray): Link is not configured ▪ 🟥 (red): Link error To view detailed port information, click the port icon (see Viewing Ethernet Port Information on page 662).
9	Fast Ethernet LAN port status icons. See Item 8 for a description.
8 & 9	Power-over-Ethernet status for LAN ports: ▪ ⬜ (gray with dark gray frame): Link is not connected ▪ 🟩 (green with dark gray frame): Ethernet delivered ▪ 🟧 (gray with orange frame): Power delivered ▪ 🟧 (green with orange frame): Ethernet and power delivered

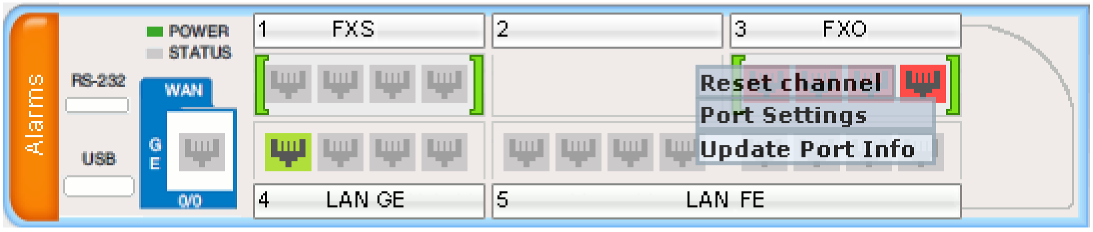
8.2.1 Assigning a Port Name

You can configure an arbitrary name or a brief description for each telephony port displayed on the Home page. This description is displayed as a tooltip when you hover your mouse over the port.

> **Note:** Only alphanumerical characters can be used in the port description.

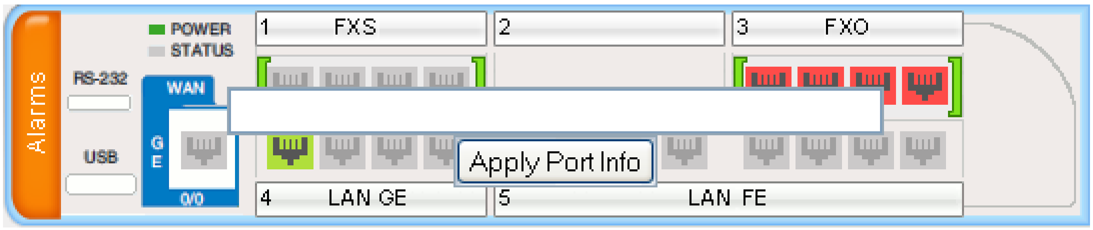
➢ **To add a port description:**

1. Open the Home page.
2. Click the required port icon; a shortcut menu appears:



3. From the shortcut menu, choose **Update Port Info**; a text box appears:

Figure 8-16: Text Box for Entering Port Name



4. Type a brief description for the port, and then click **Apply Port Info**.

8.3 Configuring Web User Accounts

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

Each user account is based on the following:

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

Table 8-8: Web User Access Levels and Privileges

User Level	Numeric Representation in RADIUS	Privileges
Security Administrator	200	Read / write privileges for all pages. It can create all user types and is the only one that can create the first Master user. **Note:** At least one Security Administrator user must exits.
Master	220	Read / write privileges for all pages. Can create all user types, including additional Master users and Security Administrators. It can delete all users except the last Security Administrator.
Administrator	100	Read / write privileges for all pages, except security-related pages (read-only).
Monitor	50	No access to security-related and file-loading pages; read-only access to all other pages.
No Access	0	No access to any page. **Note:** This access level is not applicable when using advanced Web user account configuration in the Web Users table.

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

Table 8-9: Pre-configured Web User Accounts

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

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

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

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

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

Notes:

- For security, it's recommended that you change the default username and password of the pre-configured users (i.e., Security Administrator and Monitor users).
- The Security Administrator user can change all attributes of all Web user accounts. Web users with access levels other than Security Administrator can change only their 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), by using the *ini* file parameter DisableWebConfig (see "Web and Telnet Parameters" on page 763).
- You can define additional Web user accounts using a RADIUS server (see "RADIUS Authentication" on page 206).

8.3.1 Basic User Accounts Configuration

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

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

Table 8-10: Allowed Modifications per Web User Level

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

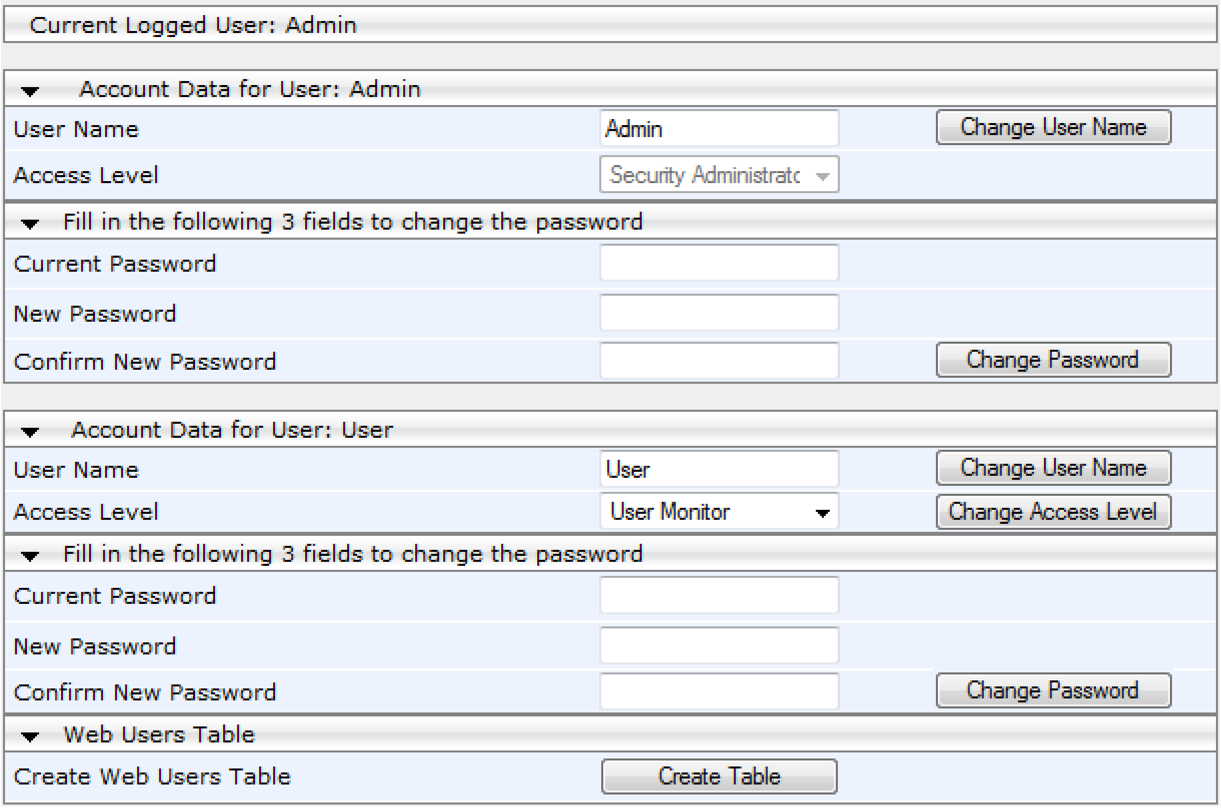
Notes:

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

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

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

Figure 8-17: WEB User Accounts Page (for Users with 'Security Administrator' Privileges)



Current Logged User: Admin

▼ Account Data for User: Admin

User Name	Admin	Change User Name
Access Level	Security Administratc	

▼ Fill in the following 3 fields to change the password

Current Password		
New Password		
Confirm New Password		Change Password

▼ Account Data for User: User

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

▼ Fill in the following 3 fields to change the password

Current Password		
New Password		
Confirm New Password		Change Password

▼ Web Users Table

Create Web Users Table	Create Table

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

8.3.2 Advanced User Accounts Configuration

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

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

Notes:

- Only the Security Administrator user can **initially** access the Web Users table. 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 71).
- 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 71). The new password must have at least four different characters than the previous password. (The Security Administrator users and Master users can change their passwords in the Web Users table and in the Web Security Settings page.)

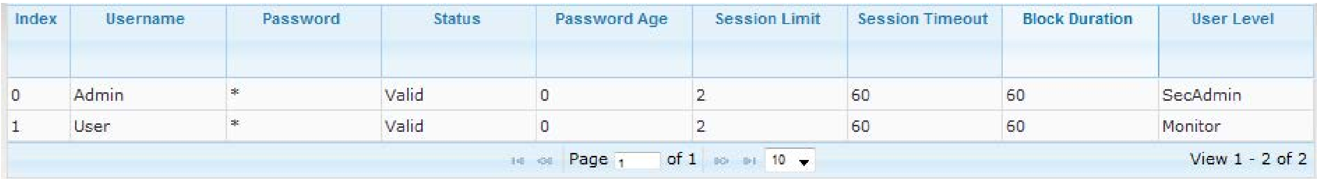
The following procedure describes how to configure Web users in the Web interface. You can also configure this using the CLI command web-users.

➢ **To add Web user accounts with advanced settings:**

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

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

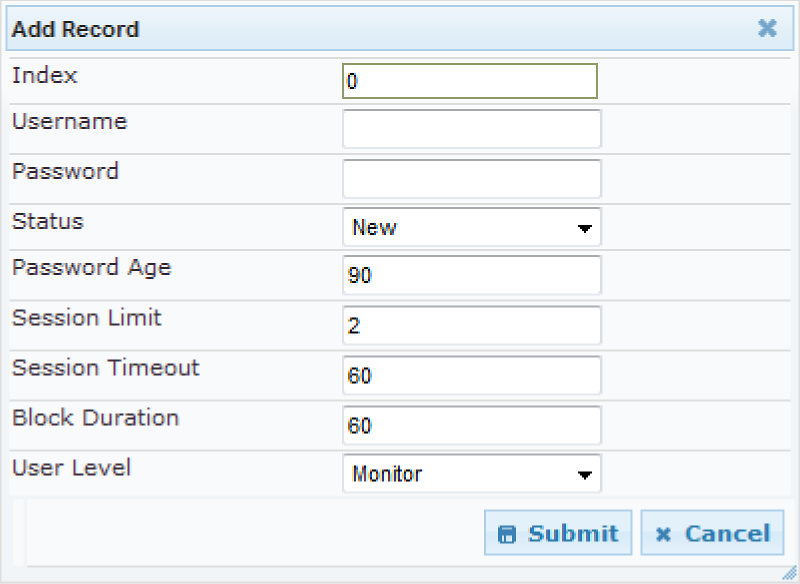
Figure 8-18: Web Users Table Page



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

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

Figure 8-19: Web Users Table - Add Record Dialog Box



Add Record

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

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

Table 8-11: Web User Table Parameter Descriptions

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

Parameter	Description
Web: Status CLI: status	Defines the status of the Web user. ▪ New = (Default) User is required to change its password on the next login. When the user logs in to the Web interface, the user is immediately prompted to change the current password. ▪ Valid = User can log in to the Web interface as normal. ▪ Failed Access = This state is automatically set for users that exceed a user-defined number of failed login attempts, set by the 'Deny Access on Fail Count' parameter (see "Configuring Web Security Settings" on page 71). These users can log in only after a user-defined timeout configured by the 'Block Duration' parameter (see below) or if their status is changed (to New or Valid) by a System Administrator or Master. ▪ Old Account = This state is automatically set for users that have not accessed the Web interface for a user-defined number of days, set by the 'User Inactivity Timer' (see "Configuring Web Security Settings" on page 71). These users can only log in to the Web interface if their status is changed (to New or Valid) by a System Administrator or Master. **Notes:** ▪ The Old Account status is applicable only to Admin and Monitor users; System Administrator and Master users can be inactive indefinitely. ▪ For security, it is recommended to set the status of a newly added user to New in order to enforce password change.
Web: Password Age CLI: pw-age-interval	Defines the duration (in days) of the validity of the password. When this duration elapses, the user is prompted to change the password; otherwise, access to the Web interface is blocked. The valid value is 0 to 10000, where 0 means that the password is always valid. The default is 90.
Web: Session Limit CLI: session-limit	Defines the maximum number of Web interface sessions allowed for the user. In other words, this allows the same user account to log in to the device from different sources (i.e., IP addresses). The valid value is 0 to 5. The default is 2. **Note:** Up to 5 users can be concurrently logged in to the Web interface.
Web: Session Timeout CLI: session-timeout	Defines the duration (in minutes) of Web inactivity of a logged-in user, after which the user is automatically logged off the Web interface. The valid value is 0 to 100000. The default value is according to the settings of the 'Session Timeout' global parameter (see "Configuring Web Security Settings" on page 71).
Web: Block Duration CLI: block-time	Defines the duration (in seconds) for which the user is blocked when the user exceeds a user-defined number of failed login attempts. This is configured by the 'Deny Access On Fail Count' parameter (see "Configuring Web Security Settings" on page 71). The valid value is 0 to 100000, where 0 means that the user can do as many login failures without getting blocked. The default is according to the settings of the 'Deny Authentication Timer' parameter (see "Configuring Web Security Settings" on page 71). **Note:** The 'Deny Authentication Timer' parameter relates to failed Web logins from specific IP addresses.

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

8.4 Displaying Login Information upon Login

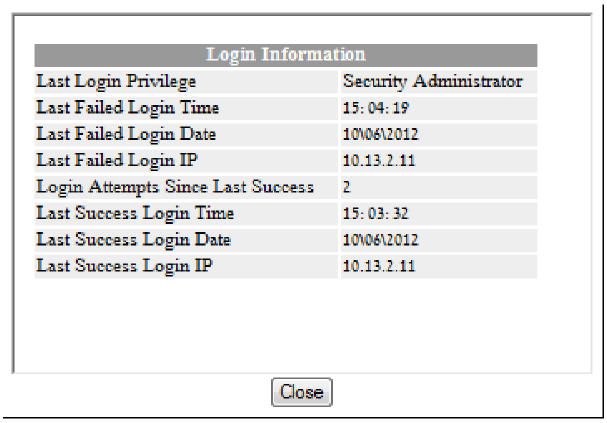
The device can display login information immediately upon Web login.

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

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

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

Figure 8-20: 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

8.5 Configuring Web Security Settings

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

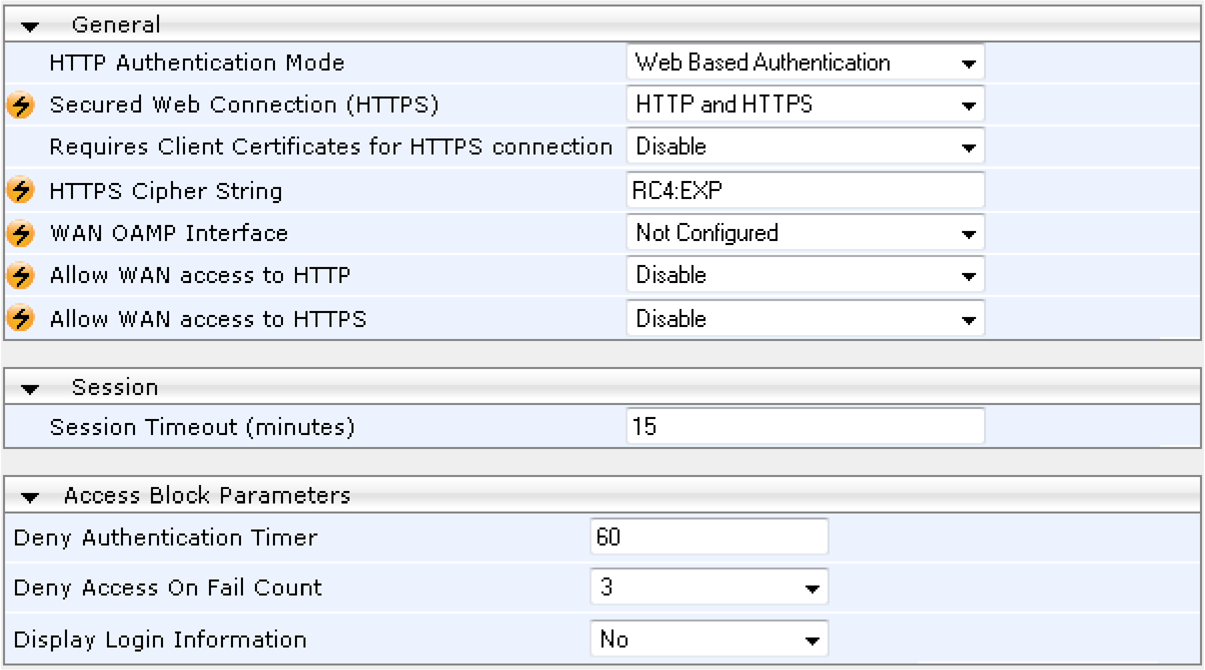
By default, the device accepts HTTP and HTTPS access. However, you can enforce secure Web access communication method by configuring the device to accept only HTTPS.

For a description of these parameters, see "Web and Telnet Parameters" on page 763.

➢ **To define Web access security:**

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

Figure 8-21: Web Security Settings Page



▼ General

HTTP Authentication Mode	Web Based Authentication
⚡ Secured Web Connection (HTTPS)	HTTP and HTTPS
Requires Client Certificates for HTTPS connection	Disable
⚡ HTTPS Cipher String	RC4:EXP
⚡ WAN OAMP Interface	Not Configured
⚡ Allow WAN access to HTTP	Disable
⚡ Allow WAN access to HTTPS	Disable

▼ Session

Session Timeout (minutes)	15

▼ Access Block Parameters

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

2. Set the 'Secured Web Connection (HTTPS)' parameter to **HTTPS Only**.
3. Configure the parameters as required.
4. Click **Submit**.
5. To save the changes to flash memory, see "Saving Configuration" on page 592.

8.6 Limiting OAMP Access to a Specific WAN Interface

You can limit the access of OAMP applications (such as HTTP, HTTPS, Telnet, and SSH) to a specific WAN interface. This OAMP-interface binding can then be associated with a Virtual Routing and Forwarding (VRF).

➢ **To limit OAMP access on a specific WAN interface, using CLI.**

1. Enable WAN management access for specific OAMP applications, using any of the following commands:

```
(config-system)# cli-terminal
(cli-terminal)# wan-ssh-allow | wan-telnet-allow | wan-snmp-
allow | wan-http-allow | wan-https-allow
```

2. Define the WAN interface for the OAMP applications, using the OAMPWanInterfaceName ini file parameter or the following CLI command:

```
(config-system)# bind interface <interface> <slot/port.vlanId>
oamp
(config-system)# bind vlan <vlanId> oamp
```

The following example enables WAN access for Telnet on interface GigabitEthernet 0/0.4 (GigabitEthernet 0/0.4 may be associated with a VRF):

```
(config-system)# cli-terminal
(cli-terminal)# wan-telnet-allow on
(cli-terminal)# exit
(config-system)# bind interface GigabitEthernet 0/0.5 oamp
```

➢ **To define the WAN OAMP interface using the Web interface:**

1. Open the WEB Security Settings page (see "Configuring Web Security Settings" on page 71).
2. From the 'WAN OAMP Interface' drop-down list, select the required WAN interface.
3. Click **Submit**.

8.7 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.

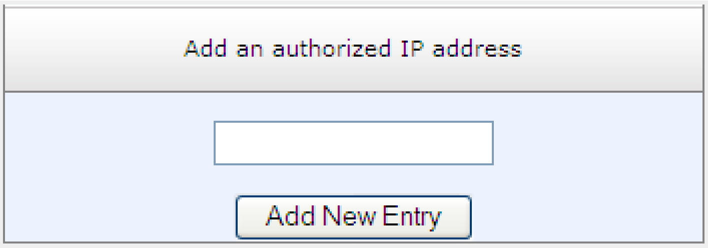
8.8 Configuring Web and Telnet Access List

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 and Telnet Parameters" on page 763).

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

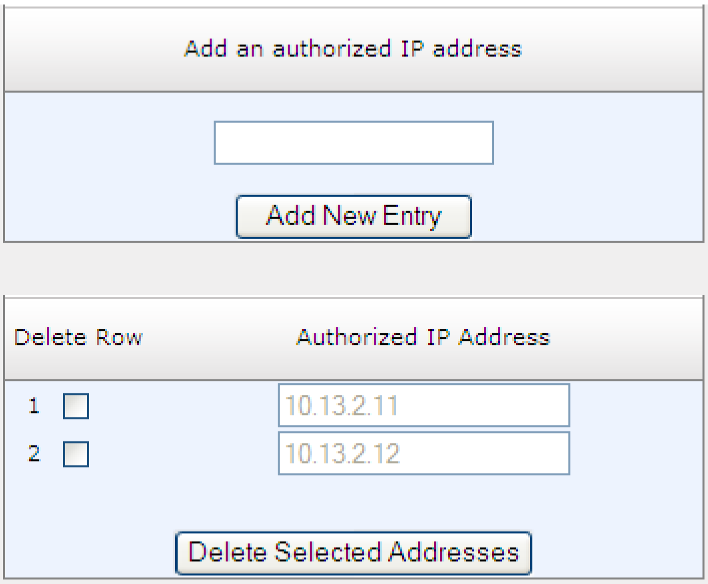
1. Open the Web & Telnet Access List page (**Configuration** tab > **System** menu > **Management** > **Web & Telnet Access List**).

Figure 8-22: 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 8-23: 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 592.

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.

9 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.
- For a description of the CLI commands, refer to the CLI Reference Guide.

9.1 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.

9.1.1 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 "#".

Notes:

- The enable command and subsequent password prompt is required only for users with Administrator or Monitor access levels; Security Administrator and Master access levels automatically enter Enable mode upon initial login. For configuring user access levels, see "Configuring Web User Accounts" on page 64.
- The default password for accessing the Enable mode is "Admin" (case-sensitive). To change this password, use the CLIPrivPass ini file parameter.

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

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

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

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

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

- configure-data: Provides the data-router related configuration commands. This set is accessed by typing the following command:

```
# configure data
(config-data)#
```

9.1.2 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 9-1: 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 ?`

Shortcut Key	Description
	`<cr>`
<Ctrl + A>	Moves the cursor to the beginning of the command line.
<Ctrl + E>	Moves the cursor to the end of the command line.
<Ctrl + U>	Deletes all the characters on the command line.
auto finish	You need only enter enough letters to identify a command as unique. For example, entering "int G 0/0" at the configuration prompt provides you access to the configuration parameters for the specified Gigabit-Ethernet interface. Entering "interface GigabitEthernet 0/0" would work as well, but is not necessary.
Space Bar at the --More--prompt	Displays the next screen of output. You can configure the size of the displayed output, as described in "Configuring Displayed Output Lines in CLI Terminal Window" on page 84.

9.1.3 Common CLI Commands

The following table contains descriptions of common CLI commands.

Table 9-2: 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:
	`(config)# interface vlan 1`
	`(conf-if-VLAN 1)# do show interfaces GigabitEthernet 0/0`

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

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

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

Command	Description

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

9.1.4 Configuring Tables in CLI

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

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

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

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

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

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

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

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

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

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

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

9.1.5 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.

9.2 Enabling CLI

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

9.2.1 Enabling Telnet for CLI

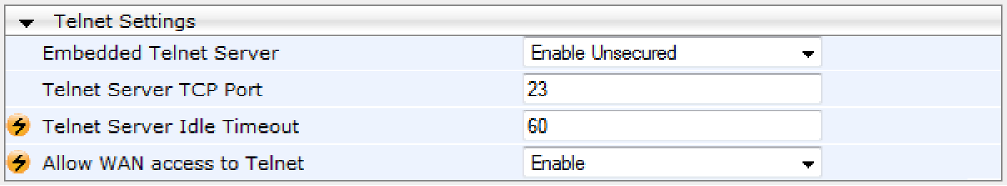
The following procedure describes how to enable Telnet. You can enable a secured Telnet that uses Secure Socket Layer (SSL) where information is not transmitted in the clear. If SSL is used, a special Telnet client is required on your PC to connect to the Telnet interface over a secured connection; examples 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 57).

➢ **To enable Telnet:**

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

Figure 9-1: Telnet Settings on Telnet/SSH Settings Page



Telnet Settings

Embedded Telnet Server	Enable Unsecured
Telnet Server TCP Port	23
Telnet Server Idle Timeout	60
Allow WAN access to Telnet	Enable

2. Set the ‘Embedded Telnet Server’ parameter to **Enable Unsecured** or **Enable Secured** (i.e, SSL).
3. To enable Telnet from the WAN, set the 'Allow WAN access to Telnet' parameter to Enable.
4. Configure the other Tenet parameters as required. For a description of these parameters, see "Telnet Parameters" on page 767.
5. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

9.2.2 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/.

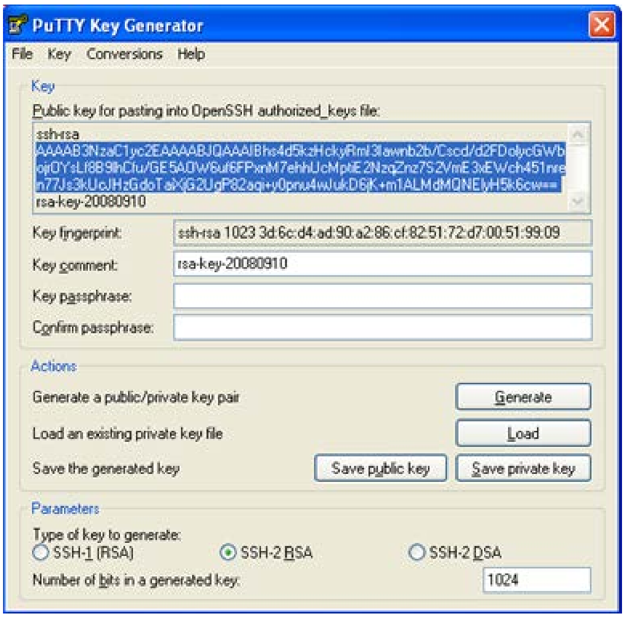
By default, SSH uses the same username and password as the Telnet and Web server. SSH supports 1024/2048-bit RSA public keys, providing carrier-grade security. Follow the instructions below to configure the device with an administrator RSA key as a means of strong authentication.

➢ **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 9-2: Selecting Public RSA Key in PuTTY



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAIBhs4d5kzHckyRml3lawnb2b/Cscd/d2FDolycGWb
ojrOY'sLf8B9lhCfu/GE5A0W6uf6FPxnM7ehhUcMptE2NzqZnz7S2VmE3xEWch451nre
n77Js3kUcJHzGdoTaXjG2UgP82aqi+y0pnu4wJukD6K+m1ALMdMQNElyH5k6cw==
rsa-key-20080910
```

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

Key comment: rsa-key-20080910

Key passphrase:

Confirm passphrase:

Generate a public/private key pair

Load an existing private key file

Save the generated key

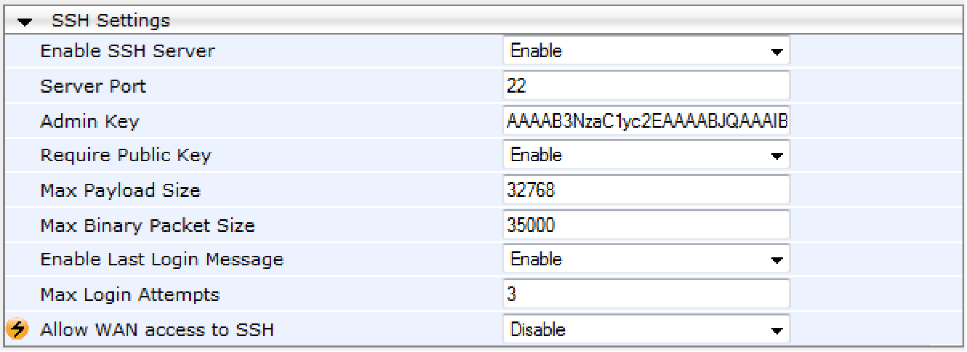
Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024

2. Open the Telnet/SSH Settings page (**Configuration** tab > **System** menu > **Management** > **Telnet/SSH Settings**), and then do the following:
   a. Set the 'Enable SSH Server' parameter to **Enable**.

b. Paste the public key that you copied in Step 1.d into the 'Admin Key' field, as shown below:

Figure 9-3: SSH Settings - Pasting Public RSA Key in 'Admin Key' Field



SSH Settings

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

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. To enable SSH from the WAN, set 'Allow WAN access to SSH' to Enable.

e. Configure the other SSH parameters as required. For a description of these parameters, see "SSH Parameters" on page 802.

f. Click **Submit**.

3. Start the PuTTY Configuration program, and then do the following:

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

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

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

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

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

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

2. Open the admin.key.pub file, and then copy the encoded string from "ssh-rsa" to the white space.

3. Open the Telnet/SSH Settings page (**Configuration** tab > **System** menu > **Management** > **Telnet/SSH Settings**), and then paste the value copied in Step 2 into the 'Admin Key' field.

4. Click **Submit**.

5. Connect to the device with SSH, using the following command:

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

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

9.3 Establishing a CLI Session

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

- **RS-232:** The device can be accessed through its RS-232 serial port, by connecting a VT100 terminal to it or using a terminal emulation program (e.g., HyperTerminal) with a PC. For connecting to the CLI through RS-232, see "CLI" on page 33.
- **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 64.

➢ 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
   ```

9.4 Configuring Maximum Telnet/SSH Sessions

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

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

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

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

9.5 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.

9.6 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 83).

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

9.7 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.

9.8 Configuring TACACS+ for CLI Login

This section describes how to enable and configure Terminal Access Controller Access-Control System (TACACS+). TACACS+ is a security protocol for centralized username and password verification. TACACS+ can be used for validating users attempting to gain access to the device through CLI. TACACS+ services are maintained on a database on a TACACS+ daemon.

You must have access to and must configure a TACACS+ server before configuring TACACS+ on your device.

TACACS+ can provide the following services:

- Authentication: provides authentication through login and password dialog
- Authorization: manages user capabilities for the duration of the user's session by placing restrictions on what commands a user may execute
- Accounting: collects and sends information for auditing and reporting to the TACACS+ daemon

The TACACS+ protocol provides authentication between the device and the TACACS+ daemon, and it ensures confidentiality as all protocol exchanges between a network access server and a TACACS+ daemon are encrypted. You need a system running TACACS+ daemon software to use the TACACS+ functionality on your network access server.

When a user attempts a simple ASCII login by authenticating to a network access server using TACACS+, the following typically occurs:

1. When the connection is established, the network access server contacts the TACACS+ daemon to obtain a username prompt, which is then displayed to the user. The user enters a username and the network access server then contacts the TACACS+ daemon to obtain a password prompt. The network access server displays the password prompt to the user, the user enters a password, and the password is then sent to the TACACS+ daemon.
2. The network access server eventually receives one of the following responses from the TACACS+ daemon:
   - ACCEPT: The user is authenticated and service may begin. If the network access server is configured to require authorization, authorization will begin at this time.
   - REJECT: The user has failed to authenticate. The user may be denied further access.
   - ERROR: An error occurred at some time during authentication. This can be at the daemon or in the network connection between the daemon and the network access server. If an ERROR response is received, the device typically attempts to use an alternative method for authenticating the user.
3. If TACACS+ authorization is needed, the TACACS+ daemon is again contacted for each CLI command entered by the user, and it returns an ACCEPT or REJECT authorization response. If an ACCEPT response is returned, the CLI command is allowed; otherwise, it is rejected.

To configure TACACS+ in the CLI, use the following commands:

- To enable TACACS+:

  ```
  (config-data)# aaa authentication login tacacs+
  ```

- To configure the IP address of the TACACS+ server (up to two servers can be configured):

  ```
  (config-data)# tacacs-server host <IP address>
  ```

- To configure the TCP port number for the TACACS+ service:

  ```
  (config-data)# tacacs-server port <port>
  ```

- To configure the shared secret between the TACACS+ server and the device:

  ```
  (config-data)# tacacs-server key <password>
  ```

- To configure how much time to wait for a TACACS+ response before failing the authentication:

  ```
  (config-data)# tacacs-server timeout <in seconds>
  ```

- To configure the device's data-router WAN interface through which communication with the TACACS+ server is done:

  ```
  (config-data)# tacacs-server source data source-address
  interface <interface name>
  ```

This page is intentionally left blank.

10 SNMP-Based Management

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

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

This section provides configuration relating to SNMP management.

Notes:

- SNMP-based management is enabled by default. For disabling it, see "Enabling SNMP and Configuring SNMP Community Strings" on page 87.
- 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 621.
- For more information on using the EMS tool, refer to the *EMS User's Manual* and *EMS Server IOM Manual*.

10.1 Enabling SNMP and Configuring SNMP Community Strings

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

Notes:

- 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 91).
- You can assign data-router Access Control List rules (ACL) to SNMP community strings. By associating an ACL rule with an SNMP community string, the source and/or destination address of the packet, received from the management station and in which the community string is received, can be specified. This adds enhanced security by reducing the likelihood of malicious attacks on the device if the community string is discovered by an attacker. To assign an ACL rule, use the following CLI command:

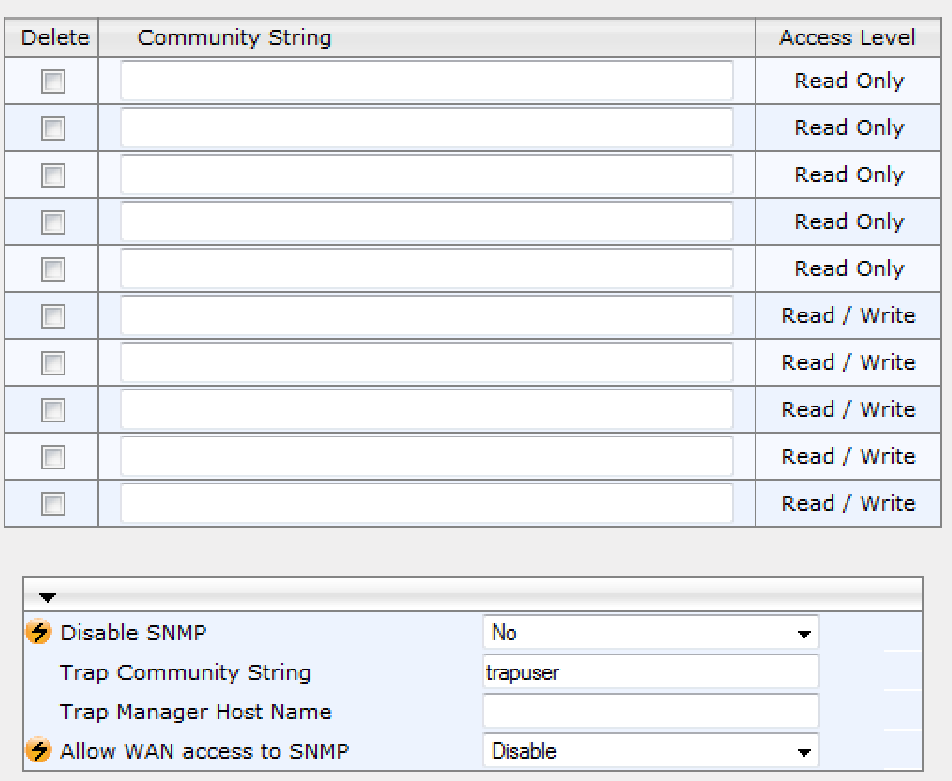
```
(config-system)# snmp
<snmp># snmp-acl community-string <Community string> rw|ro <ACL rule string
name>
```

For detailed descriptions of the SNMP parameters, see "SNMP Parameters" on page 768.

➢ To configure SNMP community strings:

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

Figure 10-1: SNMP Community String Page



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

Disable SNMP	No
Trap Community String	trapuser
Trap Manager Host Name	
Allow WAN access to SNMP	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 10-1: SNMP Community String Parameter Descriptions

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

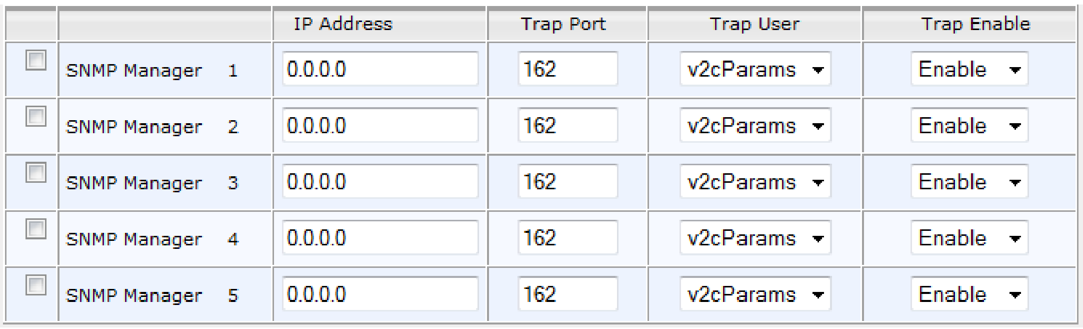
10.2 Configuring SNMP Trap Destinations

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

➢ **To configure SNMP trap destinations:**

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

Figure 10-2: SNMP Trap Destinations Page



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

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

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

Table 10-2: SNMP Trap Destinations Parameters Description

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

Parameter	Description
	encryption level. ▪ v2cParams (default) = SNMPv2 user community string ▪ SNMPv3 user configured in "Configuring SNMP V3 Users" on page 91
Trap Enable **[SNMPManagerTrapSendingEnable_x]**	Activates the sending of traps to the SNMP Manager. ▪ **[0]** Disable ▪ **[1]** Enable (Default)

10.3 Configuring SNMP Trusted Managers

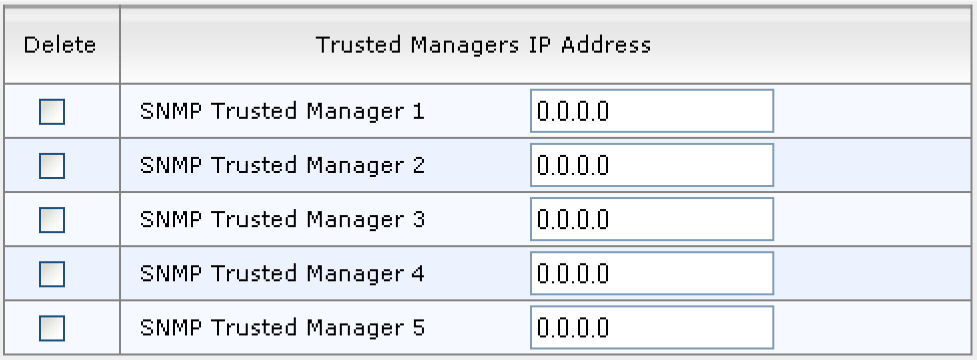
The SNMP Trusted Managers table lets you configure up to five SNMP Trusted Managers based on IP addresses. By default, the SNMP agent accepts SNMP Get and Set requests from any IP address as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers,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

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

➢ **To configure SNMP Trusted Managers:**

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

Figure 10-3: SNMP Trusted Managers



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

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

10.4 Configuring SNMP V3 Users

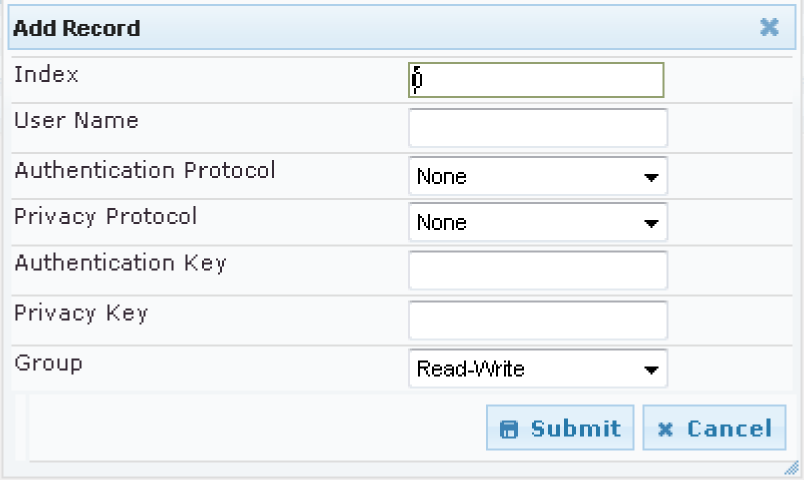
The SNMP v3 Users table lets you configure up to 10 SNMP v3 users for authentication and privacy.

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

➢ **To configure an SNMP v3 user:**

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

Figure 10-4: SNMP V3 Setting Page - Add Record Dialog Box



Add Record

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

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

Note: If you delete a user that is associated with a trap destination (see "Configuring SNMP Trap Destinations" on page 89), the configured trap destination becomes disabled and the trap user reverts to default (i.e., SNMPv2).

Table 10-3: SNMP V3 Users Parameters

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

Parameter	Description
CLI: priv-protocol **[SNMPUsers_PrivProtocol]**	▪ **[0]** None (default) ▪ **[1]** DES ▪ **[2]** 3DES ▪ **[3]** AES-128 ▪ **[4]** AES-192 ▪ **[5]** AES-256
Authentication Key CLI: auth-key **[SNMPUsers_AuthKey]**	Authentication key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Privacy Key CLI: priv-key **[SNMPUsers_PrivKey]**	Privacy key. Keys can be entered in the form of a text password or long hex string. Keys are always persisted as long hex strings and keys are localized.
Group CLI: group **[SNMPUsers_Group]**	The group with which the SNMP v3 user is associated. ▪ **[0]** Read-Only (default) ▪ **[1]** Read-Write ▪ **[2]** Trap **Note:** All groups can be used to send traps.

11 TR-069 Based Management

The device supports TR-069 CPE WAN Management Protocol (CWMP) based management, which is used for remote management of CPE devices. This allows the device to be configured and monitored from a management application running on a remote Auto-Configuration Server (ACS).

11.1 TR-069

TR-069 (Technical Report 069) is a specification published by Broadband Forum (www.broadband-forum.org) entitled CPE WAN Management Protocol (CWMP). It defines an application layer protocol for remote management of end-user devices.

TR-069 uses a bi-directional SOAP/HTTP protocol for communication between the customer premises equipment (CPE) and the Auto Configuration Servers (ACS). The TR-069 connection to the ACS can be done on the LAN or WAN interface.

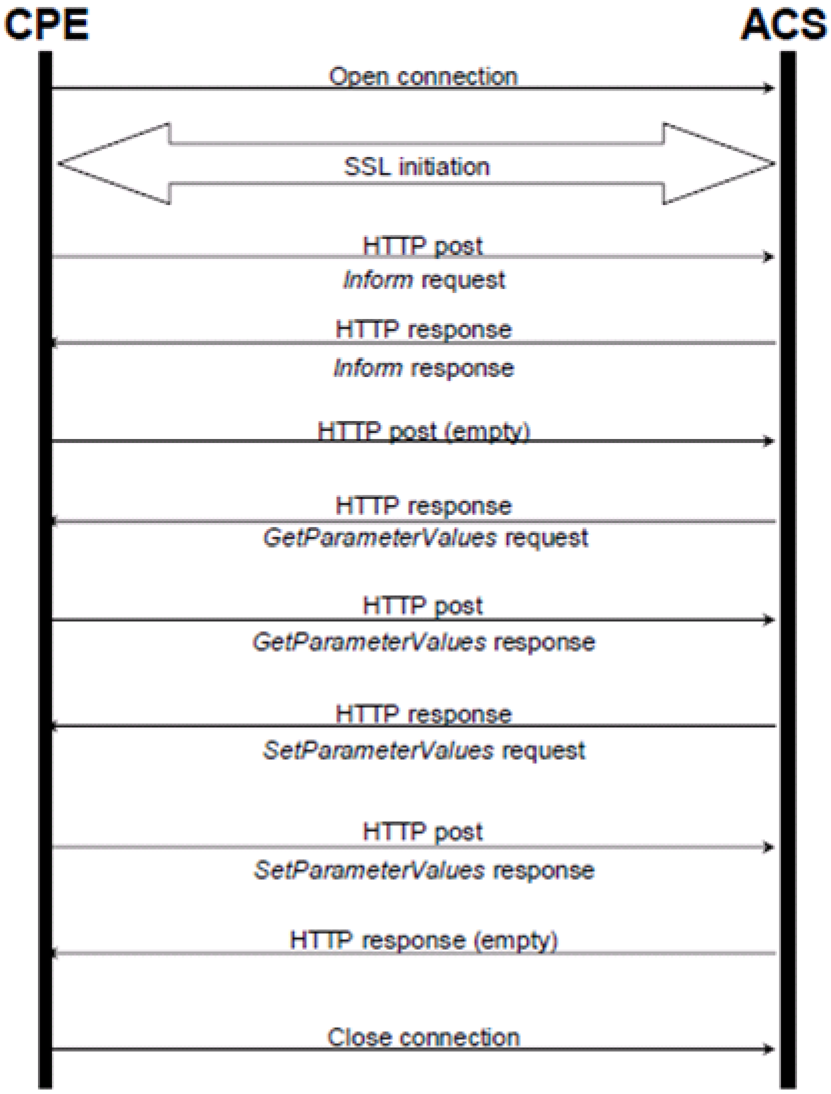
The protocol stack looks as follows:

Table 11-1: TR-069 Protocol Stack

CPE/ACS Management Application
RPC Methods
SOAP
HTTP
SSL/TLS
TCP/IP

Communication is typically established by the CPE; hence, messages from CPE to ACS are typically carried in HTTP requests, and messages from ACS to CPE in HTTP responses.

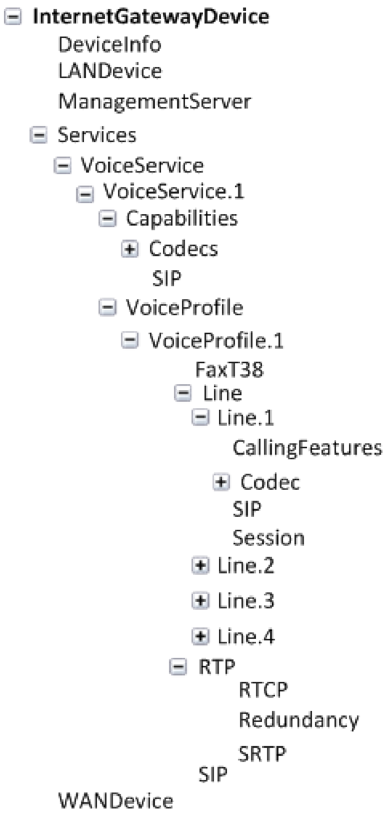
Figure 11-1: TR-069 Session Example



Communication between ACS and CPE is defined via Remote Procedure Call (RPC) methods. TR-069 defines a generic mechanism by which an ACS can read or write parameters to configure a CPE and monitor CPE status and statistics. It also defines the mechanism for file transfer and firmware/software management. However, it does not define individual parameters; these are defined in separate documents, as described below. Some of the RPC methods are Configuration File Download, Firmware upgrade, Get Parameter Value, Set Parameter Value, Reboot, and the upload and download files.

TR-106 defines the “data model” template for TR-069 enabled devices. The Data Model consists of objects and parameters hierarchically organized in a tree with a single Root Object, typically named *Device*. Arrays of objects are supported by appending a numeric index to the object name (e.g. ABCService.1 in the example below); such objects are called “multi-instance objects”.

Figure 11-2: TR-069 Model Data Example



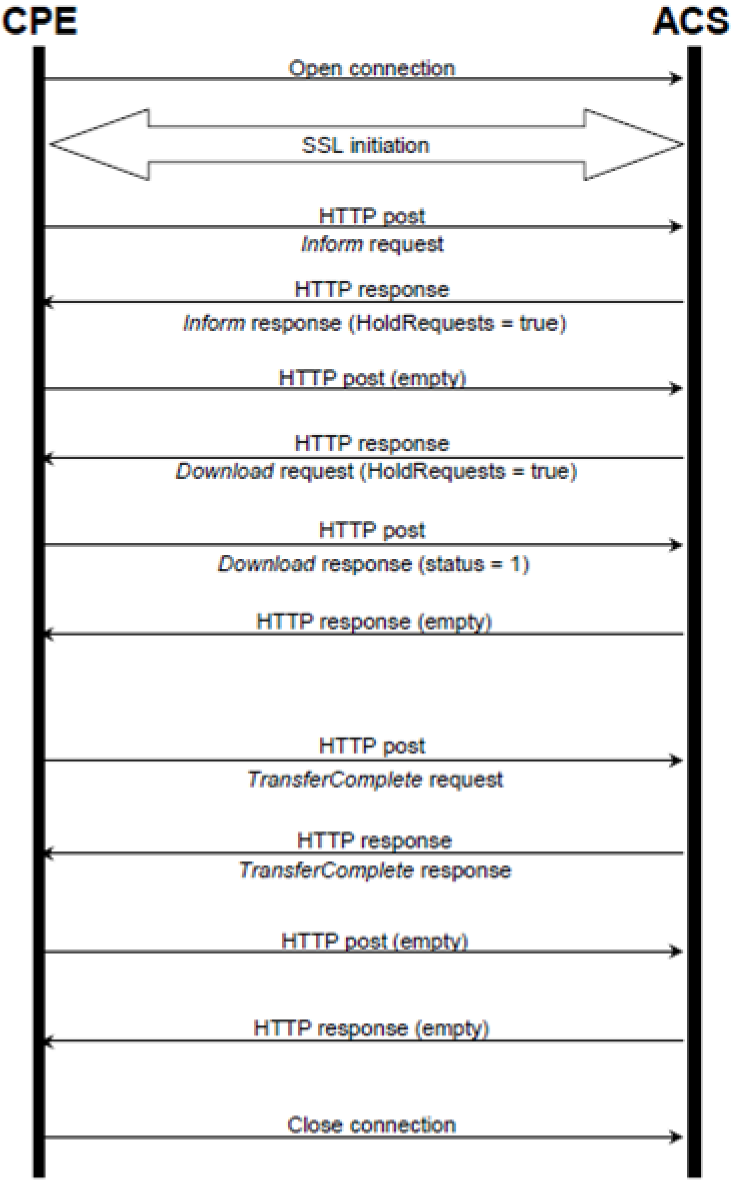
Below is a list of some of the TR-069 methods:

- CPE Methods:
  - GetRPCMethods: Used by the CPE or ACS to discover the set of methods supported by the Server or CPE it is in communication with.
  - SetParameterValues: Used by the ACS to modify the value of CPE parameter(s).
  - GetParameterValues: Used by the ACS to obtain the value of CPE parameter(s).
  - GetParameterNames: Used by the ACS to discover the parameters accessible on a particular CPE.
  - SetParameterAttributes: Used by the ACS to modify attributes associated with CPE parameter(s).
  - GetParameterAttributes: Used by the ACS to read the attributes associated with CPE parameter(s).
  - AddObject: Used by the ACS to create a new instance of a multi-instance object—a collection of parameters and/or other objects for which multiple instances are defined.
  - DeleteObject: Removes a particular instance of an object.
  - Download: Used by the ACS to cause the CPE to download the following file(s) from a designated location:
    - Firmware Upgrade Image (File Type = 1) - cmp file.
    - Vendor Configuration File (File Type = 3) - output of `show running-config` CLI command, which includes Data and Voice configuration.

The CPE responds to the Download method, indicating successful or unsuccessful completion via one of the following:

- A DownloadResponse with the Status argument set to zero (indicating success), or a fault response to the Download request (indicating failure).
- A TransferComplete message sent later in the same session as the Download request (indicating either success or failure). In this case, the Status argument in the corresponding DownloadResponse has a value of one.
- A TransferComplete message sent in a subsequent session (indicating success or failure). In this case, the Status argument in the corresponding DownloadResponse has a value of one.

Figure 11-3: Download Method Executuion Example



- Upload: Used by the ACS to cause the CPE to upload (to the ACS) the following files to a designated location:
  - Vendor Configuration File (File Type = 1 or 3): Output of `show running-config` CLI command, which includes Data and Voice configuration. For File Type 3 (where index is included – see below) only one instance of the file is supported.

        - ♦ Vendor Log File (File Type = 2 or 4): “Aggregated” log file. For File Type 2, the last file is supported. For File Type 4 (where index is included – see below), multiple files is supported.

    The CPE responds to the Upload method, indicating successful or unsuccessful completion via the UploadResponse or TransferComplete method.

    For a complete description of the Upload method, refer to TR-069 Amendment 3 section A.4.1.5.

    - Reboot: Reboots the CPE. The CPE sends the method response and completes the remainder of the session prior to rebooting.
    - X_0090F8_CommandResponse: Runs CLI commands.

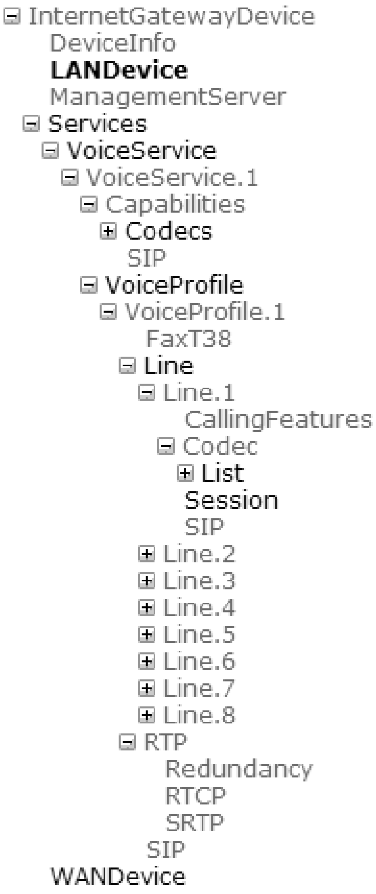
- ACS Methods:
    - Inform: A CPE must call this method to initiate a transaction sequence whenever a connection to an ACS is established.
    - TransferComplete: Informs the ACS of the completion (either successful or unsuccessful) of a file transfer initiated by an earlier Download or Upload method call.

11.2 TR-104

The device supports TR-104 for configuration. This support is for the SIP (VoIP) application layer and applies to FXS interfaces (lines) only. TR-104 defines a "data model" template for TR-069 enabled devices. The "data model" that is applicable to the AudioCodes device is defined in the DSL Forum TR-104 – "DSLHome™ Provisioning Parameters for VoIP CPE" at http://www.broadband-forum.org/technical/download/TR-104.pdf.

The hierarchical tree structure of the supported TR-104 objects is shown below:

Figure 11-4: Hierarchical Tree Structure of TR-104 Objects



```
⊟ InternetGatewayDevice
    DeviceInfo
    LANDevice
    ManagementServer
  ⊟ Services
    ⊟ VoiceService
      ⊟ VoiceService.1
        ⊟ Capabilities
          ⊞ Codecs
            SIP
        ⊟ VoiceProfile
          ⊟ VoiceProfile.1
              FaxT38
            ⊟ Line
              ⊟ Line.1
                  CallingFeatures
                ⊟ Codec
                  ⊞ List
                  Session
                  SIP
              ⊞ Line.2
              ⊞ Line.3
              ⊞ Line.4
              ⊞ Line.5
              ⊞ Line.6
              ⊞ Line.7
              ⊞ Line.8
            ⊟ RTP
                Redundancy
                RTCP
                SRTP
              SIP
    WANDevice
```

- InternetGatewayDevice.Services.VoiceService: Top-level object.
- InternetGatewayDevice.Services.VoiceService.1.Capabilities: (Read-Only) Displays the overall capabilities of the device.
    - InternetGatewayDevice.Services.VoiceService.1.Capabilities.Codecs: (Read-Only) Lists supported codecs (according to devices installed Software Feature Key).
    - InternetGatewayDevice.Services.VoiceService.1.Capabilities.SIP: (Read-Only) Displays various SIP settings such as SIP transport type.
- InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1: Corresponds to one or more FXS lines that share the same basic configuration:
    - InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.FaxT38: Configures fax T.38 relay.
    - InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line: Corresponds to an FXS line (as configured in the Trunk Group table). It enables and configures each FXS line (number).
        - InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.{i}.Code

c.List.{i}: Configures voice coder used by specific FXS line.

    - InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.{i}.CallingFeatures: Configures voice parameters per FXS line such as caller ID.
    - InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.Line.{i}.SIP: Configures username/password per FXS line. AudioCodes maps this object to the corresponding entry in the Authentication table

- InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.SIP: Configures SIP parameters specific to the UA such as Proxy server.
- InternetGatewayDevice.Services.VoiceService.1.VoiceProfile.1.RTP: Configures various RTP parameters for the FXS lines such as RTCP and SRTP.

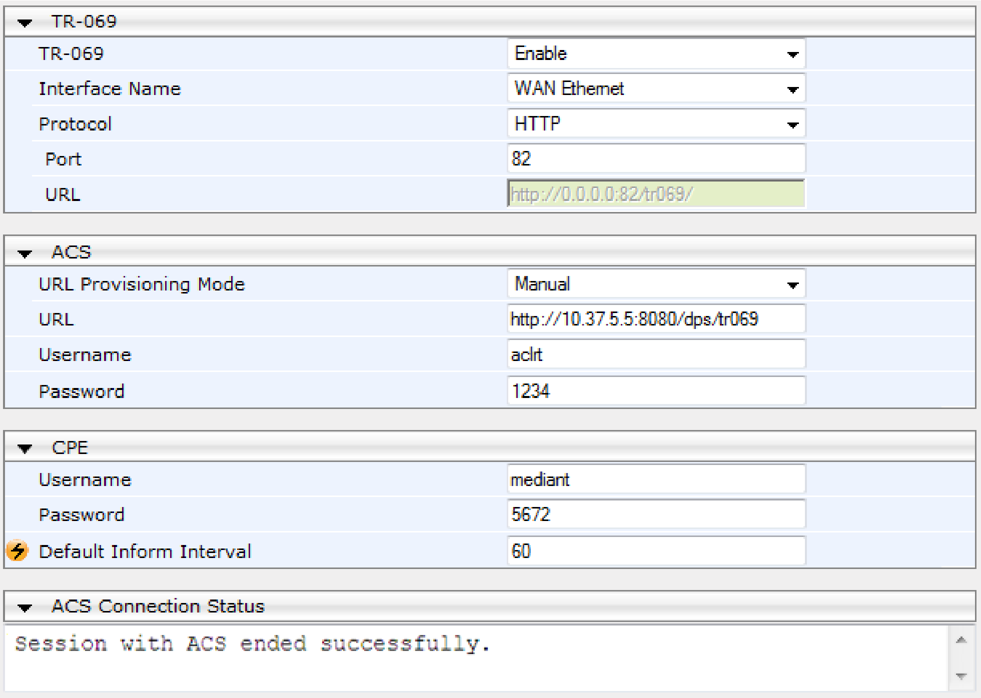
11.3 Configuring TR-069

The CWMP/TR-069 Settings page is used to enable and configure TR-069.

➢ To configure TR-069:

1. Open the CWMP/TR-069 Settings page (**Configuration** tab > **System** menu > **Management** > **CWMP**).

Figure 11-5: CWMP/TR-069 Settings Page



▼ TR-069

TR-069	Enable
Interface Name	WAN Ethernet
Protocol	HTTP
Port	82
URL	http://0.0.0.0:82/tr069/

▼ ACS

URL Provisioning Mode	Manual
URL	http://10.37.5.5:8080/dps/tr069
Username	aclrt
Password	1234

▼ CPE

Username	mediant
Password	5672
⚡ Default Inform Interval	60

▼ ACS Connection Status

```
Session with ACS ended successfully.
```

2. Configure the parameters as required. For a description of the TR-069 parameters, see "TR-069 Parameters" on page 771.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

This page is intentionally left blank.

12 INI File-Based Management

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

Notes:

- For a list and description of the *ini* file parameters, see "Configuration Parameters Reference" on page 763.
- To restore the device to default settings using the *ini* file, see "Restoring Factory Defaults" on page 651.

12.1 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

12.1.1 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.

12.1.2 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 0 = g711Alaw64k, 20, 0, 255, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0;
[ \CodersGroup0 ]
```

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

12.1.3 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.

12.2 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 592).
  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.

Note: To restore the device to default settings using the *ini* file, see "Restoring Factory Defaults" on page 651.

12.3 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>
  - Data-Router Configuration: # copy data-configuration from <URL>
- Web interface:
  - Load Auxiliary Files page (see "Loading Auxiliary Files" on page 599): 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 629): 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*.

12.4 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).

12.5 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>).

12.6 INI Viewer and Editor Utility

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

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

Part III

General System Settings

13 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 121.

13.1 Configuring TLS Certificate Contexts

The TLS Contexts table lets you configure up to 12 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. TLS Contexts are applicable to Gateway and SBC calls.

Each TLS Context can be configured with the following:

- Context ID and name
- TLS version - SSL 2.0 (only for TLS handshake), SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2)
- Encryption ciphers for server and client - DES, RC4 compatible, Advanced Encryption Standard (AES)
- Online Certificate Status Protocol (OCSP). Some Public-Key Infrastructures (PKI) can revoke a certificate after it has been issued. You can configure the device to check whether a peer's certificate has been revoked, using the OCSP. When OCSP is enabled, the device queries the OCSP server for revocation information whenever a peer certificate is received (IPSec, 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 282.
  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 267.
  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.

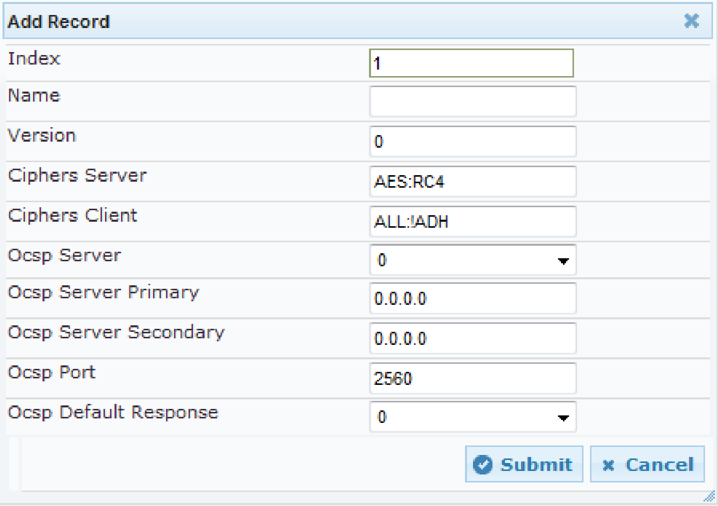
You can also configure TLS certificate expiry check, whereby the device periodically checks the validation date of the installed TLS server certificates and sends an SNMP trap event if a certificate is nearing expiry. This feature is configured globally for all TLS Contexts. For configuring TLS certificate expiry check, see "Configuring TLS Server Certificate Expiry Check" on page 120.

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

➢ To configure a TLS Context:

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

Figure 13-1: TLS Contexts Table - Add Record Dialog Box



Add Record

Index	1
Name	
Version	0
Ciphers Server	AES:RC4
Ciphers Client	ALL:!ADH
Ocsp Server	0
Ocsp Server Primary	0.0.0.0
Ocsp Server Secondary	0.0.0.0
Ocsp Port	2560
Ocsp Default Response	0

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

Table 13-1: TLS Context Parameter Descriptions

Parameter	Description
Web: Index CLI: tls <ID> **[TLSContexts_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Web: Name CLI: name **[TLSContexts_Name]**	Defines an arbitrary name to easily identify the TLS Context. The valid value is a string of up to 31 characters.
Web: Version CLI: tls-version **[TLSContexts_TLSVersion]**	Defines the supported SSL/TLS protocol version. ▪ **[0]** 0 = (Default) SSL 3.0 and all TLS versions (1.0, 1.1, and 1.2) are supported. SSL/TLS handshakes always start with an SSL 2.0-compatible handshake and then switch to the highest TLS version supported by both peers. ▪ **[1]** 1 = Only TLS 1.0 is used. Clients attempting to contact the device using any other version are rejected.

Parameter	Description
Web: Ciphers Server CLI: ciphers-server **[TLSContexts_ServerCipherString]**	Defines the supported cipher suite for the TLS server (in OpenSSL cipher list format). For valid values, refer to URL http://www.openssl.org/docs/apps/ciphers.html. The default is "AES:RC4". 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. **Note**s: ▪ If the installed Software License Key includes the Strong Encryption feature, the default of this parameter is changed to RC4:EXP, enabling RC-128-bit encryption. ▪ The value "ALL" can be used only if the installed Software License Key includes the Strong Encryption feature.
Web: Ciphers Client CLI: ciphers-client **[TLSContexts_ClientCipherString]**	Defines the supported cipher suite for TLS clients. The valid value is up to 255 strings (e.g., "EXP"). The default is "ALL:!ADH". For possible values and additional details, refer to http://www.openssl.org/docs/apps/ciphers.html.
Web: Ocsp Server CLI: ocsp-server **[TLSContexts_OcspEnable]**	Enables or disables certificate checking using OCSP. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Ocsp Server Primary CLI: 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.
Web: Ocsp Server Secondary CLI: 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.
Web: Ocsp Port CLI: ocsp-port **[TLSContexts_OcspServerPort]**	Defines the OCSP server's TCP port number. The default port number is 2560.
Web: Ocsp Default Response CLI: 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

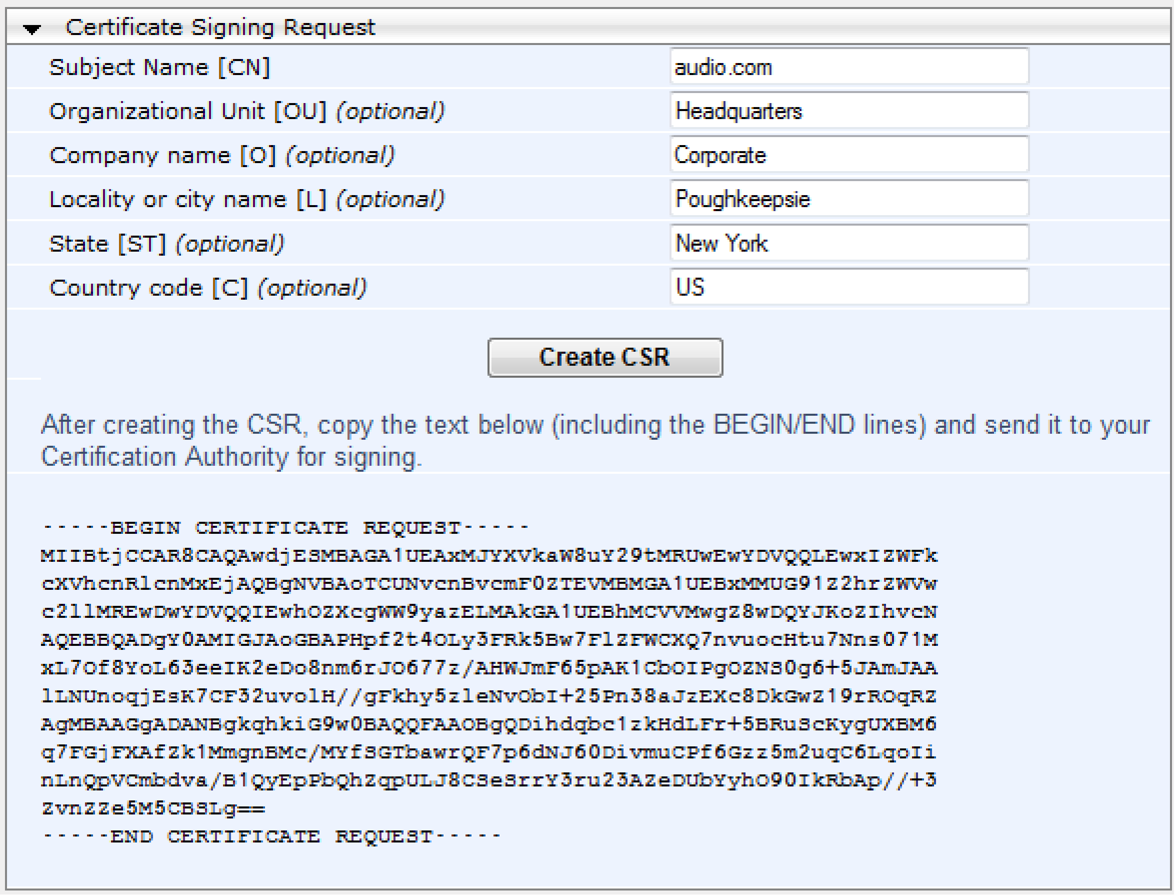
13.2 Assigning CSR-based Certificates to TLS Contexts

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

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

1. Your network administrator should allocate a unique DNS name for the device (e.g., dns_name.corp.customer.com). This DNS name is used to access the device and therefore, must be listed in the server certificate.
2. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Certificates** ➡ button, located at the bottom of the TLS Contexts page; the Context Certificates page appears.
4. Under the **Certificate Signing Request** group, do the following:
   a. In the 'Subject Name [CN]' field, enter the DNS name.
   b. Fill in the rest of the request fields according to your security provider's instructions.
   c. Click the **Create CSR** button; a textual certificate signing request is displayed in the area below the button:

Figure 13-2: 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

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

```
-----BEGIN CERTIFICATE REQUEST-----
MIIBtjCCAR8CAQAwdjESMBAGA1UEAxMJYXVkaW8uY29tMRUwEwYDVQQLEwxIZWFk
cXVhcnRlcnMxEjAQBgNVBAoTCUNvcnBvcmF0ZTEVMBMGA1UEBxMMUG91Z2hrZWVw
c2llMREwDwYDVQQIEwhOZXcgWW9yazELMAkGA1UEBhMCVVMwgZ8wDQYJKoZIhvcN
AQEBBQADgY0AMIGJAoGBAPHpf2t4OLy3FRk5Bw7FlZFWCXQ7nvuocHtu7Nns071M
xL7Of8YoL63eeIK2eDo8nm6rJO677z/AHWJmF65pAK1CbOIPgOZNS0g6+5JAmJAA
lLNUnoqjEsK7CF32uvolH//gFkhy5zleNvObI+25Pn38aJzEXc8DkGwZ19rROqRZ
AgMBAAGgADANBgkqhkiG9w0BAQQFAAOBgQDihdqbc1zkHdLFr+5BRuScKygUXBM6
q7FGjFXAfZk1MmgnBMc/MYfSGTbawrQF7p6dNJ60DivmuCPf6Gzz5m2uqC6LgoIi
nLnQpVCmbdva/B1QyEpPbQhZqpULJ8CSeSrrY3ru23AzeDUbYyhO90IkRbAp//+3
Zvnzze5M5CBSLg==
-----END CERTIFICATE REQUEST-----
```

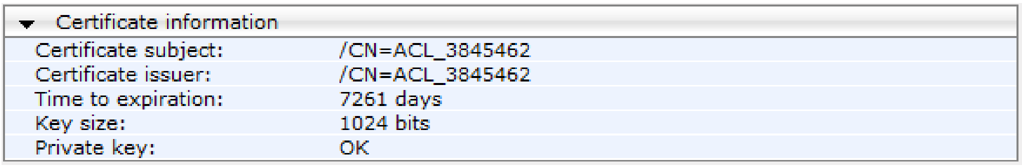
5. Copy the text and send it to your security provider (CA) to sign this request.

6. When the CA sends you a server certificate, save the certificate to a file (e.g., cert.txt). Ensure that the file is a plain-text file containing the"'BEGIN CERTIFICATE" header, as shown in the example of a Base64-Encoded X.509 Certificate below:

```
-----BEGIN CERTIFICATE-----
MIIDkzCCAnugAwIBAgIEAgAAADANBgkqhkiG9w0BAQQFADA/MQswCQYDVQQGEw
JGUjETMBEGA1UEChMKQ2VydGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBT
ZXJ2ZXVyMB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1
UEBhMCRlIxEzARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9z
dGUgU2VydmV1cjCCASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4Mz
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 13-3: Private key "OK" in Certificate Information Group



▼ Certificate information

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

Notes:

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

13.3 Assigning Externally Created Private Keys to TLS Contexts

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

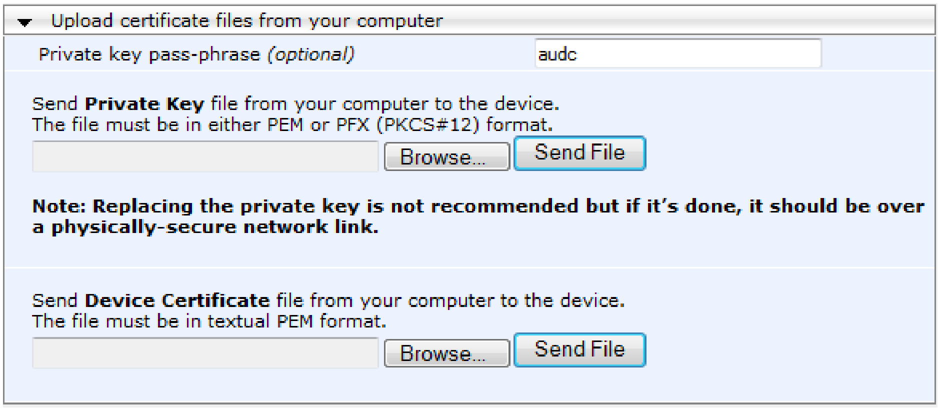
➢ **To assign an externally created private key to a TLS Context:**

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

the Context Certificates page appears.

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

Figure 13-4: Upload Certificate Files from your Computer Group



Upload certificate files from your computer

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

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

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

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

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.

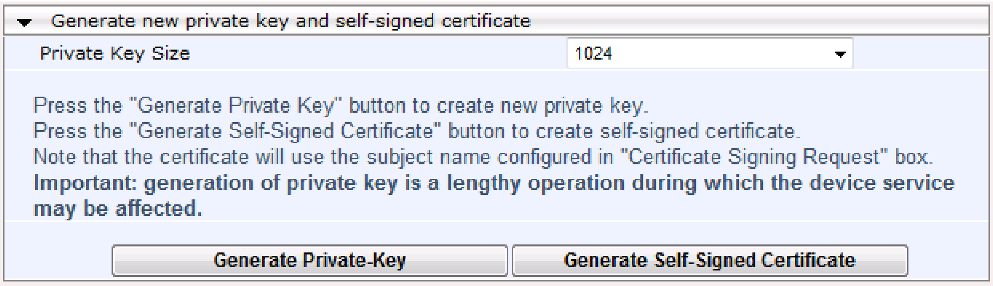
13.4 Generating Private Keys for TLS Contexts

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 TLS Contexts table, select the required TLS Context index row, and then click the **Context Certificates** ➡ button, located at the bottom of the TLS Contexts page; the Context Certificates page appears.
3. Scroll down to the **Generate new private key and self-signed certificate** group:

Figure 13-5: Generate new private key and self-signed certificate Group



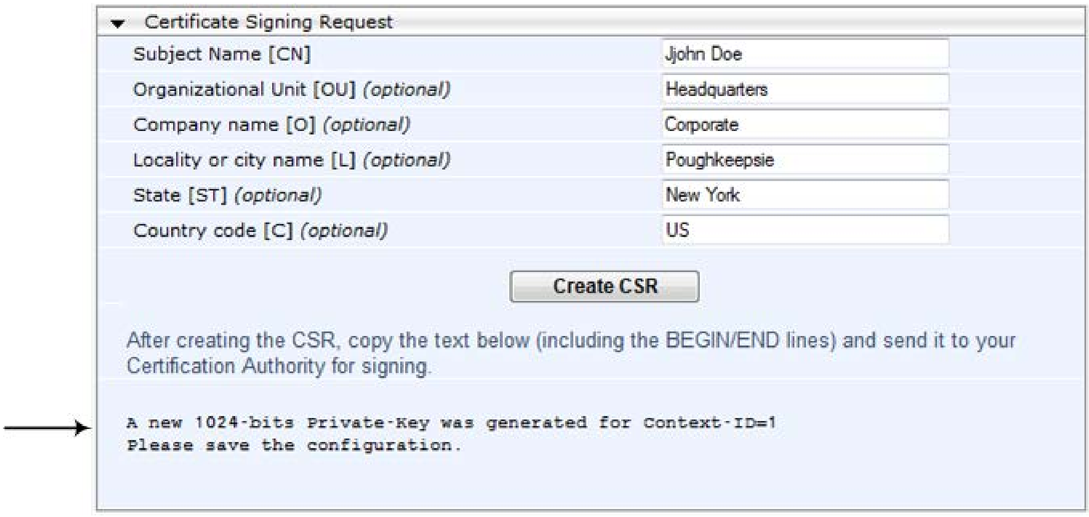
Generate new private key and self-signed certificate

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.

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
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 13-6: Indication of Newly Generated Private Key



▾ Certificate Signing Request

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

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

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

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

13.5 Creating Self-Signed Certificates for TLS Contexts

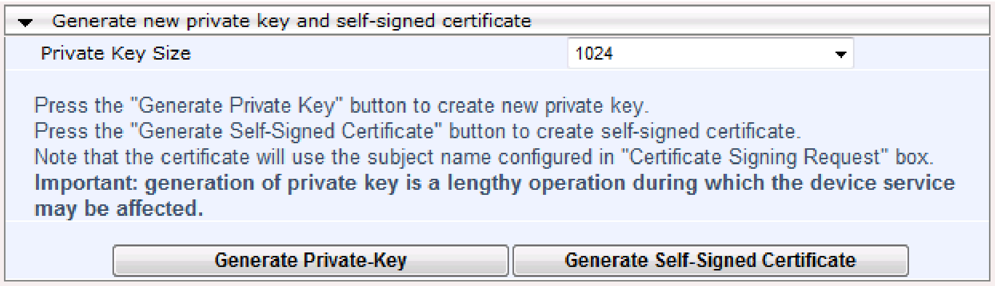
The following procedure describes how to assign a certificate that is digitally signed by the device itself to a TLS Context. In other words, the device acts as a CA.

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

1. Before you begin, make sure that:
   - You have a unique DNS name for the device (e.g., dns_name.corp.customer.com). This name is used to access the device and therefore, must be listed in the server certificate.
   - No traffic is running on the device. The certificate generation process is disruptive to traffic and should be done during maintenance time.
2. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Certificates** ➡ button, located at the bottom of the TLS Contexts page; the Context Certificates page appears.
4. Under the **Certificate Signing Request** group, in the 'Subject Name [CN]' field, enter the fully-qualified DNS name (FQDN) as the certificate subject.

5. Scroll down the page to the **Generate new private key and self-signed certificate** group:

Figure 13-7: 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.

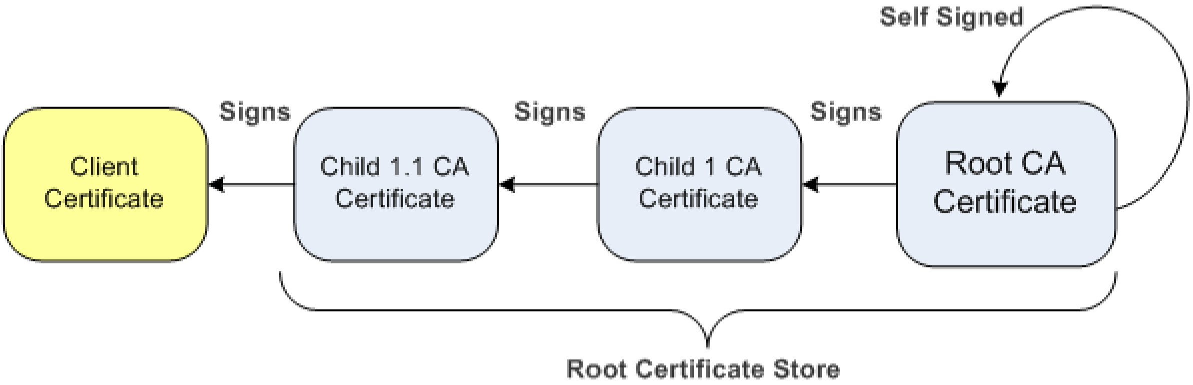
6. Click **Generate Self-Signed Certificate**; a message appears (after a few seconds) displaying the new subject name.
7. Save the configuration with a device reset for the new certificate to take effect.

13.6 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 13-8: Certificate Chain Hierarchy



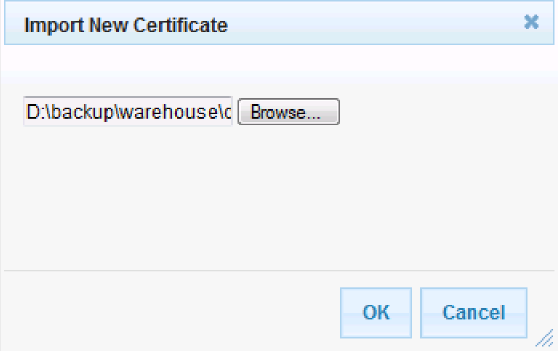
For the device to trust a whole chain of certificates per TLS Context, you need to add them to the device's Trusted Certificates Store, as described below.

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

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

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

Figure 13-9: 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**.

13.7 Configuring Mutual TLS Authentication

13.7.1 TLS for SIP Clients

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

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

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

➢ **To configure mutual TLS authentication for SIP messaging:**

1. Enable two-way authentication on the specific SIP Interface:
   a. In the SIP Interface Table page (see "Configuring SIP Interfaces" on page 267), set the 'TLS Mutual Authentication' parameter to **Enable** for the specific SIP Interface.
   b. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

2. Configure a TLS Context with the following certificates:
    - Import the certificate of the CA that signed the certificate of the SIP client, into the Trusted Root Store so that the device can authenticate the client (see "Importing Certificates and Certificate Chain into Trusted Certificate Store" on page 117).
    - Make sure that the TLS certificate is signed by a CA that the SIP client trusts so that the client can authenticate the device.

13.7.2 TLS for Remote Device Management

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

➢ **To enable mutual TLS authentication for HTTPS:**

1. Set the 'Secured Web Connection (HTTPS)' field to **HTTPS Only** in the Web Security Settings page (see "Configuring Web Security Settings" on page 71) to ensure you have a method for accessing the device in case the client certificate does not work. Restore the previous setting after testing the configuration.
2. Open the TLS Contexts page (**Configuration** tab > **System** menu > **TLS Contexts**).
3. In the TLS Contexts table, select the required TLS Context index row, and then click the **Context Trusted-Roots** ➡ button, located at the bottom of the TLS Contexts page; the Trusted Certificates page appears.
4. Click the **Import** button, and then select the certificate file.
5. When the operation is complete, set the 'Requires Client Certificates for HTTPS connection' field to **Enable** in the Web Security Settings page.
6. Save the configuration with a device reset (see "Saving Configuration" on page 592).

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).

13.8 Configuring TLS Server Certificate Expiry Check

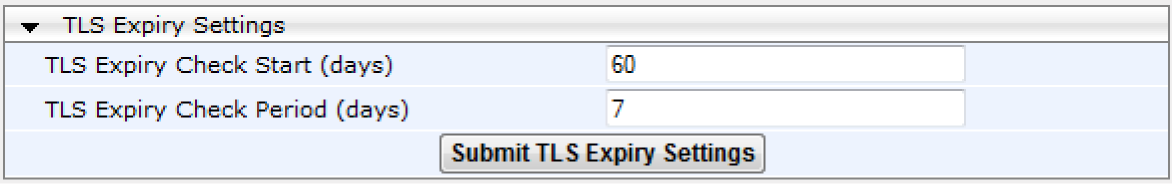
You can also configure the TLS Server Certificate Expiry Check feature, whereby the device periodically checks the validation date of the installed TLS server certificates. You can also configure the device to send a notification SNMP trap event (acCertificateExpiryNotification) at a user-defined number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

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

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

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

Figure 13-10: TLS Expiry Settings Group



TLS Expiry Settings

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

3. In the 'TLS Expiry Check 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.

14 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.

14.1 Configuring Date and Time Manually

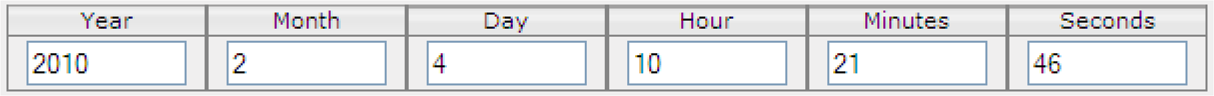
You can manually configure the date and time of the device (instead of using an NTP server), as described in the procedure below. You can also configure the following with your manually configured date and time:

- Daylight Saving Time (DST) - see 'Configuring Daylight Saving Time' on page 123
- UTC time offset (e.g., GMT +1). To configure the offset, use the 'NTP UTC Offset' (NTPServerUTCOffset) parameter (see 'Configuring Automatic Date and Time using SNTP' on page 121)

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

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

Figure 14-1: Regional Settings Page



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

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

Notes:

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

14.2 Configuring Automatic Date and Time using SNTP

The device's Simple Network Time Protocol (SNTP) client functionality generates requests and reacts to the resulting responses using the NTP Version 3 protocol definitions (according to RFC 1305). Through these requests and responses, the device, as an NTP client, synchronizes the system time to a time source within the network, thereby eliminating any potential issues should the local system clock 'drift' during operation. The NTP client follows a simple process in managing system time: the NTP client requests an NTP update, receives an NTP response, and then updates the local system clock based on an NTP server within the network. The client requests a time update from the user-defined NTP server (IP address or FQDN) at a user-defined update interval. Typically, this update interval is every 24 hours based on when the system was restarted.

You can also configure a time offset for the time received from the NTP server, according to your region. For example, Germany Berlin region is UTC/GMT +1 hours and therefore, you would configure the offset to "1". For USA New York, the UTC/GMT offset is -5 hours and therefore, the offset is a minus value and configured as "-5". To configure Daylight Saving Time (DST), see 'Configuring Daylight Saving Time' on page 123.

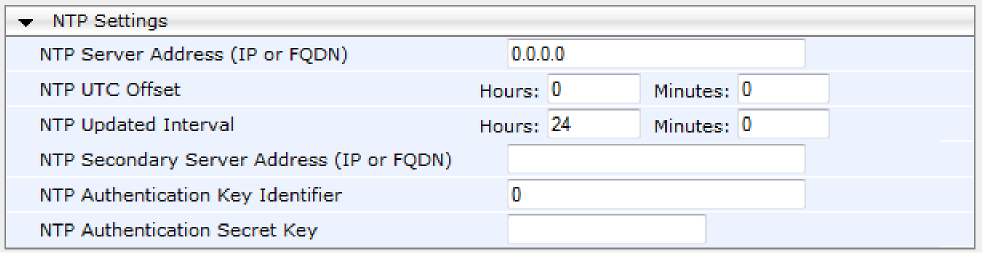
You can also configure the device to authenticate and validate the NTP messages received from the NTP server. Authentication is done using an authentication key with the MD5 cryptographic hash algorithm. When this feature is enabled, the device ignores NTP messages received without authentication.

The following procedure describes how to configure SNTP. For detailed descriptions of the configuration parameters, see NTP and Daylight Saving Time Parameters on page 783.

➢ To configure SNTP using the Web interface:

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

Figure 14-2: SNTP Configuration in Application Settings Page



▾ NTP Settings

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

3. Configure the NTP server address:
   - In the 'NTP Server Address' (NTPServerIP) field, configure the primary NTP server's address (IP or FQDN).
   - In the 'NTP Secondary Server Address' (NTPSecondaryServerIP) field, configure the secondary NTP server.
4. In the 'NTP UTC Offset' (NTPServerUTCOffset) field, configure the time offset in relation to the UTC. For example, if your region is GMT +1 (an hour ahead), enter "1".
5. In the 'NTP Updated Interval' (NTPUpdateInterval) field, configure the period after which the date and time of the device is updated.
6. Configure NTP message authentication:
   - In the 'NTP Authentication Key Identifier' field, configure the NTP authentication key identifier.
   - In the 'NTP Authentication Secret Key' field, configure the secret authentication key shared between the device and the NTP server.
7. Verify that the device has received the correct date and time from the NTP server. You can do this by viewing the date and time in the Regional Settings page (see 'Configuring Date and Time Manually' on page 121).

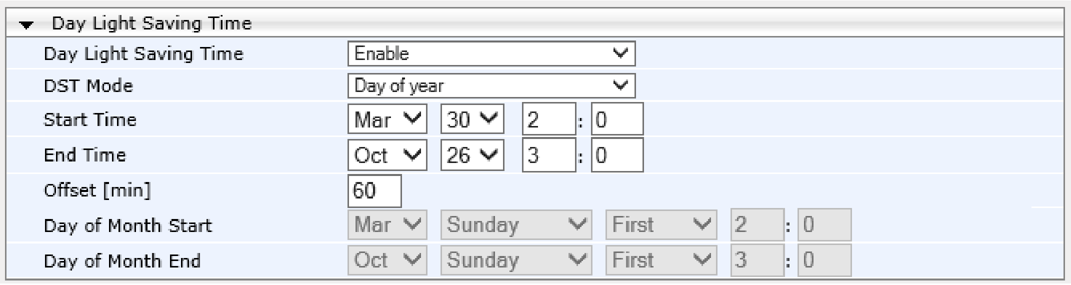
14.3 Configuring Daylight Saving Time

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

➢ To configure DST using the Web interface:

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

Figure 14-3: Configuring DST



▼ Day Light Saving Time

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

3. From the 'Day Light Saving Time' (DayLightSavingTimeEnable) drop-down list, select **Enable**.
4. From the 'DST Mode' drop-down list, select the range type for configuring the start and end dates for DST:
    - **Day of year:** The range is configured by exact date (day number of month), for example, from March 30 to October 30. If 'DST Mode' is set to **Day of year**, in the 'Start Time' (DayLightSavingTimeStart) and 'End Time' (DayLightSavingTimeEnd) drop-down lists, configure the period for which DST is relevant.
    - **Day of month:** The range is configured by month and day type, for example, from the last Sunday of March to the last Sunday of October. If 'DST Mode' is set to **Day of month**, in the 'Day of Month Start' and 'Day of Month End' drop-down lists, configure the period for which DST is relevant.
5. In the 'Offset' (DayLightSavingTimeOffset) field, configure the DST offset in minutes.
6. If the current date falls within the DST period, verify that it has been successful applied to the device's current date and time. You can view the device's date and time in the Regional Settings page (see 'Configuring Date and Time Manually' on page 121).

This page is intentionally left blank.

15 Configuring Power over Ethernet

The device supports Power over Ethernet (PoE) according to the IEEE 802.3af-2003 and IEEE 802.3at standards, providing power on the Ethernet lines through all the Ethernet LAN ports. The ports can transfer electrical power, along with the usual data, over the Ethernet cable to connected equipment (e.g., IP phones) that are capable of receiving PoE.

The LAN ports automatically detect the presence of IEEE 802.3 compliant equipment. Upon plugging in a PoE client to one of the ports, the device also automatically detects the class to which the client belongs and consequently, the maximum power allowed:

- IEEE 802.3af-2003: The device can supply up to 15.4W per port, and a total budget of 50W or 120W (depending on model) for all ports:
  - Class 0: configurable, up to 15.4W
  - Class 1: up to 4W
  - Class 2: up to 7W
  - Class 3: up to 15.4W
- IEEE 802.3at – Class 4: The device can supply up to 30W per port to the connected equipment and a total budget of 50W, 120W, or 200W (depending on model) for all ports.

PoE is supplied on Pins 4,5: (+), pins 7,8: (-).

You can enable PoE per port and set the maximum port power consumption (up to 15.4W) when the plugged-in client is detected as Class 0. If the plugged-in client is detected as Class 0, the device saves the user-defined wattage from the total wattage budget (i.e., 15.4W). If the plugged-in client is detected as Class 1, Class 2, or Class 3, the device saves 4W, 7W, or 15.4W respectively from the total wattage budget. If the power budget has been exhausted and a new client is plugged in, power is unavailable to this client. You can also enable Class 4 PoE per port.

Notes:

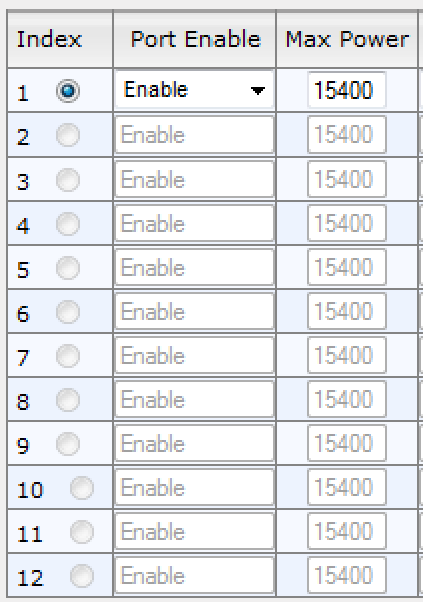
- PoE is a customer ordered feature.
- Upon device startup, PoE is enabled on all LAN ports.
- The power is always taken off the total budget according to the class detected, regardless of what is actually consumed per port.
- To view allocated power per port and various PoE allocation information, see "Viewing Ethernet Port Information" on page 662.

The following procedure describes how to configure PoE in the Web interface. You can also configure this using the table ini file parameter, POETable or CLI command, configure system/poe.

➢ To configure PoE:

1. Open the Power Over Ethernet Settings page (**Configuration** tab > **System** menu > **Power over Ethernet Settings**).

Figure 15-1: Power Over Ethernet Settings Page



Index	Port Enable	Max Power
1 ●	Enable ▾	15400
2 ○	Enable	15400
3 ○	Enable	15400
4 ○	Enable	15400
5 ○	Enable	15400
6 ○	Enable	15400
7 ○	Enable	15400
8 ○	Enable	15400
9 ○	Enable	15400
10 ○	Enable	15400
11 ○	Enable	15400
12 ○	Enable	15400

2. Select the 'Index' radio button corresponding to the required LAN port.
3. Click the **Edit** button.
4. From the 'Port Enable' drop-down list, select whether you want to enable or disable PoE.
5. In the 'Max Power' field, enter the required maximum power consumption for the port.
6. From the 'AT Enable' drop-down list, select whether you want to enable PoE according to IEEE 802.3at.
7. Click **Apply**.

Part IV

General VoIP Configuration

16 Network

This section describes the network-related configuration.

16.1 Configuring Underlying Ethernet Devices

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

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

You can view configured Ethernet Devices that have been successfully applied to the device (saved to flash), in the Ethernet Device Status Table page. This page is accessed by clicking the **Ethernet Device Status Table** button, located at the bottom of the Ethernet Device Table page. The Ethernet Device Status Table page can also be accessed from the **Status & Diagnostics** tab > **VoIP Status** menu > **Ethernet Device Status Table** (see "Viewing Ethernet Device Status" on page 680).

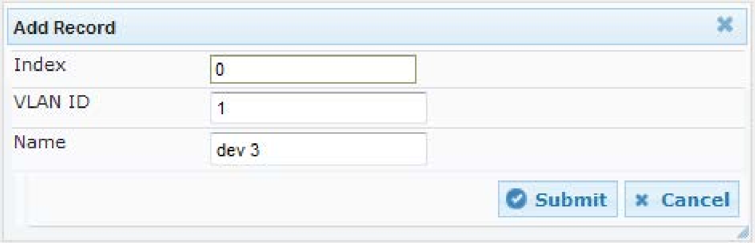
> **Note:** You cannot delete an Ethernet Device that is associated with an IP network interface (in the Interface table). Only after the Ethernet Device has been disassociated from the IP network interface can it be deleted.

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

➢ **To configure an Ethernet Device:**

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

Figure 16-1: Ethernet Device Table - Add Record



Add Record

Index	0
VLAN ID	1
Name	dev 3

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

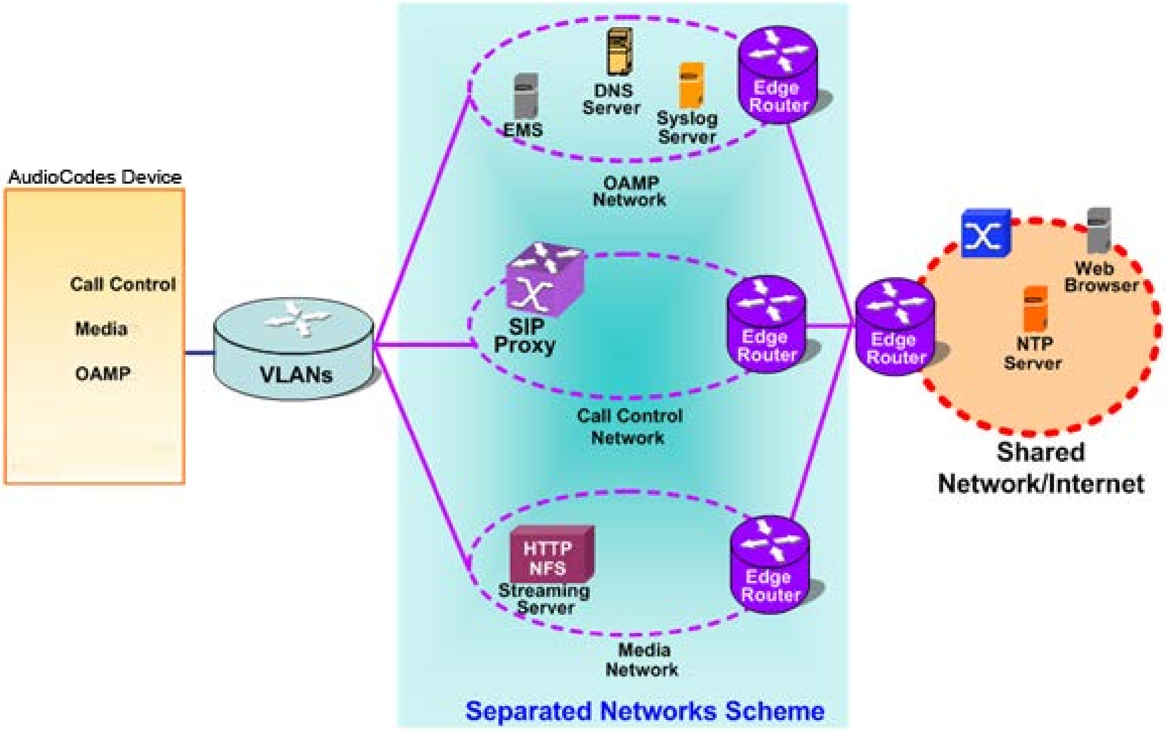
Table 16-1: Ethernet Device Table Parameter Descriptions

Parameter	Description
Index **[DeviceTable_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
VLAN ID CLI: vlan-id **[DeviceTable_VlanID]**	Defines a VLAN ID. The valid value is 1 to 3999. The default value is 1.
Name CLI: name **[DeviceTable_DeviceName]**	Defines a name for the VLAN. This name is used to associate the VLAN with an IP network interface in the Interface table ('Underlying Device' field - see "Configuring IP Network Interfaces" on page 130) and/or with a static route in the Static Route table ('Device Name' field - see "Configuring Static IP Routing" on page 139). By default, the device automatically assigns a name using the following syntax: "dev <next available table row index>" (e.g., "dev 3").

16.2 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 16-2: Multiple Network Interfaces



The device is shipped with a default OAMP interface. For more information, see "Default OAMP IP Address" on page 29. 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 12 interfaces, consisting of up to 11 Control and Media interfaces 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)
- IP address (IPv4 and IPv6) and subnet mask (prefix length)
- Complementing this network configuration is the On-Board Ethernet Switch configuration. This enables you to configure the VLAN IDs accessible through each physical port, as well as the Native VLAN ID per physical port. Layer3 (DiffServ) and Layer 2 (VLAN priority) Quality of Service parameters can also be configured. For configuring Quality of Service (QoS), see "Configuring the QoS Settings" on page 142.
- 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. 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 VoIP network static routing rules for non-local hosts/subnets. For more information, see "Configuring Static IP Routing" on page 139.

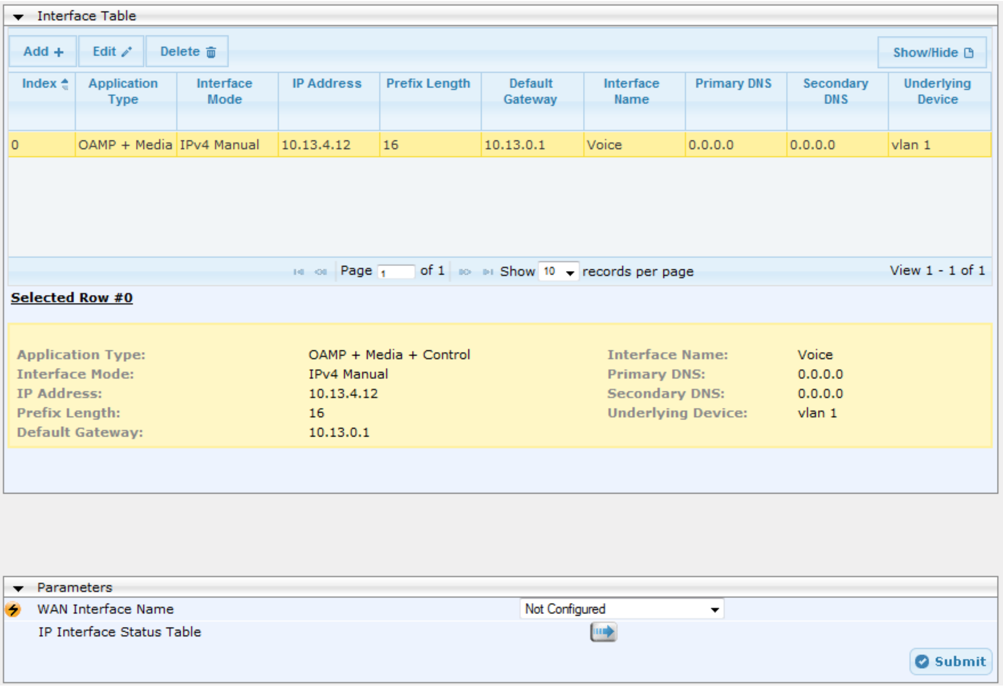
Notes:

- Before configuring IP interfaces, it is recommended that you read the IP interface configuration guidelines in "Interface Table Configuration Guidelines" on page 135.
- The IPv6 feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see Software License Key on page 621.

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

➢ To configure IP network interfaces:

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



Index	Application Type	Interface Mode	IP Address	Prefix Length	Default Gateway	Interface Name	Primary DNS	Secondary DNS	Underlying Device
0	OAMP + Media	IPv4 Manual	10.13.4.12	16	10.13.0.1	Voice	0.0.0.0	0.0.0.0	vlan 1

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

Selected Row #0

Application Type:	OAMP + Media + Control	Interface Name:	Voice
Interface Mode:	IPv4 Manual	Primary DNS:	0.0.0.0
IP Address:	10.13.4.12	Secondary DNS:	0.0.0.0
Prefix Length:	16	Underlying Device:	vlan 1
Default Gateway:	10.13.0.1		

Parameters

WAN Interface Name	Not Configured
IP Interface Status Table	

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

To view configured network interfaces that are currently active, click the **IP Interface Status Table** button. For more information, see "Viewing Active IP Interfaces" on page 680.

Table 16-2: Interface Table Parameters Description

Parameter	Description
Table parameters	
Index CLI: network-if **[InterfaceTable_Index]**	Table index row of the interface. The range is 0 to 11.
Web: Application Type EMS: Application Types CLI: application-type **[InterfaceTable_ApplicationTypes]**	Defines the applications allowed on the interface. ▪ **[0]** OAMP = Operations, Administration, Maintenance and Provisioning (OAMP) applications (e.g., Web, Telnet, SSH, and SNMP). ▪ **[1]** Media = Media (i.e., RTP streams of voice). ▪ **[2]** Control = Call Control applications (e.g., SIP). ▪ **[3]** OAMP + Media = OAMP and Media applications. ▪ **[4]** OAMP + Control = OAMP and Call Control applications. ▪ **[5]** Media + Control = Media and Call Control applications. ▪ **[6]** OAMP + Media + Control = All application types are allowed on the interface.

Parameter	Description
Web: Interface Mode [InterfaceTable_InterfaceMode]	Defines the method that the interface uses to acquire its IP address. ▪ [3] IPv6 Manual Prefix = IPv6 manual prefix IP address assignment. The IPv6 prefix (higher 64 bits) is set manually while the interface ID (the lower 64 bits) is derived from the device's MAC address. ▪ [4] IPv6 Manual = IPv6 manual IP address (128 bits) assignment. ▪ [10] IPv4 Manual = IPv4 manual IP address (32 bits) assignment.
Web/EMS: IP Address CLI: ip-address **[InterfaceTable_IPAddress]**	Defines the IPv4/IPv6 address, in dotted-decimal notation.
Web/EMS: Prefix Length CLI: prefix-length **[InterfaceTable_PrefixLength]**	Defines the prefix length of the related IP address. This is a Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation. The CIDR-style representation uses a suffix indicating the number of bits which are set in the dotted-decimal format. For example, 192.168.0.0/16 is synonymous with 192.168.0.0 and subnet 255.255.0.0. This CIDR lists the number of ‘1’ bits in the subnet mask (i.e., replaces the standard dotted-decimal representation of the subnet mask for IPv4 interfaces). For example, a subnet mask of 255.0.0.0 is represented by a prefix length of 8 (i.e., 11111111 00000000 00000000 00000000) and a subnet mask of 255.255.255.252 is represented by a prefix length of 30 (i.e., 11111111 11111111 11111111 11111100). The prefix length is a Classless Inter-Domain Routing (CIDR) style presentation of a dotted-decimal subnet notation. The CIDR-style presentation is the latest method for interpretation of IP addresses. Specifically, instead of using eight-bit address blocks, it uses the variable-length subnet masking technique to allow allocation on arbitrary-length prefixes. The prefix length for IPv4 must be set to a value from 0 to 30. The prefix length for IPv6 must be set to a value from 0 to 64.
Web/EMS: Default Gateway CLI: gateway **[InterfaceTable_Gateway]**	Defines the IP address of the default gateway for the interface. When traffic is sent from this interface to an unknown destination (i.e., not in the same subnet and not defined for any static routing rule), it is forwarded to this default gateway. **Note:** When using both voice and data-routing functionalities, it is recommended to set this Default Gateway's IP address in the same subnet and VLAN ID as the IP address configured for data-routing.
Web/EMS: Interface Name CLI: name **[InterfaceTable_InterfaceName]**	Defines a name for the interface. This name is used in various configuration tables to associate the network interface with other configuration entities such as Media Realms. It is also displayed in management interfaces (Web, CLI, and SNMP) for clarity where it has no functional use. The valid value is a string of up to 16 characters.
Web/EMS: Primary DNS CLI: primary-dns **[InterfaceTable_PrimaryDNSServerIPAddress]**	(Optional) Defines the primary DNS server's IP address (in dotted-decimal notation), which is used for translating domain names into IP addresses for the interface. By default, no IP address is defined.

Parameter	Description
Web/EMS: Secondary DNS CLI: secondary-dns **[InterfaceTable_SecondaryDNSServerIPAddress]**	(Optional) Defines the secondary DNS server's IP address (in dotted-decimal notation), which is used for translating domain names into IP addresses for the interface. By default, no IP address is defined.
Web: Underlying Device CLI: underlying-dev [InterfaceTable_UnderlyingDevice]	Assigns an Ethernet Device to the IP interface. An Ethernet Device is a VLAN ID associated with a physical Ethernet port (Ethernet Group). To configure Ethernet Devices, see Configuring Underlying Ethernet Devices on page 129.
WAN Interface Name CLI: bind interface <interface name> voip [WanInterfaceName]	Assigns a WAN interface to the VoIP traffic (i.e., SIP signaling and media / RTP interfaces). The available WAN interface options depends on the hardware configuration (e.g., Ethernet and SHDSL) and/or whether VLANs are defined for the WAN interface. If VLANs are configured, for example, for the Ethernet WAN interface, you can select the WAN VLAN on which you want to run the SIP signaling and/or media interfaces. The WAN interface can be assigned to SIP signaling and media interfaces in the SIP Interface table (see Configuring SIP Interfaces on page 267) and Media Realm table (see Configuring Media Realms on page 259), where the WAN interface is denoted as "WAN". Once this association is set, VoIP traffic is sent via the WAN and incoming traffic is identified as coming from the WAN. The device also automatically configures the required port forwarding and static NAT rules. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If you do not assign the WAN interface to SIP and media interfaces, then the WAN interface may not be used for VoIP traffic. In such scenarios, the VoIP traffic can be sent and received within the LAN, or sent to the WAN via a third-party LAN router. If a third-party router is used as the interface to the WAN, then you need to define NAT rules (using the NATTranslation parameter) to translate the VoIP LAN IP addresses (defined in the Interface table and associated with SIP and media interfaces) into global, public IP addresses. ▪ This parameter is applicable only if the data-routing functionality is supported (i.e., relevant Software License Key is installed on the device).

16.2.1 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.

16.2.2 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.

16.2.3 Networking Configuration Examples

This section provides configuration examples of networking interfaces.

16.2.3.1 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 16-3: 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 16-4: 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.

16.2.3.2 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 16-5: 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 16-6: 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.

16.2.3.3 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 16-7: 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 16-8: Example of Static Route Table

Destination	Prefix Length	Gateway
176.85.49.0	24	192.168.0.10

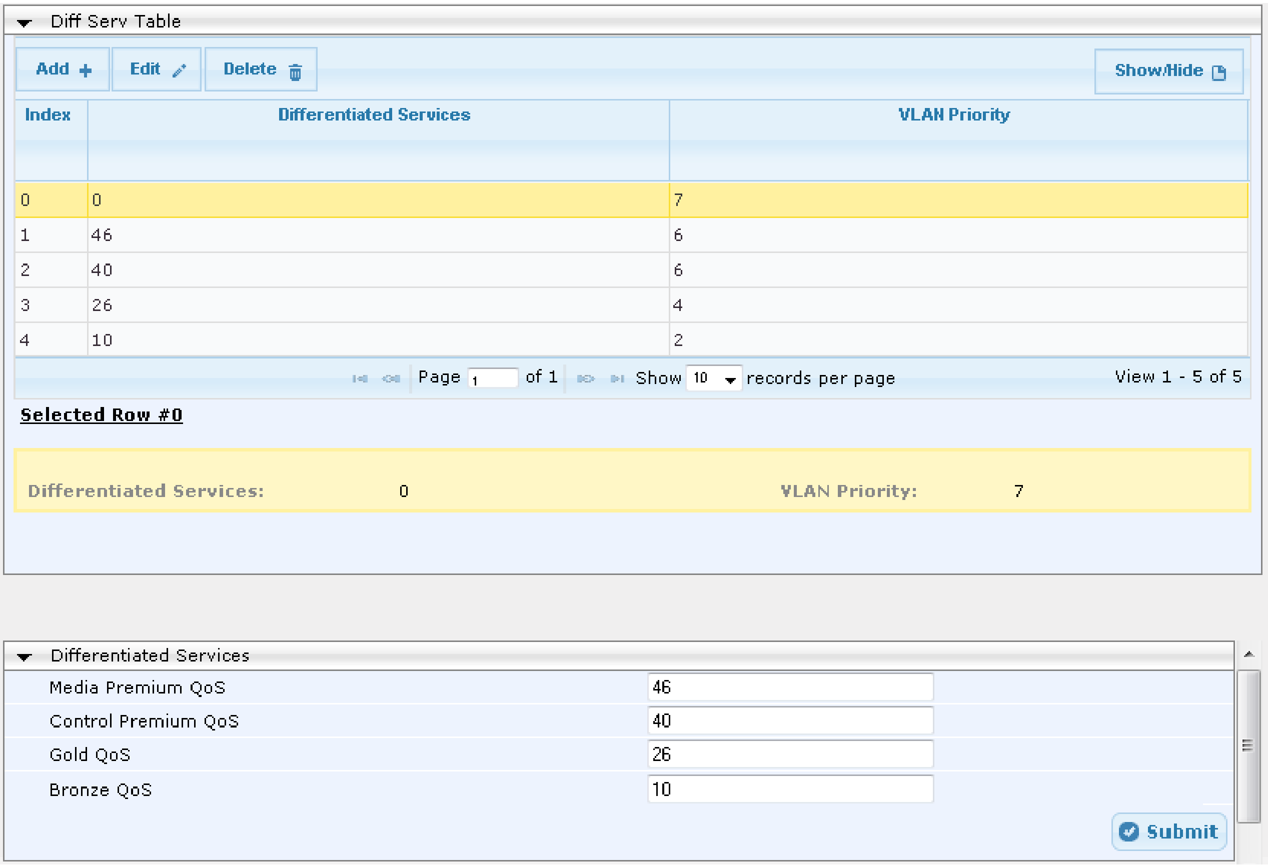
3. The NTP application is configured (using the ini file) to serve as OAMP applications:

```
EnableNTPasOAM = 1
```

4. DiffServ table:
   - Layer-2 QoS values are assigned:
     - For packets sent with DiffServ value of 46, set VLAN priority to 6
     - For packets sent with DiffServ value of 40, set VLAN priority to 6
     - For packets sent with DiffServ value of 26, set VLAN priority to 4
     - For packets sent with DiffServ value of 10, set VLAN priority to 2

- Layer-3 QoS values are assigned:
  - ♦ For Media Service class, the default DiffServ value is set to 46
  - ♦ For Control Service class, the default DiffServ value is set to 40
  - ♦ For Gold Service class, the default DiffServ value is set to 26
  - ♦ For Bronze Service class, the default DiffServ value is set to 10

Figure 16-3: Example of Layer-2 QoS in DiffServ Table



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

Selected Row #0

Differentiated Services: 0 VLAN Priority: 7

Differentiated Services

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

16.2.3.4 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 16-9: Configured Default Gateway Example

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

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

Table 16-10: 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

16.3 Configuring Static IP Routes

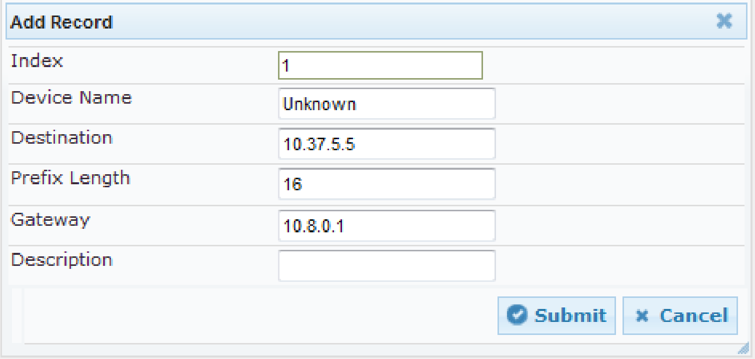
The Static Route table lets you configure up to 30 static IP routing rules. Using static routes lets you communicate with LAN networks that are not located behind the Default Gateway specified for the IP network interface, configured in the Interface table, from which the packets are sent.

You can view the status of the configured static routes in the IP Routing Status Table page. This page can be accessed by clicking the **Static Route Status Table** button, located at the bottom of the Static Route table page, or it can be accessed from the Navigation tree under the **Status & Diagnostics** tab (see "Viewing Static Routes Status" on page 681).

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

➢ **To configure a static IP route:**

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



Add Record

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

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

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

Table 16-11: Static Route Table Parameter Descriptions

Parameter	Description
Index **[StaticRouteTable_Index]**	Defines an index number for the new table record. The valid value is 0 to 29. **Note:** Each table row must be configured with a unique index.
Device Name CLI: device-name [StaticRouteTable_DeviceName]	Assigns an IP network interface through which the static route's Gateway is reached. The Device Name (or underlying device) represents the IP network interface, including VLAN ID and associated physical port(s). The value must be identical to the value in the 'Underlying Device' parameter of the required IP network interface in the Interface table (see Configuring IP Network Interfaces on page 130). For configuring Ethernet Devices, see Configuring Underlying Ethernet Devices on page 129.
Destination CLI: destination **[StaticRouteTable_Destination]**	Defines the IP address of the destination host/network. The destination can be a single host or a whole subnet, depending on the prefix length configured for this routing rule.
Prefix Length CLI: prefix-length **[StaticRouteTable_PrefixLength]**	Defines the Classless Inter-Domain Routing (CIDR)-style representation of a dotted-decimal subnet notation of the destination host/network. The CIDR-style representation uses a suffix indicating the number of bits that are set in the dotted-decimal format. For example, the value 16 represents subnet 255.255.0.0. The 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 CLI: gateway **[StaticRouteTable_Gateway]**	Defines the IP address of the Gateway (next hop) used for traffic destined to the subnet/host defined in the 'Destination' / 'Prefix Length' field. **Notes:** ▪ The Gateway's address must be in the same subnet as the IP address of the network interface that is associated with the static route (using the 'Device Name' parameter - see above). ▪ The IP network interface associated with the static route must be of the same IP address family (IPv4 or IPv6).
Description CLI: description **[StaticRouteTable_Description]**	Defines an arbitrary name to easily identify the static route rule. The valid value is a string of up to 20 characters.

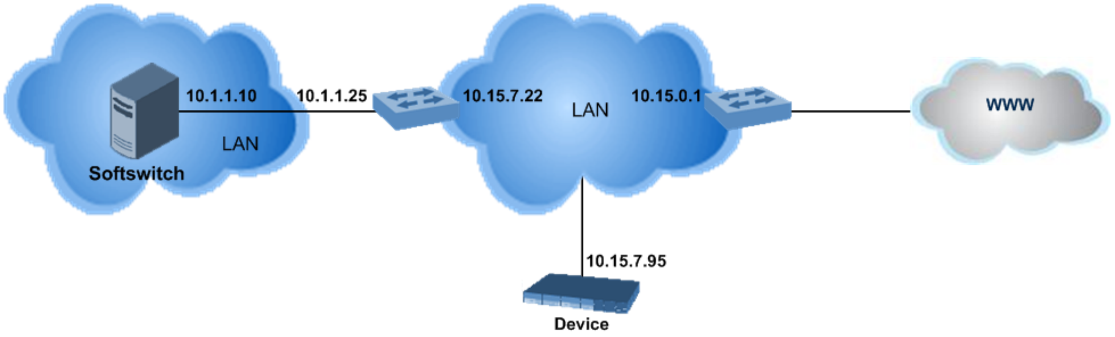
16.3.1 Configuration Example of Static IP Routes

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.

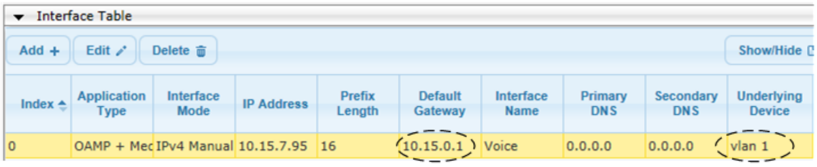
Note the following configuration:

- 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 16-4: Example of using a Static Route

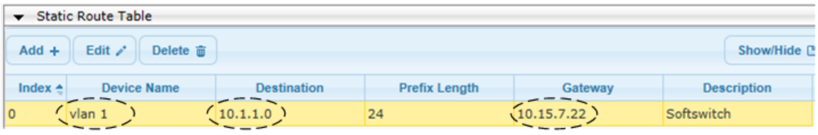


No Static Route:
The device sends packets to 10.15.0.1, which is the <u>Default Gateway</u> defined for this IP network interface in the Interface table. Therefore, the device will not succeed in reaching the softswitch.



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 <u>gateway</u> 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.



Index	Device Name	Destination	Prefix Length	Gateway	Description
0	vlan 1	10.1.1.0	24	10.15.7.22	Softswitch

16.3.2 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.

16.4 Configuring Quality of Service

The QoS Settings page lets you configure Layer-2 and Layer-3 Quality of Service (QoS) for VoIP. 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 16-12: 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

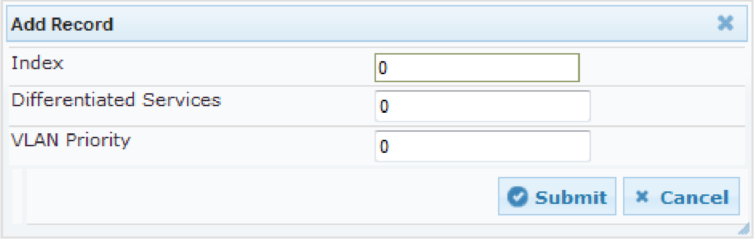
Application	Traffic / Network Types	Class-of-Service (Priority)
DNS client	Varies according to DNS settings: ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium Control ▪ Management: Bronze
NTP	Varies according to the interface type associated with NTP (see "Assigning NTP Services to Application Types" on page 135): ▪ OAMP ▪ Control	Depends on traffic type: ▪ Control: Premium control ▪ Management: Bronze

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

➢ **To configure QoS:**

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

Figure 16-5: DiffServ Table Page - Add Record



Add Record

Index: 0

Differentiated Services: 0

VLAN Priority: 0

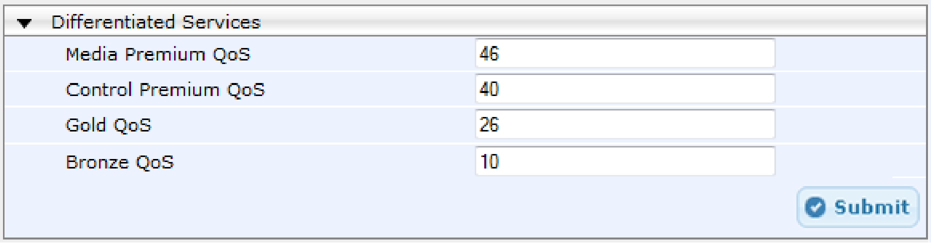
   b. Configure a DiffServ-to-VLAN priority mapping (Layer-2 QoS) according to the parameters described in the table below.
   c. Click Submit, and then save ("burn") your settings to flash memory.

Table 16-13: DiffServ Table Parameter Descriptions

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

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

Figure 16-6: QoS Settings Page - Differentiated Services



▼ Differentiated Services

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

16.5 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 144
- Internal SRV table - see "Configuring the Internal SRV Table" on page 145

16.5.1 Configuring the Internal DNS Table

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

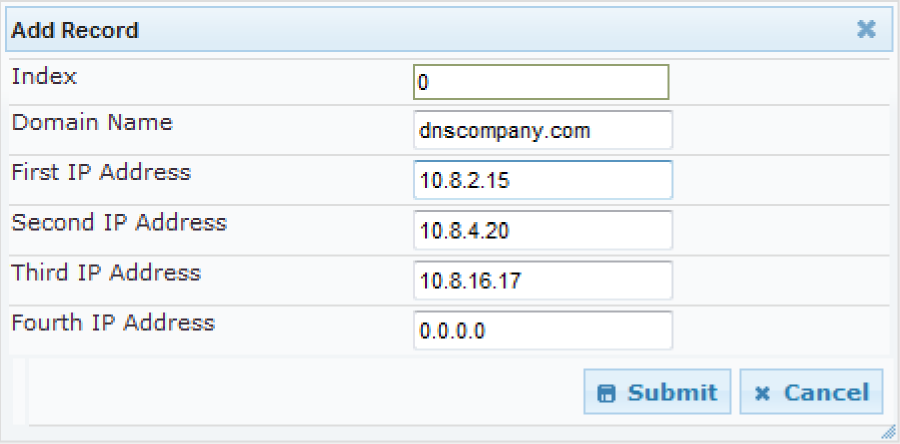
Note: The device initially attempts to resolve a domain name using the Internal DNS table. If the domain name is not configured in the table, the device performs a DNS resolution using an external DNS server for the related IP network interface (see "Configuring IP Network Interfaces" on page 130).

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

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

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

Figure 16-7: Internal DNS Table - Add Record Dialog Box



Add Record

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

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

Table 16-14: Internal DNS Table Parameter Description

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

16.5.2 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.

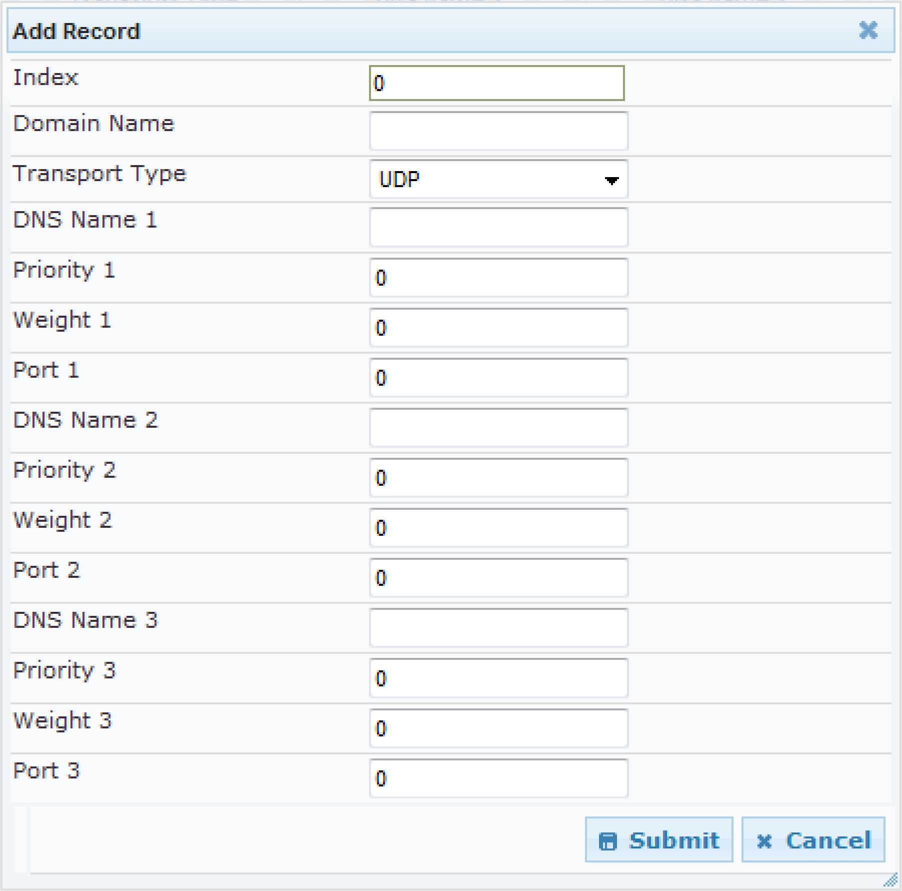
Note: If you configure the Internal SRV table, the device initially attempts to resolve a domain name using this table. If the domain is not configured in the table, the device performs a Service Record (SRV) resolution using an external DNS server, configured in the Interface table (see "Configuring IP Network Interfaces" on page 130).

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

➢ To configure an SRV rule:

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

Figure 16-8: Internal SRV Table Page



Add Record

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

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

Table 16-15: Internal SRV Table Parameter Descriptions

Parameter	Description
Web: Domain Name CLI: domain-name **[Srv2Ip_InternalDomain]**	Defines the host name to be translated. The valid value is a string of up to 31 characters.

Parameter	Description
Web: Transport Type CLI: transport-type **[Srv2Ip_TransportType]**	Defines the transport type. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS
Web: DNS Name (1-3) CLI: dns-name-1|2|3 **[Srv2Ip_Dns1/2/3]**	Defines the first, second or third DNS A-Record to which the host name is translated.
Web: Priority (1-3) CLI: priority-1|2|3 **[Srv2Ip_Priority1/2/3]**	Defines the priority of the target host. A lower value means that it is more preferred.
Web: Weight (1-3) CLI: weight-1|2|3 **[Srv2Ip_Weight1/2/3]**	Defines a relative weight for records with the same priority.
Web: Port (1-3) CLI: port-1|2|3 **[Srv2Ip_Port1/2/3]**	Defines the TCP or UDP port on which the service is to be found.

16.6 Open Solution Network (OSN) Server

This section describes various networking settings for the OSN server.

Notes:

- The OSN is a customer-ordered item.
- For information on cabling the OSN, refer to the device's *Hardware Installation Manual*.

16.6.1 Configuring Native VLAN for OSN Server

You can configure a VLAN ID associated with any application type (Media, Control, or OAMP) to access the Open Solution Network (OSN) server through the device's internal port switch. By default, access to the OSN is from the OAMP interface VLAN (ID 1). Configuring a Native VLAN is useful, for example, to separate traffic within the device - between the OSN (i.e., application traffic) and voice traffic (and data traffic).

This feature is applicable only if you connect to the OSN through one of the device's VoIP LAN ports (on the front panel). If you are connecting to the OSN through the OSN's Gigabit Ethernet port (on the rear panel), this feature is not applicable.

➢ **To configure a Native VLAN for the OSN:**

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).
2. In the 'OSN Native VLAN ID' field, set the VLAN ID.
3. Click **Submit**.

When set to 0 (default), the OSN uses the OAMP VLAN ID. When set to any other value, it specifies a VLAN configured in the Ethernet Device table (see "Configuring Underlying Ethernet Devices" on page 129), which is assigned to a Media and/or Control application in the Interface table.

16.6.2 Disabling Internal Switch Port for OSN

You can enable (default) or disable the Ethernet port of the device's internal switch that interfaces with the OSN. You can view the port status (Up or Down) by running the following CLI command:

```
# show system interface osn
```

➢ **To enable / disable the internal switch's Ethernet port interfacing with OSN:**

1. Open the Application Settings page (**Configuration** tab > **System** menu > **Application Settings**).
2. From the 'Block OSN Port' drop-down list, select **Enable** or **Disable**.
3. Click **Submit**.

16.7 Configuring NFS Settings

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

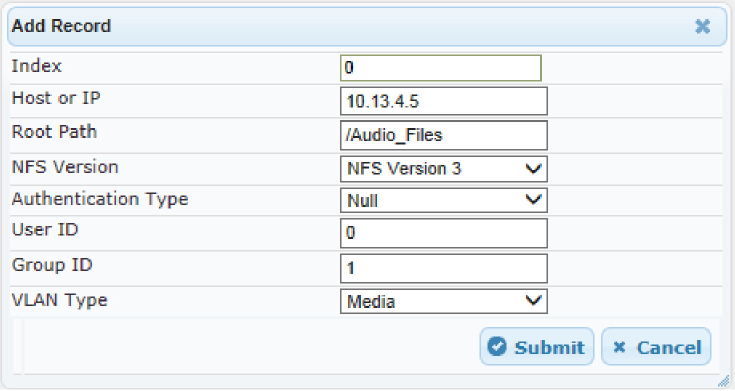
You can configure up to 16 different NFS file systems. As a file system, the NFS is independent of machine types, operating systems and network architectures. Note that an NFS file server can share multiple file systems. There must be a separate row for each remote file system shared by the NFS file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

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

➢ **To configure an NFS file systems:**

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

Figure 16-9: NFS Table Page - Add Record



Add Record

Index	0
Host or IP	10.13.4.5
Root Path	/Audio_Files
NFS Version	NFS Version 3
Authentication Type	Null
User ID	0
Group ID	1
VLAN Type	Media

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

Notes:

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

Table 16-16: NFS Table Parameter Descriptions

Parameter	Description
Web: Index	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Web: Host or IP CLI: host **[NFSServers_HostOrIP]**	The domain name or IP address of the NFS server. If a domain name is provided, a DNS server must be configured.
Web: Root Path CLI: root-path **[NFSServers_RootPath]**	Path to the root of the remote file system in the format: /[path]. For example, '/audio'.
Web: NFS Version CLI: version **[NFSServers_NfsVersion]**	NFS version used to access the remote file system. ▪ **[2]** NFS Version 2 ▪ **[3]** NFS Version 3 (default)
Web: Authentication Type CLI: authentication-type **[NFSServers_AuthType]**	Authentication method used for accessing the remote file system. ▪ **[0]** Null ▪ **[1]** Unix (default)
Web: User ID CLI: uid **[NFSServers_UID]**	User ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 0.
Web: Group ID CLI: gid **[NFSServers_GID]**	Group ID used in authentication when using Unix. The valid range is 0 to 65537. The default is 1.
Web: VLAN Type CLI: vlan-type **[NFSServers_VlanType]**	The VLAN type for accessing the remote file system. ▪ **[0]** OAMP ▪ **[1]** Media (default) **Note:** This parameter applies only if VLANs are enabled or if Multiple IPs is configured (see "Configuring IP Network Interfaces" on page 130).

16.8 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.

16.8.1 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 151.

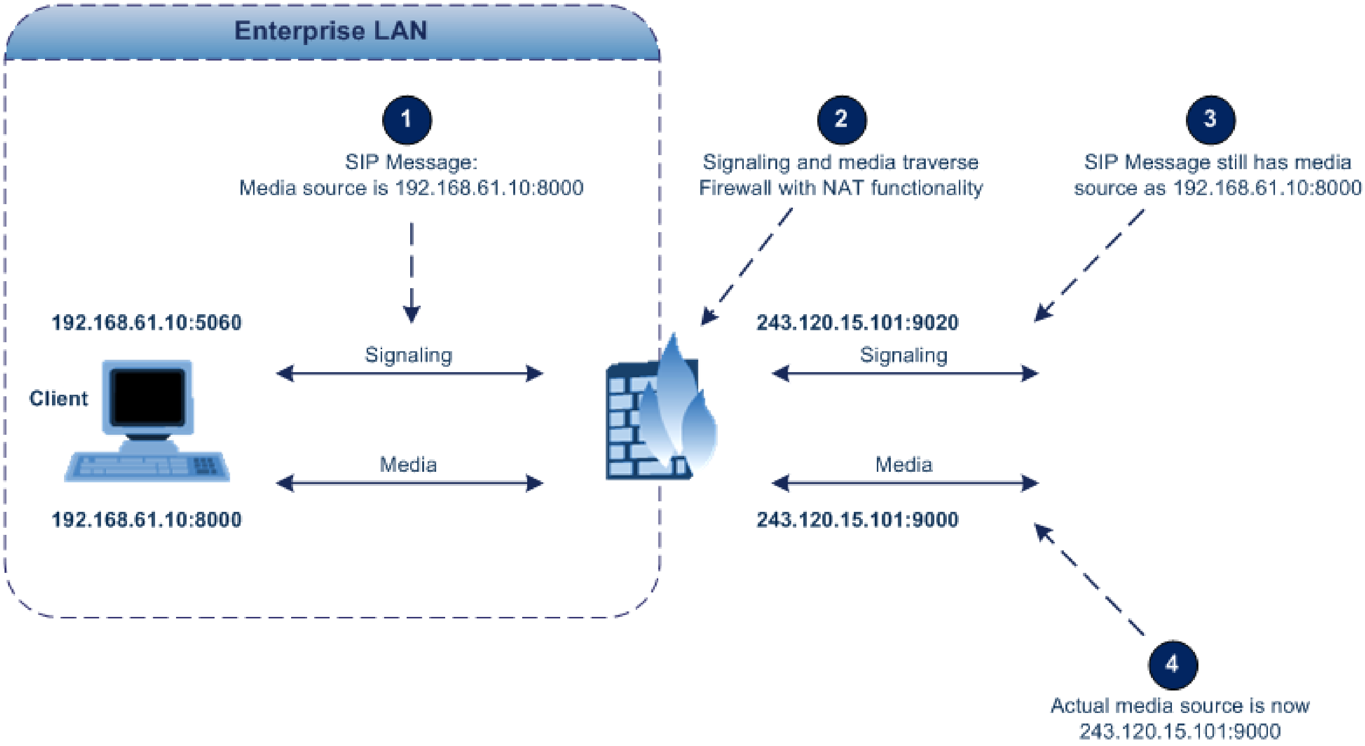
b. If configured, uses the NAT Translation table which configures NAT per interface - see Configuring NAT Translation per IP Interface on page 151.

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.

Note: The priority list above is applicable only to the Gateway calls.

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

Figure 16-10: Device behind NAT and NAT Issues


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

16.8.1.1 Configuring a Static NAT IP Address for All Interfaces

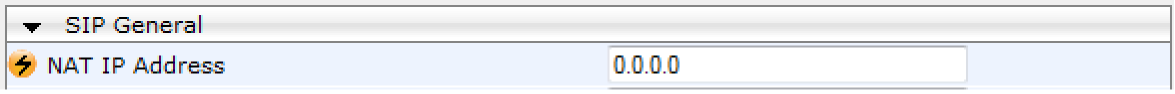
You can configure a global (public) IP address of the router to enable static NAT between the device and the Internet for all network interfaces. Thus, the device replaces the source IP address for media of all outgoing SIP messages sent on any of its network interfaces to this public IP address.

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

➢ **To configure a single static NAT IP address:**

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

Figure 16-11: Configuring Static NAT IP Address in SIP General Parameters Page



NAT IP Address 0.0.0.0

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

16.8.1.2 Configuring NAT Translation per IP Interface

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

If the device is configured with two network interfaces, for example, one LAN and one WAN, only one NAT rule is required and without specifying ports. This rule is defined with the network interface representing the WAN and with a public IP address. If the device is configured with only one network interface (e.g., "Voice") and you have an SRD configured for WAN and LAN, then you need to specify ports in order to differentiate between these SRDs. In such a scenario, the device replaces the source IP address only for messages sent from the WAN SRD, not from the LAN SRD.

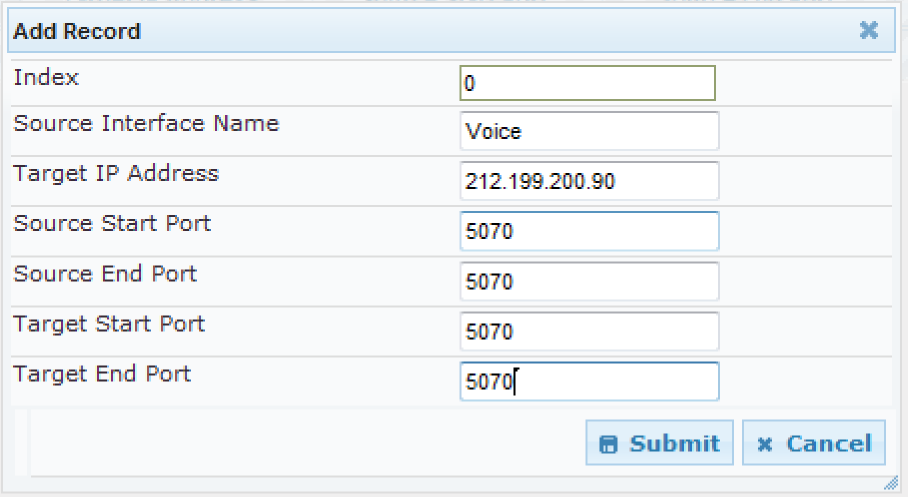
The following procedure describes how to configure NAT translation rules in the Web interface. You can also configure Bandwidth Profiles using the table ini file parameter, NATTranslation or CLI command, voip-network NATTranslation.

➢ **To configure NAT translation rules:**

1. Open the NAT Translation Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **NAT Translation Table**).

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

Figure 16-12: NAT Translation Table Page



Add Record

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

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

Table 16-17: NAT Translation Table Parameter Descriptions

Parameter	Description
Index CLI: index **[NATTranslation_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Source Interface Name CLI: SourceIPInterfaceName **[NATTranslation_SourceIPInterfaceName]**	Defines the name of the IP interface, as configured in the Interface table.
Target IP Address CLI: TargetIPAddress **[NATTranslation_TargetIPAddress]**	Defines the global IP address. This address is set in the SIP Via and Contact headers as well as in the o= and c= SDP fields.
Source Start Port CLI: SourceStartPort **[NATTranslation_SourceStartPort]**	Defines the optional starting port range (1-65536) of the IP interface, used as matching criteria for this NAT rule. If not configured, the match is done on the entire port range. Only IP addresses and ports of matched source ports will be replaced.
Source End Port CLI: SourceEndPort **[NATTranslation_SourceEndPort]**	Defines the optional ending port range (1-65536) of the IP interface, used as matching criteria for this NAT rule. If not configured, the match is done on the entire port range. Only IP addresses and ports of matched source ports will be replaced.
Target Start Port CLI: TargetStartPort **[NATTranslation_TargetStartPort]**	Defines the optional, starting port range (1-65536) of the global address. If not configured, the ports are not replaced. Matching source ports are replaced with the target ports. This address is set in the SIP Via and Contact headers, as well as in the o= and c= SDP fields.

Parameter	Description
Target End Port CLI: TargetEndPort **[NATTranslation_TargetEndPort]**	Defines the optional, ending port range (1-65536) of the global address. If not configured, the ports are not replaced. Matching source ports are replaced with the target ports. This address is set in the SIP Via and Contact headers, as well as in the o= and c= SDP fields.

16.8.2 Remote UA behind NAT

16.8.2.1 SIP Signaling Messages

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

If necessary, you can also configure the device to always consider incoming SIP INVITE messages as sent from endpoints that are located behind NAT. When this is enabled, the device sends responses to the INVITE (to the endpoint), using the the source IP address of the packet (INVITE) initially received from the endpoint. This is especially useful in scenarios where the endpoint is located behind a NAT firewall and the device (for whatever reason) is unable to identify NAT using its regular NAT Detection mechanism. This feature is enabled per specific calls using IP Groups. To configure this feature, use the 'Always Use Source Address' parameter in the IP Group table (see "Configuring IP Groups" on page 270). If this feature is disabled, the device's NAT detection is according to the settings of the global parameter, 'SIP NAT Detection' parameter.

16.8.2.2 Media (RTP/RTCP/T.38)

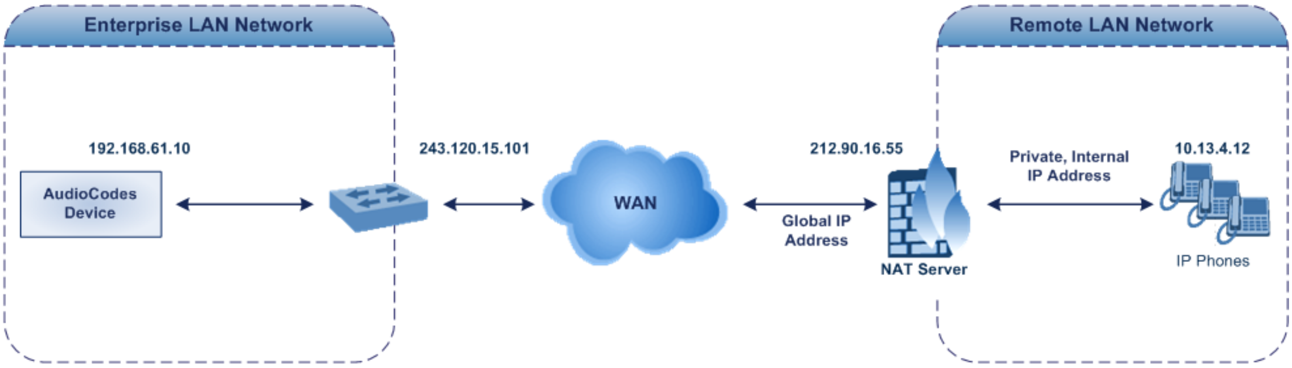
When a remote UA initiates a call and is not located behind a NAT server, the device sends the RTP (or RTCP, T.38) packets to the remote UA using the IP address / UDP port in the SIP message (Contact header). However, if the UA is located behind NAT, the device sends the RTP with the IP address of the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

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

- First Incoming Packet Mechanism - see "First Incoming Packet Mechanism" on page 154
- RTP No-Op packets according to the avt-rtp-noop draft - see "No-Op Packets" on page 155

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

Figure 16-13: Remote UA behind NAT



16.8.2.2.1First Incoming Packet Mechanism

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

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

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

- **Auto-Detect:** NAT is performed only if necessary. If the UA is identified as being located behind NAT, the device sends the media packets to the public IP address:port obtained from the source address of the first media packet received from the UA. Otherwise, the packets are sent using the IP address:port obtained from the first received SIP message. Note also that if the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA, does it determine whether the UA is behind NAT.
- **NAT Is Not Used:** (Default) NAT feature is disabled. The device considers the UA as not located behind NAT and always sends the media packets to the UA using the IP address:port obtained from the first received SIP message.
- **NAT Is Used:** NAT is always performed. The device considers the UA as located behind NAT and always sends the media packets to the UA using the source address obtained from the first media packet received from the UA. In this mode, the device does not send any packets until it receives the first packet from the UA (in order to obtain the IP address).

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

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **Media** > **General Media Settings**).
2. Set the 'NAT Mode parameter to one of the following:
   - [0] Auto-Detect
   - [1] NAT Is Not Used
   - [2] NAT Is Used
3. Click **Submit**.

16.8.2.2.2 No-Op Packets

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

You can control the activation of No-Op packets by using the *ini* file parameter NoOpEnable. If No-Op packet transmission is activated, you can control the time interval in which No-Op packets are sent in the case of silence (i.e., no RTP or T.38 traffic). This is done using the *ini* file parameter NoOpInterval. For a description of the RTP No-Op *ini* file parameters, see "Networking Parameters" on page 778.

- **RTP No-Op:** The RTP No-Op support complies with IETF Internet-Draft draft-wing-avt-rtp-noop-03 ("A No-Op Payload Format for RTP"). This IETF document defines a No-Op payload format for RTP. The draft defines the RTP payload type as dynamic. You can control the payload type with which the No-Op packets are sent. This is performed using the RTPNoOpPayloadType *ini* parameter (see "Networking Parameters" on page 778). The default payload type is 120.
- **T.38 No-Op:** T.38 No-Op packets are sent only while a T.38 session is activated. Sent packets are a duplication of the previously sent frame (including duplication of the sequence number).

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

16.9 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.

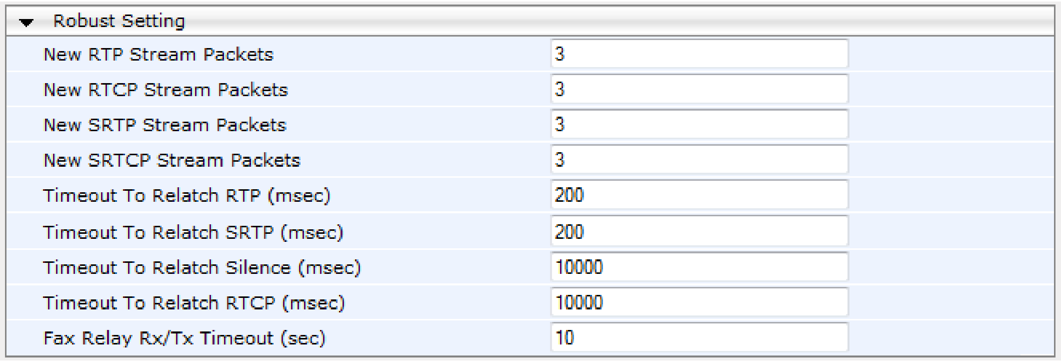
Latching onto a new T.38 stream is reported in CDR using the CDR fields, LatchedT38Ip (new IP address) and LatchedT38Port (new port). In addition, the SIP PUBLISH message updates the latched RTP SSRC, for example:

```
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 16-14: General Settings Page - Robust Setting



▾ Robust Setting

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

3. If you have set the InboundMediaLatchMode parameter to 1 or 2, scroll down to the Robust Settings group and do the following:
   - Define the minimum number of continuous media (RTP, RTCP, SRTP, and SRTCP) packets that need to be received by the channel before it can latch onto this new incoming stream:
     - 'New RTP Stream Packets'
     - 'New RTCP Stream Packets'
     - 'New SRTP Stream Packets'
     - 'New SRTCP Stream Packets'
   - Define a period (msec) during which if no packets are received from the current media session, the channel can re-latch onto another stream:
     - 'Timeout To Relatch RTP'
     - 'Timeout To Relatch SRTP'
     - 'Timeout To Relatch Silence'
     - 'Timeout To Relatch RTCP'
     - 'Fax Relay Rx/Tx Timeout'
4. Click Submit, and then save ("burn") your settings to flash memory.

For a detailed description of the robust media parameters, see "General Security Parameters" on page 794.

16.10 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.

17 Security

This section describes the VoIP security-related configuration.

17.1 Configuring Firewall Settings

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

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

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

Notes:

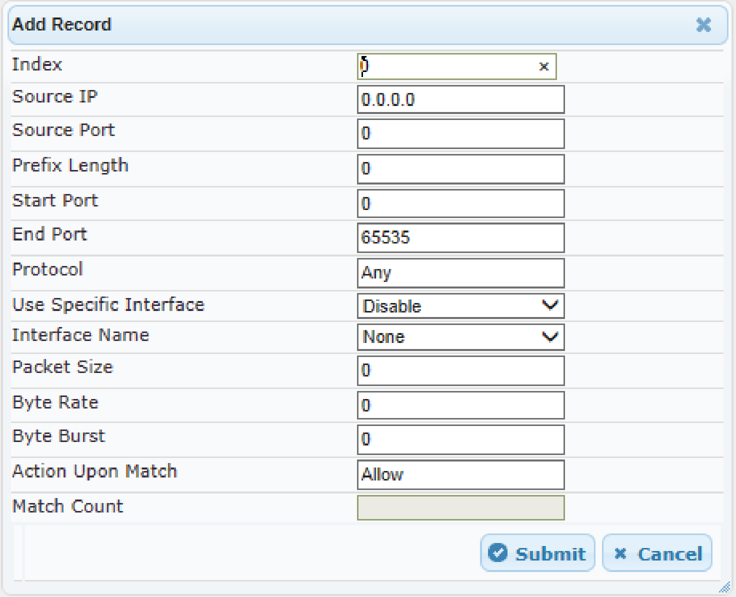
- This firewall applies to a very low-level network layer and overrides all your other security-related configuration. Thus, if you have configured higher-level security features (e.g., on the Application level), you must also configure firewall rules to permit this necessary traffic. For example, if you have configured IP addresses to access the Web and Telnet interfaces in the Web Access List (see "Configuring Web and Telnet Access List" on page 73), 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.
- Setting the 'Prefix Length' field to **0** means that the rule applies to **all** packets, regardless of the defined IP address in the 'Source IP' field. Thus, it is highly recommended to set this parameter to a value other than 0.
- It is recommended to add a rule at the end of your table that blocks all traffic and to add firewall rules above it that allow required traffic (with bandwidth limitations). To block all traffic, use the following firewall rule:
  - √ Source IP: 0.0.0.0
  - √ Prefix Length: 0 (i.e., rule matches all IP addresses)
  - √ Start Port - End Port: 0-65535
  - √ Protocol: **Any**
  - √ Action Upon Match: **Block**
-

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

➢ To configure a Firewall rule:

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

Figure 17-1: Firewall Settings Page - Add Record



Add Record

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

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

Table 17-1: Firewall Settings Table Parameter Descriptions

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

Parameter	Description
Prefix Length CLI: prefixLen **[AccessList_PrefixLen]**	(**Mandatory**) Defines the IP network mask - 32 for a single host or the appropriate value for the source IP addresses. ▪ A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0). ▪ A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0). ▪ A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0). The IP address of the sender of the incoming packet is trimmed in accordance with the prefix length (in bits) and then compared to the parameter 'Source IP'. The default is 0 (i.e., applies to all packets). You **must** change this value to any of the above options. **Note:** A value of 0 applies to **all** packets, regardless of the defined IP address. Therefore, you must set this parameter to a value other than 0.
Start Port CLI: start-port **[AccessList_Start_Port]**	Defines the destination UDP/TCP start port (on this device) to where packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.
End Port CLI: end-port **[AccessList_End_Port]**	Defines the destination UDP/TCP end port (on this device) to where packets are sent. The valid range is 0 to 65535. **Note:** When the protocol type isn't TCP or UDP, the entire range must be provided.
Protocol CLI: protocol **[AccessList_Protocol]**	Defines the protocol type (e.g., UDP, TCP, ICMP, ESP or 'Any') or the IANA protocol number in the range of 0 (Any) to 255. **Note:** This field also accepts the abbreviated strings 'SIP' and 'HTTP'. Specifying these strings implies selection of the TCP or UDP protocols, and the appropriate port numbers as defined on the device.
Use Specific Interface CLI: use-specific-interface **[AccessList_Use_Specific_Interface]**	Determines whether you want to apply the rule to a specific network interface defined in the Interface table (i.e., packets received from that defined in the Source IP field and received on this network interface): ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If enabled, then in the 'Interface Name' field (described below), select the interface to which the rule is applied. ▪ If disabled, then the rule applies to all interfaces.
Interface Name CLI: network-interface-name **[AccessList_Interface_x]**	Defines the network interface to which you want to apply the rule. This is applicable if you enabled the 'Use Specific Interface' field. The list displays interface names as defined in the Interface table in "Configuring IP Network Interfaces" on page 130.
Packet Size CLI: packet-size	Defines the maximum allowed packet size. The valid range is 0 to 65535.

Parameter	Description
[AccessList_Packet_Size]	**Note:** When filtering fragmented IP packets, this field relates to the overall (re-assembled) packet size, and not to the size of each fragment.
Byte Rate CLI: byte-rate **[AccessList_Byte_Rate]**	Defines the expected traffic rate (bytes per second), i.e., the allowed bandwidth for the specified protocol. In addition to this field, the 'Burst Bytes' field provides additional allowance such that momentary bursts of data may utilize more than the defined byte rate, without being interrupted. For example, if 'Byte Rate' is set to 40000 and 'Burst Bytes' to 50000, then this implies the following: the allowed bandwidth is 40000 bytes/sec with extra allowance of 50000 bytes; if, for example, the actual traffic rate is 45000 bytes/sec, then this allowance would be consumed within 10 seconds, after which all traffic exceeding the allocated 40000 bytes/sec is dropped. If the actual traffic rate then slowed to 30000 bytes/sec, then the allowance would be replenished within 5 seconds.
Burst Bytes CLI: byte-burst **[AccessList_Byte_Burst]**	Defines the tolerance of traffic rate limit (number of bytes). The default is 0.
Action Upon Match CLI: allow-type **[AccessList_Allow_Type]**	Defines the firewall action to be performed upon rule match. ▪ "Allow" = (Default) Permits these packets ▪ "Block" = Rejects these packets
Match Count **[AccessList_MatchCount]**	(Read-only) Displays the number of packets accepted or rejected by the rule.

The table below provides an example of configured firewall rules:

Table 17-2: Configuration Example of Firewall Rules

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

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

- **Rules 1 and 2:** Typical firewall rules that allow packets ONLY from specified IP addresses (e.g., proxy servers). Note that the prefix length is configured.

- **Rule 3:** A more "advanced" firewall rule - bandwidth rule for ICMP, which allows a maximum bandwidth of 40,000 bytes/sec with an additional allowance of 50,000 bytes. If, for example, the actual traffic rate is 45,000 bytes/sec, then this allowance would be consumed within 10 seconds, after which all traffic exceeding the allocated 40,000 bytes/sec is dropped. If the actual traffic rate then slowed to 30,000 bytes/sec, the allowance would be replenished within 5 seconds.
- **Rule 4:** Allows traffic from the LAN voice interface and limits bandwidth.
- **Rule 5:** Blocks all other traffic.

17.2 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).

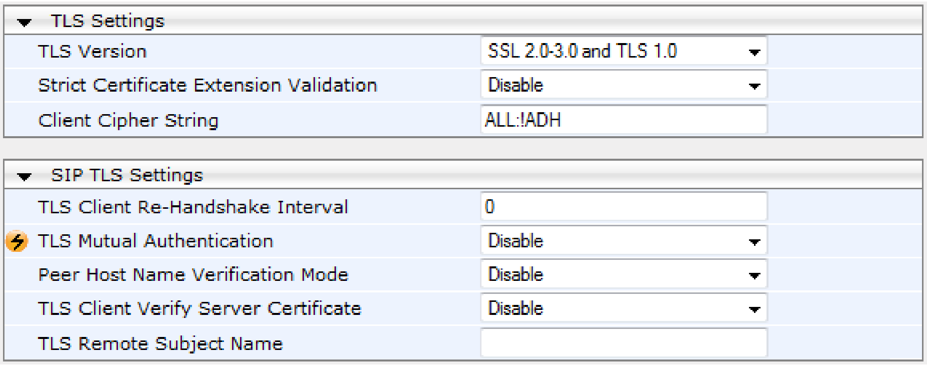
Notes:

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

➢ **To configure SIPS:**

1. Configure a TLS Context as required.
2. Assign the TLS Context to a Proxy Set or SIP Interface (see "Configuring Proxy Sets" on page 282 and "Configuring SIP Interfaces" on page 267, 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 17-2: 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 800.

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**).

17.3 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 171.

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).

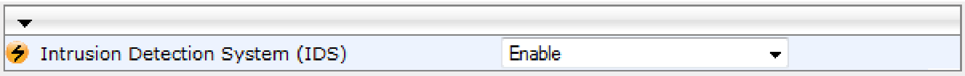
17.3.1 Enabling IDS

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 17-3: Enabling IDS on IDS Global Parameters Page



Intrusion Detection System (IDS) Enable

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

17.3.2 Configuring IDS Policies

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

1. **IDS Policy (parent table):** Defines a name and description for the IDS Policy. You can configure up to 20 IDS Policies.
2. **IDS Rules table (child table):** Defines the actual rules for the IDS Policy. Each IDS Policy can be configured with up to 20 rules.

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

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

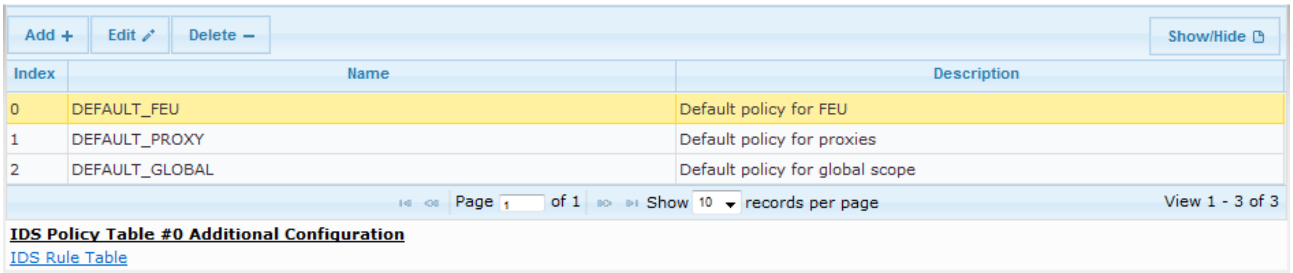
- "DEFAULT_FEU": IDS Policy for far-end users in the WAN
- "DEFAULT_PROXY": IDS Policy for proxy server
- "DEFAULT_GLOBAL": IDS Policy with global thresholds

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

➢ **To configure an IDS Policy:**

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

Figure 17-4: IDS Policy Table with Default Rules



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

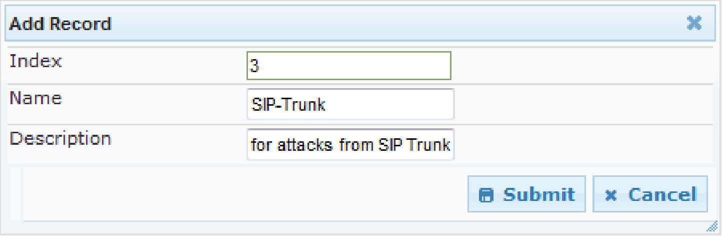
Page 1 of 1 Show 10 records per page View 1 - 3 of 3

IDS Policy Table #0 Additional Configuration

IDS Rule Table

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

Figure 17-5: IDS Policy Table - Add Record



Add Record

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

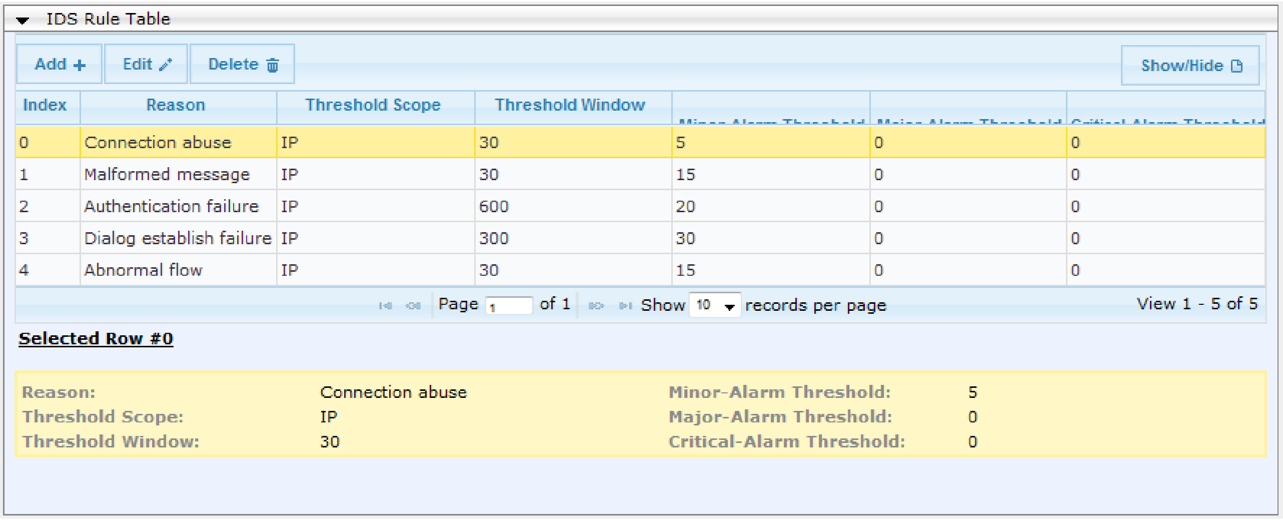
3. Configure an IDS Policy name according to the parameters described in the table below.
4. Click **Submit**.

Table 17-3: IDS Policy Table Parameter Descriptions

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

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

Figure 17-6: 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

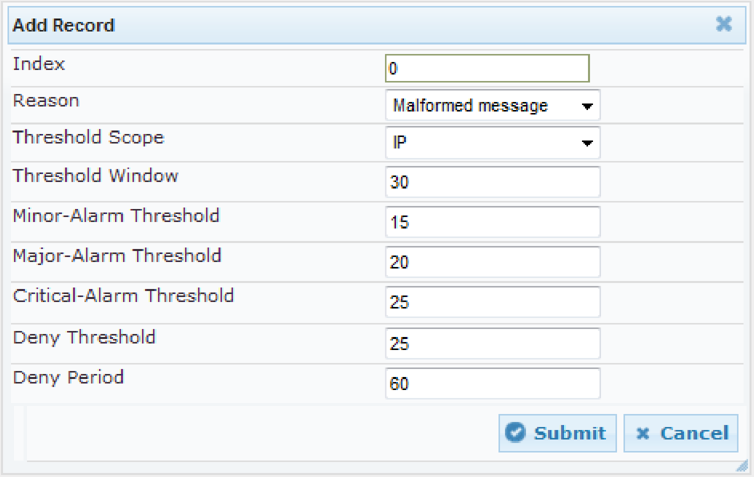
Page 1 of 1 Show 10 records per page View 1 - 5 of 5

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 17-7: IDS Rule Table - Add Record



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

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

7. Configure an IDS Rule according to the parameters described in the table below.
8. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 17-4: IDS Rule Table Parameter Descriptions

Parameter	Description
Index CLI: rule-id **[IDSRule_RuleID]**	Defines an index number for the new table record.
Reason CLI: reason **[IDSRule_Reason]**	Defines the type of intrusion attack (malicious event). ▪ **[0]** Any = All events listed below are considered as attacks and are counted together. ▪ **[1]** Connection abuse (default) = TLS authentication failure. ▪ **[2]** Malformed message = ✓ Message exceeds a user-defined maximum message length (50K) ✓ Any SIP parser error ✓ Message Policy match (see "Configuring SIP Message Policy Rules") ✓ Basic headers not present ✓ Content length header not present (for TCP) ✓ Header overflow ▪ **[3]** Authentication failure = ✓ Local authentication ("Bad digest" errors) ✓ Remote authentication (SIP 401/407 is sent if original message includes authentication) ▪ **[4]** Dialog establish failure = ✓ Classification failure (see "Configuring Classification Rules" on page 545) ✓ Routing failure ✓ Other local rejects (prior to SIP 180 response) ✓ Remote rejects (prior to SIP 180 response)

Parameter	Description
	▪ **[5]** Abnormal flow = ✓ Requests and responses without a matching transaction user (except ACK requests) ✓ Requests and responses without a matching transaction (except ACK requests)
Threshold Scope CLI: threshold-scope **[IDSRule_ThresholdScope]**	Defines the source of the attacker to consider in the device's detection count. ▪ **[0]** Global = All attacks regardless of source are counted together during the threshold window. ▪ **[2]** IP = Attacks from each specific IP address are counted separately during the threshold window. ▪ **[3]** IP+Port = Attacks from each specific IP address:port are counted separately during the threshold window. This option is useful for NAT servers, where numerous remote machines use the same IP address but different ports. However, it is not recommended to use this option as it may degrade detection capabilities.
Threshold Window CLI: threshold-window **[IDSRule_ThresholdWindow]**	Defines the threshold interval (in seconds) during which the device counts the attacks to check if a threshold is crossed. The counter is automatically reset at the end of the interval. The valid range is 1 to 1,000,000. The default is 1.
Minor-Alarm Threshold CLI: minor-alrm-thr **[IDSRule_MinorAlarmThreshold]**	Defines the threshold that if crossed a minor severity alarm is sent. The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.
Major-Alarm Threshold CLI: major-alrm-thr **[IDSRule_MajorAlarmThreshold]**	Defines the threshold that if crossed a major severity alarm is sent. The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.
Critical-Alarm Threshold CLI: critical-alrm-thr **[IDSRule_CriticalAlarmThreshold]**	Defines the threshold that if crossed a critical severity alarm is sent. The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.
Deny Threshold **[IDSRule_DenyThreshold]**	Defines the threshold that if crossed, the device blocks (blacklists) the remote host (attacker). The default is -1 (i.e., not configured). **Note:** This parameter is applicable only if the 'Threshold Scope' parameter is set to **IP** or **IP+Port**.
Deny Period **[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).

17.3.3 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:

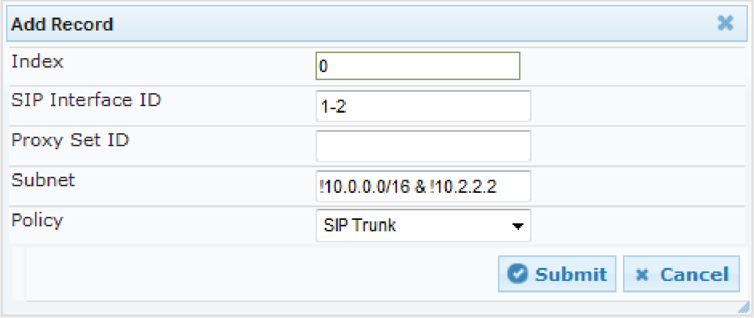
- **SIP Interface:** For detection of malicious attacks on specific SIP Interface(s). For configuring SIP Interfaces, see "Configuring SIP Interfaces" on page 267.
- **Proxy Sets:** For detection of malicious attacks from specified Proxy Set(s). For configuring Proxy Sets, see "Configuring Proxy Sets" on page 282.
- **Subnet addresses:** For detection of malicious attacks from specified subnet addresses.

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

➢ **To configure an IDS Policy-Matching rule:**

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

Figure 17-8: IDS Match Table - Add Record



Add Record

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

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

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

Table 17-5: IDS Match Table Parameter Descriptions

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

Parameter	Description
Proxy Set ID CLI: 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 CLI: subnet **[IDSMatch_Subnet]**	Defines the subnet to which the IDS Policy is assigned. This indicates the subnets from where the attacks are coming from. The following syntax can be used: ▪ Basic syntax is a subnet in CIDR notation (e.g., 10.1.0.0/16 means all sources with IP address in the range 10.1.0.0–10.1.255.255) ▪ An IP address can be specified without the prefix length to refer to the specific IP address. ▪ Each subnet can be negated by prefixing it with "!", which means all IP addresses outside that subnet. ▪ Multiple subnets can be specified by separating them with "&" (and) or "|" (or) operations. For example: ✓ 10.1.0.0/16 | 10.2.2.2: includes subnet 10.1.0.0/16 and IP address 10.2.2.2. ✓ !10.1.0.0/16 & !10.2.2.2: includes all addresses except those of subnet 10.1.0.0/16 and IP address 10.2.2.2. Note that the exclamation mark "!" appears before each subnet. ✓ 10.1.0.0/16 & !10.1.1.1: includes subnet 10.1.0.0/16, except IP address 10.1.1.1.
Policy CLI: policy **[IDSMatch_Policy]**	Assigns an IDS Policy (configured in "Configuring IDS Policies" on page 165).

17.3.4 Viewing IDS Alarms

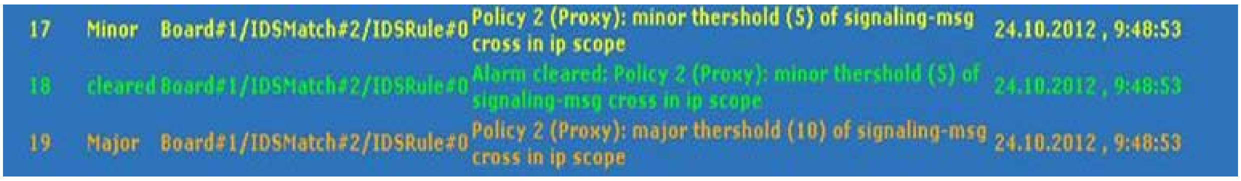
For the IDS feature, the device sends the following SNMP traps:

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

    If the severity level is raised, the alarm of the former severity is cleared and the device sends a new alarm with the new severity. The alarm is cleared after a user-defined period (configured by the ini file parameter, IDSAlarmClearPeriod) during which no thresholds have been crossed. However, this "quiet" period must be at least twice the 'Threshold Window' value (configured in "Configuring IDS Policies" on page 165). For example, if you set IDSAlarmClearPeriod to 20 sec and 'Threshold Window' to 15 sec, the IDSAlarmClearPeriod parameter is ignored and the alarm is cleared only after 30 seconds (2 x 15 sec).

    The figure below displays an example of IDS alarms in the Active Alarms table ("Viewing Active Alarms" on page 665). In this example, a Minor threshold alarm is cleared and replaced by a Major threshold alarm:

Figure 17-9: IDS Alarms in Active Alarms Table



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

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

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

- To view all active IDS alarms:

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

- To view all IP addresses that 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
```

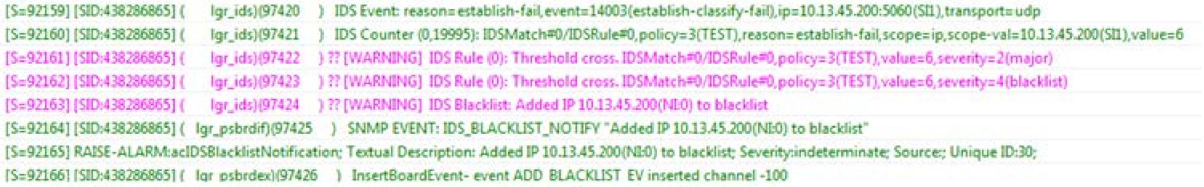
For example:

```
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 719). An example of a Syslog message with IDS alarms and notifications is shown below:

Figure 17-10: 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(SI1),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(SI1),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_psbrdex)(97426     )  InsertBoardEvent- event ADD_BLACKLIST_EV inserted channel -100
```

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

Table 17-6: 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

18 Media

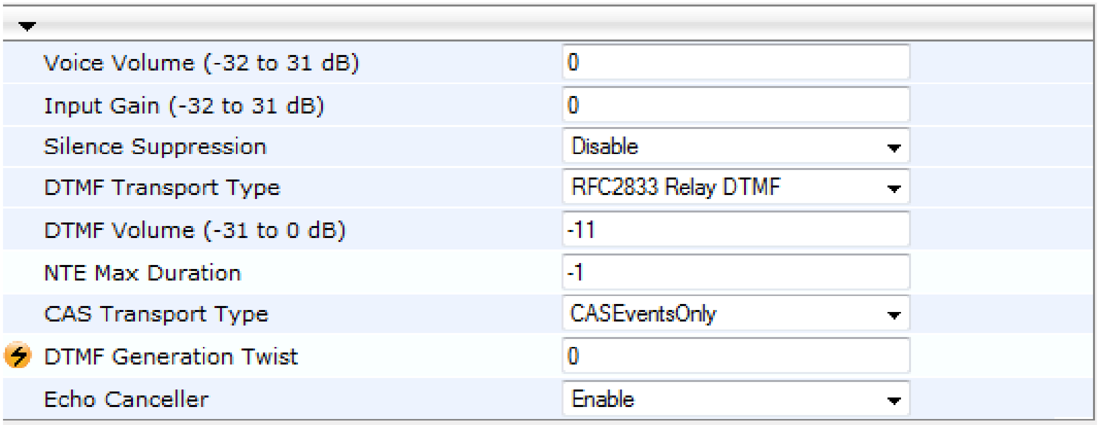
This section describes the media-related configuration.

18.1 Configuring Voice Settings

The Voice Settings page configures various voice parameters such as voice volume, silence suppression, and DTMF transport type. For a detailed description of these parameters, see "Configuration Parameters Reference" on page 763.

➢ **To configure the voice parameters:**

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



Voice Volume (-32 to 31 dB)	0
Input Gain (-32 to 31 dB)	0
Silence Suppression	Disable
DTMF Transport Type	RFC2833 Relay DTMF
DTMF Volume (-31 to 0 dB)	-11
NTE Max Duration	-1
CAS Transport Type	CASEventsOnly
⚡ DTMF Generation Twist	0
Echo Canceller	Enable

2. Configure the Voice parameters as required.
3. Click Submi**t**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

18.1.1 Configuring Voice Gain (Volume) Control

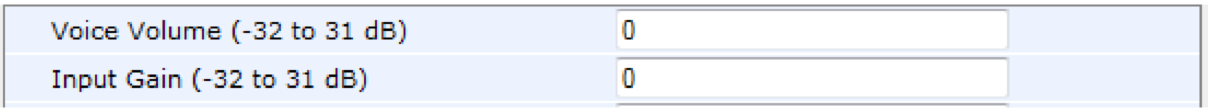
The device allows you to configure the level of the received (input gain) Tel-to-IP signal and the level of the transmitted (output gain) IP-to-Tel signal. The gain can be set between -32 and 31 decibels (dB).

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 18-1: 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**.

18.1.2 Silence Suppression (Compression)

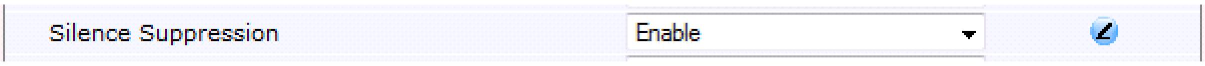
Silence suppression (compression) is a method for conserving bandwidth on VoIP calls by not sending packets when silence is detected. The device uses its VAD feature to detect periods of silence in the voice channel during an established call. When silence is detected, it stops sending packets in the channel.

The following procedure describes how to enable silence suppression using the Web interface.

➢ **To enable silence suppression using the Web interface:**

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

Figure 18-2: Enabling Silence Suppression in Voice Settings Page



Silence Suppression: Enable

2. Set the 'Silence Suppression' (*EnableSilenceCompression*) field to **Enable**.
3. Click **Submit**.

18.1.3 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.

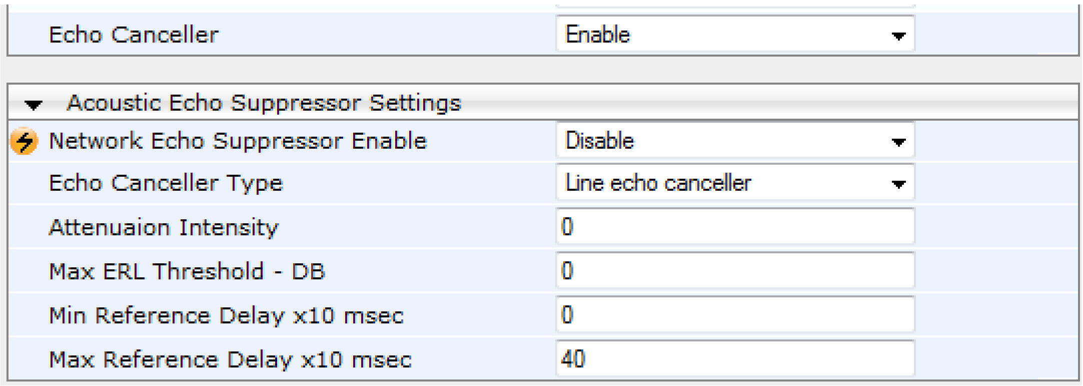
The device also supports acoustic echo cancellation for SBC calls. These echoes are composed of undesirable acoustical reflections (non-linear) of the received signal (i.e., from the speaker) which find their way from multiple reflections such as walls and windows into the transmitted signal (i.e., microphone). 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 Canceler, 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

18.2 Fax and Modem Capabilities

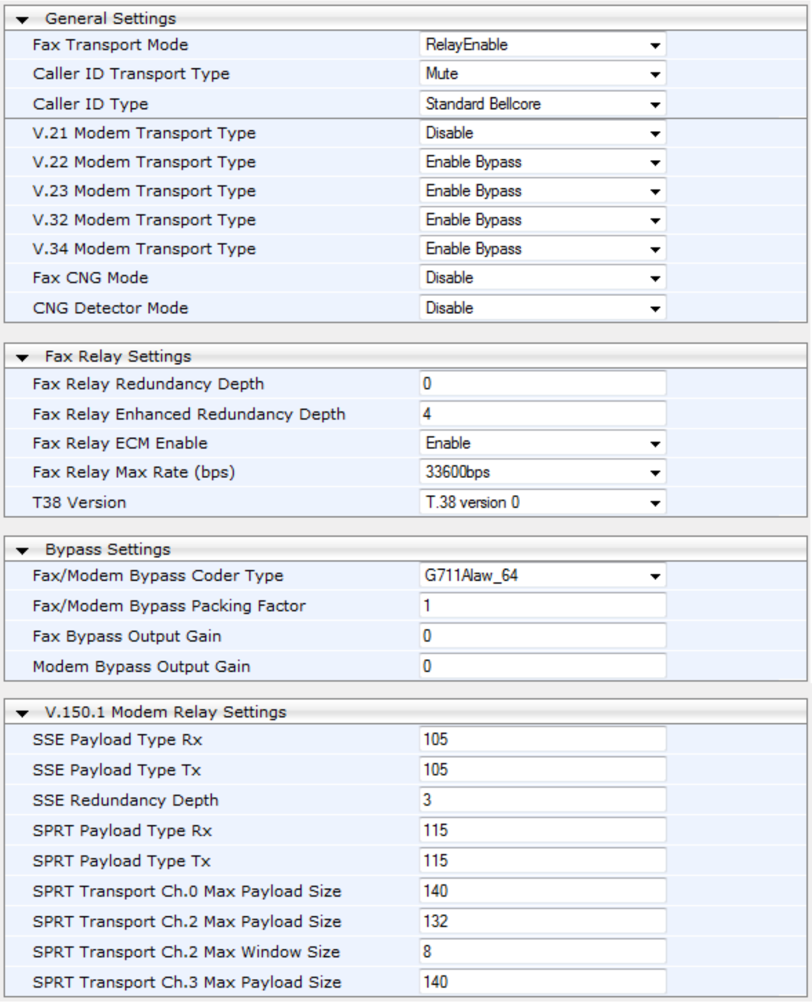
This section describes the device's fax and modem capabilities and corresponding configuration. The fax and modem configuration is done in the Fax/Modem/CID Settings page.

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 189.
- For a detailed description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 763.

➢ To access the fax and modem parameters:

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



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
T38 Version	T.38 version 0
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
V.150.1 Modem Relay Settings	
SSE Payload Type Rx	105
SSE Payload Type Tx	105
SSE Redundancy Depth	3
SPRT Payload Type Rx	115
SPRT Payload Type Tx	115
SPRT Transport Ch.0 Max Payload Size	140
SPRT Transport Ch.2 Max Payload Size	132
SPRT Transport Ch.2 Max Window Size	8
SPRT Transport Ch.3 Max Payload Size	140

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

18.2.1 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 189): 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).

18.2.2 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 177)
- G.711 Transport: switching to G.711 when fax/modem is detected (see "G.711 Fax / Modem Transport Mode" on page 180)
- Fax fallback to G.711 if T.38 is not supported (see "Fax Fallback" on page 180)
- Fax and modem bypass: a proprietary method that uses a high bit rate coder (see "Fax/Modem Bypass Mode" on page 181)
- NSE Cisco’s Pass-through bypass mode for fax and modem (see "Fax / Modem NSE Mode" on page 182)
- Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (see "Fax / Modem Transparent with Events Mode" on page 183)
- Transparent: passing the fax / modem signal in the current voice coder (see "Fax / Modem Transparent Mode" on page 183)
- RFC 2833 ANS Report upon Fax/Modem Detection (see "RFC 2833 ANS Report upon Fax/Modem Detection" on page 184)

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

18.2.2.1 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 178)
- 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 178)

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). This 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.

18.2.2.1.1 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).

18.2.2.1.2 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)

18.2.2.1.3 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=ftmp' 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=ftmp' 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.

18.2.2.2 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 ‘gpmd’ attribute is added to the SDP according to the following format:

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

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

- **For G.711 µ-law:**

  ```
  a=gpmd: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).

18.2.2.3 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 ‘gpmd’ attribute is added to the SDP according to the following format:

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

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

- **For G.711 µ-law:**

  ```
  a=gpmd: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).

18.2.2.4 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).
- FaxModemBypasDJBufMinDelay (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.

18.2.2.5 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.

18.2.2.6 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).

18.2.2.7 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 307) 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. 'Silence Suppression' (EnableSilenceCompression) - Voice Settings page (**Configuration** tab > **VoIP** menu > **Media** > **Voice Settings**).
d. 'Echo Canceller' (EnableEchoCanceller) - 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 181) or Transparent with Events modes (see "Fax / Modem Transparent with Events Mode" on page 183) for modem.

18.2.2.8 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. This 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).

18.2.3 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:

- T38 Version 3 - V.34 fax relay mode
- Bypass mechanism for V.34 fax transmission (see "Bypass Mechanism for V.34 Fax Transmission" on page 185)
- T38 Version 0 relay mode, i.e., fallback to T.38 (see "Relay Mode for T.30 and V.34

Faxes" on page 186)

To configure whether to pass V.34 over T38 fax relay, or use Bypass over the High Bit Rate coder (e.g. PCM A-Law), use the 'V.34 Fax Transport Type' parameter (V34FaxTransportType).

You can use the 'SIP T.38 Version' parameter (SIPT38Version) in the Advanced Parameters page (Configuration tab > VoIP menu > SIP Definitions > Advanced Parameters) to configure one of the following:

- Pass V.34 over T.38 fax relay using bit rates of up to 33,600 bps ('SIP T.38 Version' is set to Version 3).
- Use Fax-over-T.38 fallback to T.30, using up to 14,400 bps ('SIP T.38 Version' is set to Version 0).

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**.

18.2.3.1 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).
2. Set the ini file parameter, V34FaxTransportType to 2 (Bypass).

➢ **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).
2. Set the ini file parameter, V34FaxTransportType to 2 (Bypass).

18.2.3.2 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 **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).
2. Set the ini file parameter, V34FaxTransportType to 1 (Relay).

➢ **To allow V.34 fax relay over T.38:**

- In the Advanced Parameters page (Configuration tab > VoIP menu > SIP Definitions > Advanced Parameters), set the 'SIP T.38 Version' parameter to Version 3 (SIPT38Version = 3).

➢ **To force V.34 fax machines to use their backward compatibility with T.30 faxes and operate in the slower T.30 mode:**

- Set the 'SIP T.38 Version' parameter to Version 0 (SIPT38Version = 0).

18.2.3.3 V.34 Fax Relay for SG3 Fax Machines

Super Group 3 (SG3) is a standard for fax machines that support speeds of up to 33.6 kbps through V.34 half duplex (HD) modulation. The following procedure describes how to configure V.34 (SG3) fax relay support based on ITU Specification T.38 version 3.

➢ **To enable support for V.34 fax relay (T.38) at SG3 speed:**

1. In the IP Profile table (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **IP Profile Settings**), configure an IP Profile with the 'Fax Signaling Method' parameter (IpProfile_IsFaxUsed) set to **T.38 Relay**.
2. In the Coders Table (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**), set the coder used by the device to G.729 (or any other supported codec).
3. On the Fax/Modem/CID Settings page, do the following settings:
   a. 'SIP T.38 Version' to **Version 3** (SIPT38Version = 3).
   b. 'Fax Relay Max Rate' (RelayMaxRate) to **33,600bps** (default).
   c. 'CNG Detector Mode' (CNGDetectorMode) to **Disable** (default).
   d. 'V.21 Modem Transport Type' to **Disable** (V21ModemTransportType = 0).
   e. 'V.22 Modem Transport Type' to **Disable** (V22ModemTransportType = 0).
   f. 'V.23 Modem Transport Type' to **Disable** (V23ModemTransportType = 0).
   g. 'V.32 Modem Transport Type' to **Disable** (V32ModemTransportType = 0).
   h. 'V.34 Modem Transport Type' to **Disable** (V34ModemTransportType = 0).
   i. 'CED Transfer Mode' to Fax Relay or VBD (CEDTransferMode = 0).
4. Set the ini file parameter, V34FaxTransportType to 1 (i.e., relay).
5. Set the ini file parameter, T38MaxDatagramSize to 560 (default).

Notes:

- The T.38 negotiation should be completed at call start according to V.152 procedure (as shown in the INVITE example below).
- T.38 mid-call Re-INVITEs are supported.
- If the remote party supports only T.38 Version 0, the device "downgrades" the T.38 Version 3 to T.38 Version 0.

For example, the device sends or receives the following INVITE message, negotiating both audio and image media:

```
INVITE sip:2001@10.8.211.250;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.8.6.55;branch=z9hG4bKac1938966220
Max-Forwards: 70
From: <sip:318@10.8.6.55>;tag=1c1938956155
To: <sip:2001@10.8.211.250;user=phone>
Call-ID: 193895529241200022331@10.8.6.55
CSeq: 1 INVITE
Contact: <sip:318@10.8.6.55:5060>
Supported: em,100rel,timer,replaces,path,resource-priority,sdp-
anat
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
Remote-Party-ID:
<sip:318@10.8.211.250>;party=calling;privacy=off;screen=no;screen-
ind=0;npi=1;ton=0
Remote-Party-ID: <sip:2001@10.8.211.250>;party=called;npi=1;ton=0
User-Agent: Audiocodes-Sip-Gateway-/v.6.80A.227.005
Content-Type: application/sdp
Content-Length: 433
v=0
o=AudiocodesGW 1938931006 1938930708 IN IP4 10.8.6.55
s=Phone-Call
c=IN IP4 10.8.6.55
t=0 0
m=audio 6010 RTP/AVP 18 97
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
a=rtpmap:97 telephone-event/8000
a=fmtp:97 0-15
a=ptime:20
a=sendrecv
m=image 6012 udptl t38
a=T38FaxVersion:3
a=T38MaxBitRate:33600
a=T38FaxMaxBuffer:1024
a=T38FaxMaxDatagram:122
a=T38FaxRateManagement:transferredTCF
a=T38FaxUdpEC:t38UDPRedundancy
```

18.2.4 V.150.1 Modem Relay

The device can be configured to transfer modem calls using a subset of the ITU-T V.150.1 Modem Relay protocol. The device also supports V.150.1 modem relay coder negotiation in the initial SIP INVITE and 200 OK, using the SDP body according to the USA Department of Defense (DoD) UCR-2008, Change 2 specification. This eliminates the need for sending a re-INVITE to negotiate V.150.1. The device sends an INVITE's SDP offer in a format to negotiate V.150 modem relay using the same port as RTP, as shown in the example below:

```
a=cdsc:1 audio udpsprt 114\r\n
a=cpar:a=sprtmap:114 v150mr/8000\r\n
a=cpar:a=fmtp:114
mr=1;mg=0;CDSCselect=1;mrmods=1,3;jmdelay=no;versn=1.1\r\n\
```

You can configure the payload type for the outgoing SDP offer, using the NoAudioPayloadType parameter. You can set this parameter to "NoAudio", whereby RTP is not sent and the device adds an audio media only for the Modem Relay purpose. This is also in accordance to DOD UCR 2008 specification: "The AS-SIP signaling appliance MUST advertise the “NoAudio” payload type to interoperate with a “Modem Relay-Preferred” endpoint that immediately transitions to the Modem Relay state without first transmitting voice information in the Audio state."

Notes:

- The V.150.1 Modem Relay 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 621.
- The V.150.1 Modem Relay feature support is a subset of the full V.150.1 protocol and is designed according to the US DoD requirement document. It therefore, cannot be used for general purposes.
- The V.150.1 feature has been tested with certain IP phones. For more details, please contact your AudioCodes sales representative.
- The V.150.1 SSE Tx payload type is according to the offered SDP of the remote side.
- The V.150.1 SPRT Rx payload type is according to the 'Payload Type' field in the Coders table.
- The V.150.1 SPRT Tx payload type is according to the remote side offered SDP.

➢ **To configure V.150.1 Modem relay:**

1. In the Coders Table (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**), set the coder to **V.150**.
2. On the Fax/Modem/CID Settings page, configure the V.150.1 parameters appearing under the 'V.150.1 Modem Relay Settings' group:
   a. Set the 'SSE Payload Type Rx' parameter to the V.150.1 SSE payload type that the device uses when it offers the SDP.
   b. Set the 'SSE Redundancy Depth' parameter to the number of sent SSE redundant packets. This parameter is important in case of network impairments.
   c. For additional V.150.1 related parameters, see "Fax and Modem Parameters" on page 861.

18.2.5 Simultaneous Negotiation of Fax (T.38) and Modem (V.150.1) Relay

The device can negotiate fax relay (T.38) and modem relay (V.150.1) sessions in the same, already established call channel. Fax relay sessions require bypass answering tone (CED) while modem relay requires RFC 2833 answering tone. As the device is not always aware at the start of the session whether the answering tone is fax or modem, it uses both methods for CED tone transfer and sends both answering tone types. Only when the answering tone is detected, does the device send the fax or modem.

To support this functionality, you need to configure a Coders Group (in the Coders Group table - see "Configuring Coder Groups" on page 310) that includes the T.38, V.150, and G.711/VBD coders.

18.2.6 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 307).

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 <IPAdressA>
s=-
t=0 0
p=+1
c=IN IP4  <IPAddressA
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.

18.2.7 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.

18.3 Configuring RTP/RTCP Settings

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

18.3.1 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.

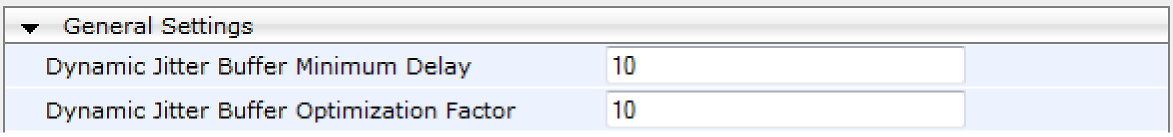
Fax and modem devices are sensitive to small packet losses or to added BFI packets. Therefore, to achieve better performance during modem and fax calls, the Optimization Factor should be set to 13. In this special mode the clock drift correction is performed less frequently - only when the Jitter Buffer 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 18-3: 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**.

18.3.2 Comfort Noise Generation

The device can generate artificial background noise, called *comfort* noise, in the voice channel during periods of silence (i.e. when no call party is speaking) for Gateway calls. This is useful in that it reassures the call parties that the call is still connected. The device detects silence using its Voice Activity Detection (VAD) mechanism. When the Calling Tone (CNG) is enabled and silence is detected, the device transmits Silence Identifier Descriptors (SIDs) parameters to reproduce the local background noise at the remote (receiving) side.

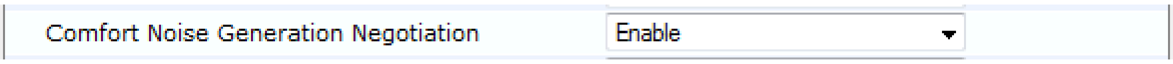
The Comfort Noise Generation (CNG) support also depends on the silence suppression (SCE) setting for the coder used in the voice channel. For more information, see the description of the CNG-related parameters.

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

➢ **To configure CNG 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 18-4: Comfort Noise Parameter in RTP/RTCP Settings Page



Comfort Noise Generation Negotiation: Enable

2. Set the 'Comfort Noise Generation Negotiation' parameter (ComfortNoiseNegotiation) to **Enable**.

3. Click **Submit**.

> **Note:** This feature is applicable only to the Gateway application.

18.3.3 Dual-Tone Multi-Frequency Signaling

This section describes the configuration of Dual-Tone Multi-Frequency (DTMF) signaling.

18.3.3.1 Configuring DTMF Transport Types

The device supports various methods for transporting DTMF digits over the IP network to the remote endpoint. These methods and their configuration are configured in the DTMF & Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **DTMF & Dialing**):

- **Using INFO message according to Nortel IETF draft:** DTMF digits are sent to the remote side in INFO messages. To enable this mode, define the following:
  a. Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).
  b. Set the '1st Tx DTMF Option' parameter to **INFO (Nortel)** (TxDTMFOption = 1).

  Note: In this mode, DTMF digits are removed from the audio stream (and the 'DTMF Transport Type' parameter is automatically set to **Mute DTMF**).
- **Using INFO message according to Cisco's mode:** DTMF digits are sent to the remote side in INFO messages. To enable this mode, define the following:
  a. Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).
  b. Set the '1st Tx DTMF Option' parameter to **INFO (Cisco)** (TxDTMFOption = 3).

  Note: In this mode, DTMF digits are removed from the audio stream (and the 'DTMF Transport Type' parameter is automatically set to **Mute DTMF**).
- **Using NOTIFY messages according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01**: DTMF digits are sent to the remote side using NOTIFY messages. To enable this mode, define the following:
  a. Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).
  b. Set the '1st Tx DTMF Option' parameter to **NOTIFY** (TxDTMFOption = 2).

  Note: In this mode, DTMF digits are removed from the audio stream (and the 'DTMF Transport Type' parameter is automatically set to **Mute DTMF**).
- **Using RFC 2833 relay with Payload type negotiation:** DTMF digits are sent to the remote side as part of the RTP stream according to RFC 2833. To enable this mode, define the following:
  a. Set the 'Declare RFC 2833 in SDP' parameter to **Yes** (RxDTMFOption = 3).
  b. Set the '1st Tx DTMF Option' parameter to **RFC 2833** (TxDTMFOption = 4).

  Note: To set the RFC 2833 payload type with a value other than its default, use the RFC2833PayloadType parameter. The device negotiates the RFC 2833 payload type using local and remote SDP and sends packets using the payload type from the received SDP. The device expects to receive RFC 2833 packets with the same payload type as configured by this parameter. If the remote side doesn't include 'telephony-event' in its SDP, the device sends DTMF digits in transparent mode (as part of the voice stream).
- **Sending DTMF digits (in RTP packets) as part of the audio stream (DTMF Relay is disabled):** This method is typically used with G.711 coders. With other low-bit rate (LBR) coders, the quality of the DTMF digits is reduced. To enable this mode, define

the following:

a. Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).

b. Set the '1st Tx DTMF Option' parameter to **Not Supported** (TxDTMFOption = 0).

c. Set the ini file parameter, DTMFTransportType to 2 (i.e., transparent).

- **Using INFO message according to Korea mode:** DTMF digits are sent to the remote side in INFO messages. To enable this mode, define the following:

  a. Set the 'Declare RFC 2833 in SDP' parameter to **No** (RxDTMFOption = 0).

  b. Set the '1st Tx DTMF Option' parameter to **INFO (Cisco)** (TxDTMFOption = 3).

  Note: In this mode, DTMF digits are removed from the audio stream (and the 'DTMF Transport Type' parameter is automatically set to **Mute DTMF**).

Notes:

- The device is always ready to receive DTMF packets over IP in all possible transport modes: INFO messages, NOTIFY, and RFC 2833 (in proper payload type) or as part of the audio stream.
- To exclude RFC 2833 Telephony event parameter from the device's SDP, set the 'Declare RFC 2833 in SDP' parameter to **No**.

The following parameters affect the way the device handles the DTMF digits:

- TxDTMFOption, RxDTMFOption, RFC2833TxPayloadType, and RFC2833RxPayloadType
- MGCPDTMFDetectionPoint, DTMFVolume, DTMFTransportType, DTMFDigitLength, and DTMFInterDigitInterval

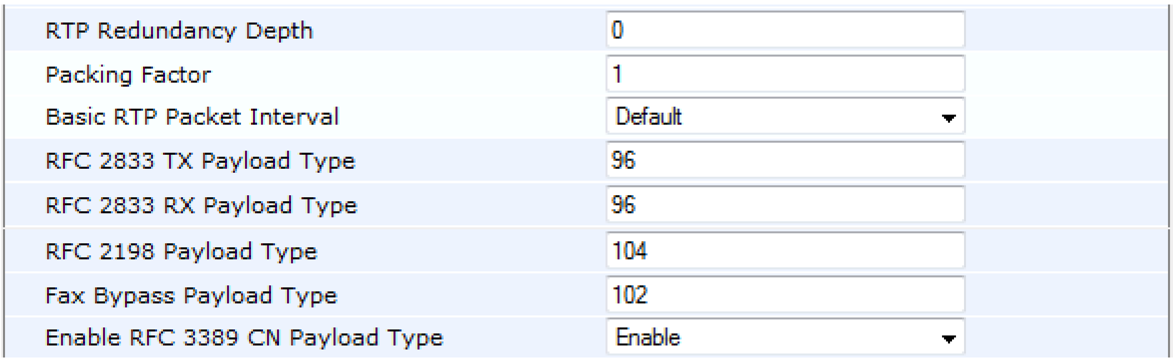
18.3.3.2 Configuring RFC 2833 Payload

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 18-5: RFC 2833 Payload Parameters in RTP/RTCP Settings Page



RTP Redundancy Depth	0
Packing Factor	1
Basic RTP Packet Interval	Default
RFC 2833 TX Payload Type	96
RFC 2833 RX Payload Type	96
RFC 2198 Payload Type	104
Fax Bypass Payload Type	102
Enable RFC 3389 CN Payload Type	Enable

2. Configure the following parameters:

   - 'RTP Redundancy Depth' (RTPRedundancyDepth) - enables the device to generate RFC 2198 redundant packets.

- 'Enable RTP Redundancy Negotiation' (EnableRTPRedundancyNegotiation) - enables the device to include the RTP redundancy dynamic payload type in the SDP, according to RFC 2198.
- '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**.

18.3.4 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 10 (default). 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).

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

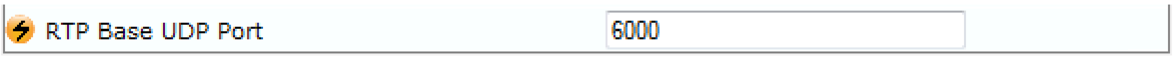
You can also configure specific port ranges per logical media-type IP network interface, using Media Realms (see "Configuring Media Realms" on page 259). However, the port range configured for the Media Realm must be within the range configured by the BaseUDPPort parameter.

The following procedure describes how to configure the RTP base UDP port in the Web interface.

➢ To configure the RTP base UDP port:

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

Figure 18-6: RTP Based UDP Port in RTP/RTCP Settings Page



RTP Base UDP Port: 6000

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

> **Note:** You can also configure the base UDP port for calls belonging to a specific SIP entity (IP Group), using IP Profiles (see "Configuring IP Profiles" on page 314).

18.4 Configuring IP Media Settings

This section describes the configuration of various IP media features.

18.4.1 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 or Tel, 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.

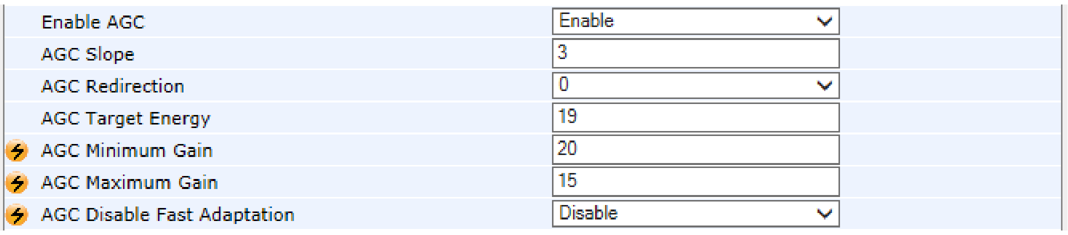
> **Note:** The AGC feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 621.

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 18-7: 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. This parameter can either be the global parameter or per IP Profile.
4. Click **Submit**.

18.5 Configuring Various Codec Attributes

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

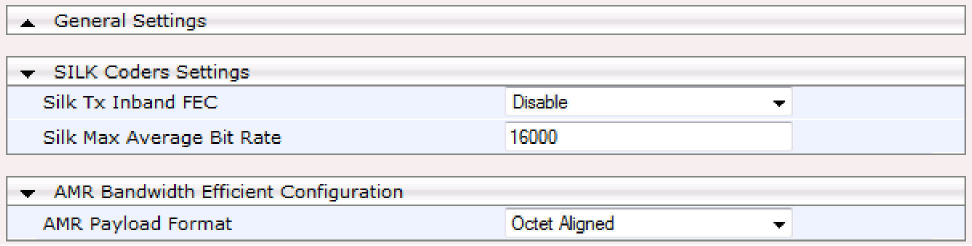
- AMR coder:
  - 'Payload Format': Defines the AMR payload format type.
- SILK coder (Skype's default audio codec):
  - 'Silk Tx Inband FEC': Enables forward error correction (FEC) for the SILK coder.
  - 'Silk Max Average Bit Rate': Defines the maximum average bit rate for the SILK coder.

For a detailed description of these parameters and for additional codec parameters, see "Coder Parameters" on page 854.

➢ To configure codec attributes:

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

Figure 18-8: Codec Settings in General Media Settings Page



Silk Tx Inband FEC: Disable
Silk Max Average Bit Rate: 16000

AMR Payload Format: Octet Aligned

2. Configure the parameters as required, and then click click **Submit**.
3. To save the changes to flash memory, see "Saving Configuration" on page 592.

18.6 Configuring Analog Settings

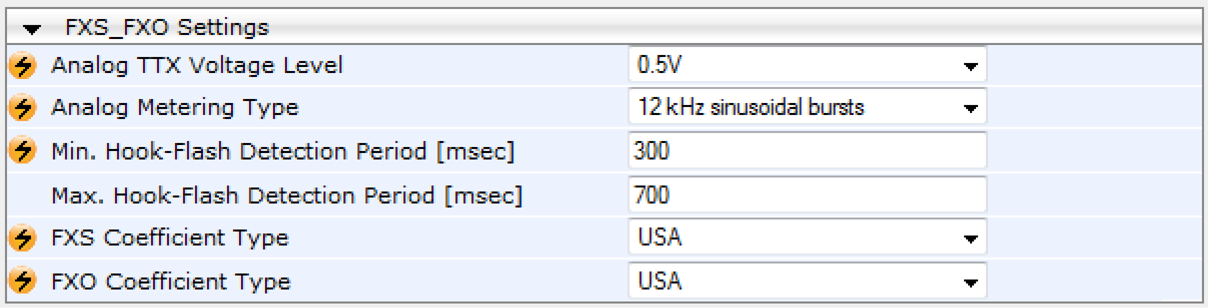
The Analog Settings page allows you to configure various analog parameters. For a detailed description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 763.

This page also selects the type (USA or Europe) of FXS and/or FXO coefficient information. The FXS coefficient contains the analog telephony interface characteristics such as DC and AC impedance, feeding current, and ringing voltage.

➢ **To configure the analog parameters:**

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

Figure 18-9: Analog Settings Page



FXS_FXO Settings

Analog TTX Voltage Level	0.5V
Analog Metering Type	12 kHz sinusoidal bursts
Min. Hook-Flash Detection Period [msec]	300
Max. Hook-Flash Detection Period [msec]	700
FXS Coefficient Type	USA
FXO Coefficient Type	USA

2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

18.7 Configuring Media (SRTP) Security

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

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

- AES_CM_128_HMAC_SHA1_32
- AES_CM_128_HMAC_SHA1_80
- ARIA_CM_128_HMAC_SHA1_80
- ARIA_CM_192_HMAC_SHA1_80

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

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

- UNENCRYPTED_SRTP
- UNENCRYPTED_SRTCP
- UNAUTHENTICATED_SRTP

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

Below is an example of crypto attributes usage:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:PsKoMpHlCg+b5X0YLuSvNrImEh/dAe
```

```
a=crypto:2 AES_CM_128_HMAC_SHA1_32
inline:IsPtLoGkBf9a+c6XVzRuMqHlDnEiAd
```

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

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

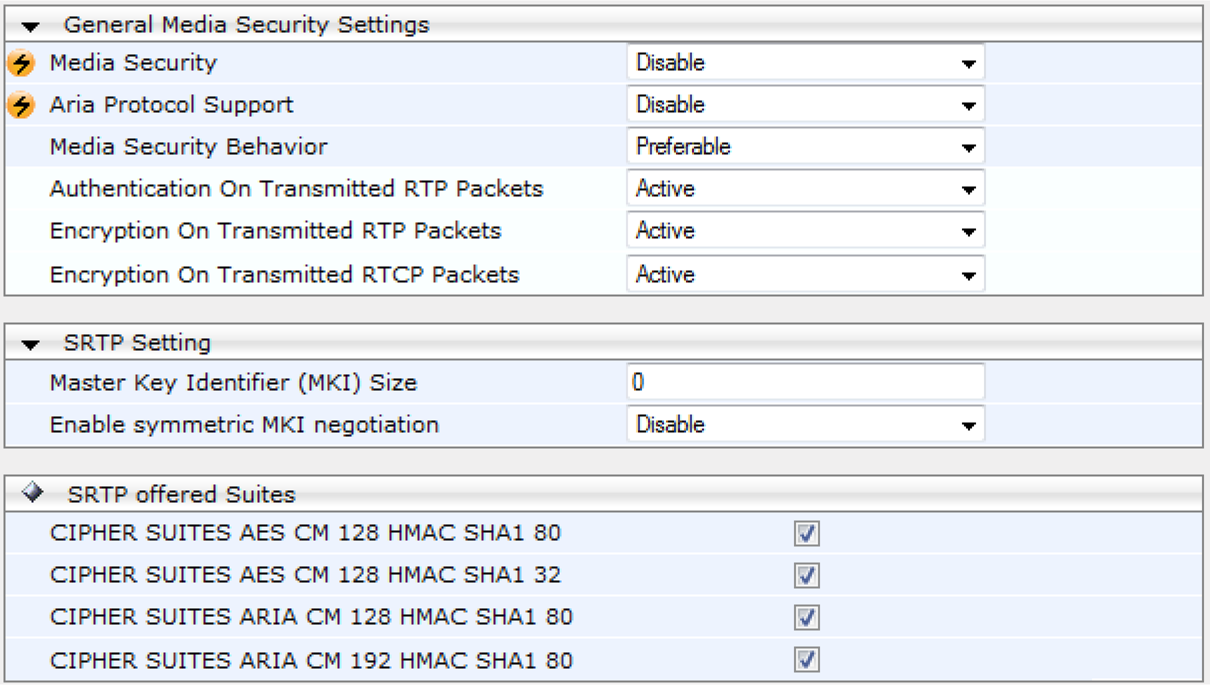
Notes:

- For a detailed description of the SRTP parameters, see "SRTP Parameters" on page 797.
- When SRTP is used, the channel capacity may be reduced.

➢ To enable and configure SRTP:

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

Figure 18-10: 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 Setting

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

SRTP offered Suites

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

2. Set the 'Media Security' parameter to **Enable** to enable SRTP.
3. Configure the other SRTP parameters as required.
4. Click **Submit**.
5. To save the changes to flash memory, see "Saving Configuration" on page 592.

19 Services

This section describes configuration for various supported services.

19.1 SIP-based Media Recording

The device can record SIP-based media (call sessions) traversing it. This applies only to SBC calls. 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 o(draft-ietf-siprec-architecture-03).

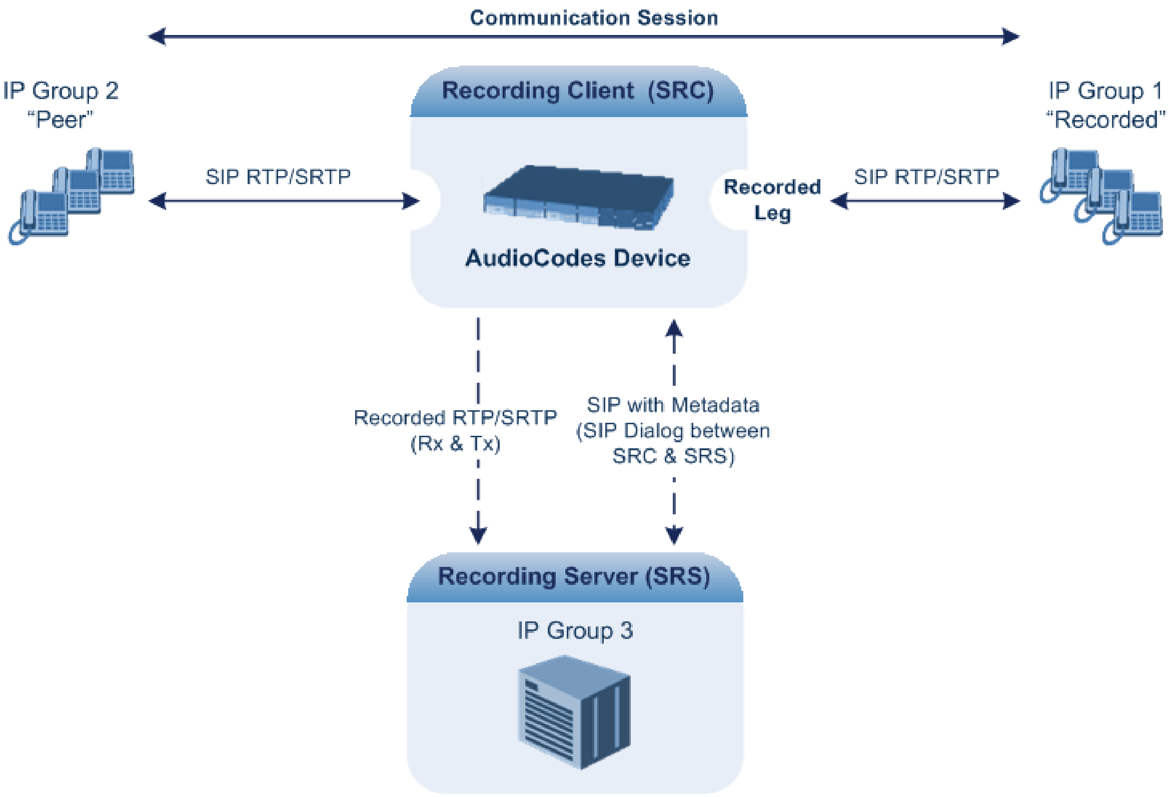
Notes:

- The SIP-based Media Recording feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 621. 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 19-1: 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. This recording leg must be one that is interfacing with one of the IP Groups. Specifying the leg is important as it determines the various call media attributes of the recorded RTP (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 Routing table must be set to the RTP leg if recording is required to be RTP, or set to the SRTP leg if recording is required to be SRTP.

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

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

- Two 'm=' lines that represent the two RTP/SRTP streams (Rx and Tx).
- Two 'a=label:' lines that identify the streams.
- XML body (also referred to as metadata) that provides information on the participants of the call session:
    - <group id>: Logging Session ID (displayed as [SID:nnnnn] in Syslog), converted from decimal to hex. This number remains the same even if the call is forwarded or transferred. This is important for recorded calls.

- <session id>: Originally recorded Call-ID, converted from decimal to hex.
- <group-ref>: same as <group id>.
- <participant id>: SIP From / To user.
- <nameID aor>: From/To user@host.
- <send> and <recv>: ID's for the RTP/SRTP streams in hex - bits 0-31 are the same as group, bits 32-47 are the RTP 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 (but without metadata) 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_ac1fffff85b
Content-Length: 1832

--boundary_ac1fffff85b
Content-Type: application/sdp
v=0
o=AudiocodesGW 921244928 921244893 IN IP4 10.33.8.70
s=SBC-Call
c=IN IP4 10.33.8.70
t=0 0
m=audio 6020 RTP/AVP 8 96
c=IN IP4 10.33.8.70
a=ptime:20
a=sendonly
a=label:1
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
m=audio 6030 RTP/AVP 8 96
c=IN IP4 10.33.8.70
a=ptime:20
a=sendonly
a=label:2
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
```

```
--boundary_ac1fffff85b
Content-Type: application/rs-metadata
Content-Disposition: recording-session
<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
  <datamode>complete</datamode>
  <group id="00000000-0000-0000-0000-00003a36c4e3">
    <associate-time>2010-01-24T01:11:57Z</associate-time>
  </group>
  <session id="0000-0000-0000-0000-00000000d0d71a52">
    <group-ref>00000000-0000-0000-0000-00003a36c4e3</group-ref>
    <start-time>2010-01-24T01:11:57Z</start-time>
    <ac:AvayaUCID
xmlns="urn:ietf:params:xml:ns:Avaya">FA080030C4E34B5B9E59</ac:Avay
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_ac1fffff85b–
```

19.1.1 Enabling SIP-based Media Recording

The following procedure describes how to enable the SIP-based media Recording feature. Once you have enabled this feature, your SIP Recording Routing rules (configured in "Configuring SIP Recording Routing Rules" on page 203) 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.

19.1.2 Configuring SIP Recording Routing Rules

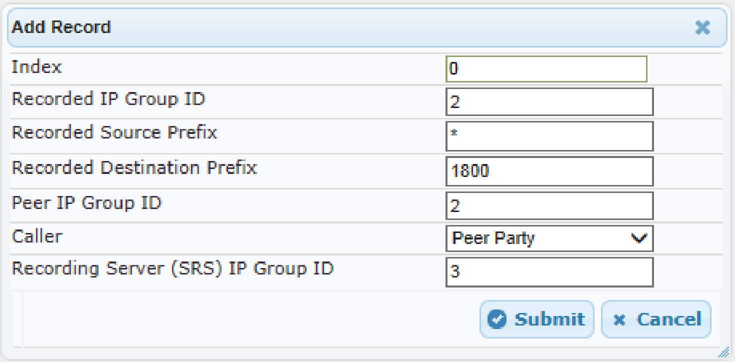
The SIP Recording Routing table lets you configure up to 30 SIP-based media recording rules. A SIP Recording Routing rule defines calls that you want to record. For an overview of this feature, see "SIP-based Media Recording" on page 199.

The following procedure describes how to configure SIP Recording Routing rules in the Web interface. You can also configure SIP Recording Routing rules using the table ini file parameter, SIPRecRouting or CLI command, configure voip/services sip-recording sip-rec-routing.

➢ **To configure a SIP Recording Routing rule:**

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

Figure 19-2: SIP Recording Routing Table - Add Record



Add Record

Index	0
Recorded IP Group ID	2
Recorded Source Prefix	*
Recorded Destination Prefix	1800
Peer IP Group ID	2
Caller	Peer Party
Recording Server (SRS) IP Group ID	3

The figure above shows a configuration example where the device records calls made by IP Group 1 to IP Group 2 that have the destination number prefix "1800". The device records the calls from the leg interfacing with IP Group 2, sending the recorded media to IP Group 3 (i.e., the SRS).

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

Table 19-1: SIP Recording Routing Parameter Descriptions

Parameter	Description
Index **[SIPRecRouting_Index]**	Defines an index number for the new table record.
Recorded IP Group ID CLI: recorded-ip-group-id **[SIPRecRouting_RecordedIPGroupID]**	Defines the IP Group participating in the call and the recording is done on the leg interfacing with this IP Group. **Note:** For an SBC RTP-SRTP session, the recorded IP Group must be set to the RTP leg if recording is required to be RTP, or set to the SRTP leg if recording is required to be SRTP.
Recorded Source Prefix CLI: recorded-src-prefix **[SIPRecRouting_RecordedSourcePrefix]**	Defines calls to record based on source number or URI.

Parameter	Description
Recorded Destination Prefix CLI: recorded-dst-prefix **[SIPRecRouting_RecordedDestinationPrefix]**	Defines calls to record based on destination number or URI.
Peer IP Group ID CLI: peer-ip-group-id **[SIPRecRouting_PeerIPGroupID]**	Defines the peer IP Group that is participating in the call.
Caller CLI: caller **[SIPRecRouting_Caller]**	Defines which calls to record according to which party is the caller. ▪ **[0]** Both (default) = Caller can be peer or recorded side ▪ **[1]** Recorded Party ▪ **[2]** Peer Party
Recording Server (SRS) IP Group ID CLI: srs-ip-group-id **[SIPRecRouting_SRSIPGroupID]**	Defines the IP Group of the recording server (SRS). **Note:** The SIP Interface used for communicating with the SRS is according to the SRD assigned to the SRS IP Group (in the IP Group table). If two SIP Interfaces are associated with the SRD - one for "SBC" and one for "GW &IP2IP" – the device uses the "SBC" SIP Interface. If no SBC SIP Interface type is configured, the device uses the “GW & IP2IP” interface.

19.1.3 Configuring SIP User Part for SRS

You can configure the SIP user part of the Request-URI for the recording server (SRS). The device inserts this user part in the SIP To header of the INVITE message sent to the SRS.

➢ **To configure the SIP user part for SRS:**

1. Open the SIP Recording page (**Configuration** tab > **VoIP** menu > **Services** > **SIP Recording**).
2. In the 'Recording Server (SRS) Destination Username' field, enter a user part value (string of up to 50 characters).
3. Click **Submit**, and then save ("burn") your settings to flash memory.

19.1.4 Interworking SIP-based Media Recording with Third-Party Vendors

The device can interwork the SIP-based Media Recording feature with third-party vendors, as described in the following subsections.

19.1.4.1 Genesys

The device's SIP-based media recording can interwork with Genesys' equipment. Genesys sends its proprietary X-Genesys-CallUUID header (which identifies the session) in the first SIP message, typically in the INVITE and the first 18x response. If the device receives a SIP message with Genesys SIP header, it adds the header's information to AudioCodes' proprietary tag in the XML metadata of the SIP INVITE that it sends to the recording server, as shown below:

```
<ac:GenesysUUID
xmlns="urn:ietf:params:xml:ns:Genesys">4BOKLLA3VH66JF112M1CC9VHKS1
4F0KP</ac:GenesysUUID>
```

19.1.4.2 Avaya UCID

The device's SIP-based media recording can interwork with Avaya equipment. The Universal Call Identifier (UCID) is Avaya's proprietary call identifier used to correlate call records between different systems and identifies sessions. Avaya generates this in outgoing calls. If the device receives a SIP INVITE from Avaya, it adds the UCID value, received in the User-to-User SIP header to AudioCodes' proprietary tag in the XML metadata of the SIP INVITE that it sends to the recording server. For example, if the received SIP header is:

```
User-to-User: 00FA080019001038F725B3;encoding=hex
```

the device includes the following in the XML metadata:

```
xml metadata:
<ac:AvayaUCID xmlns="urn:ietf:params:xml:ns:Avaya">
FA080019001038F725B3</ac:AvayaUCID>
```

19.2 RADIUS Authentication

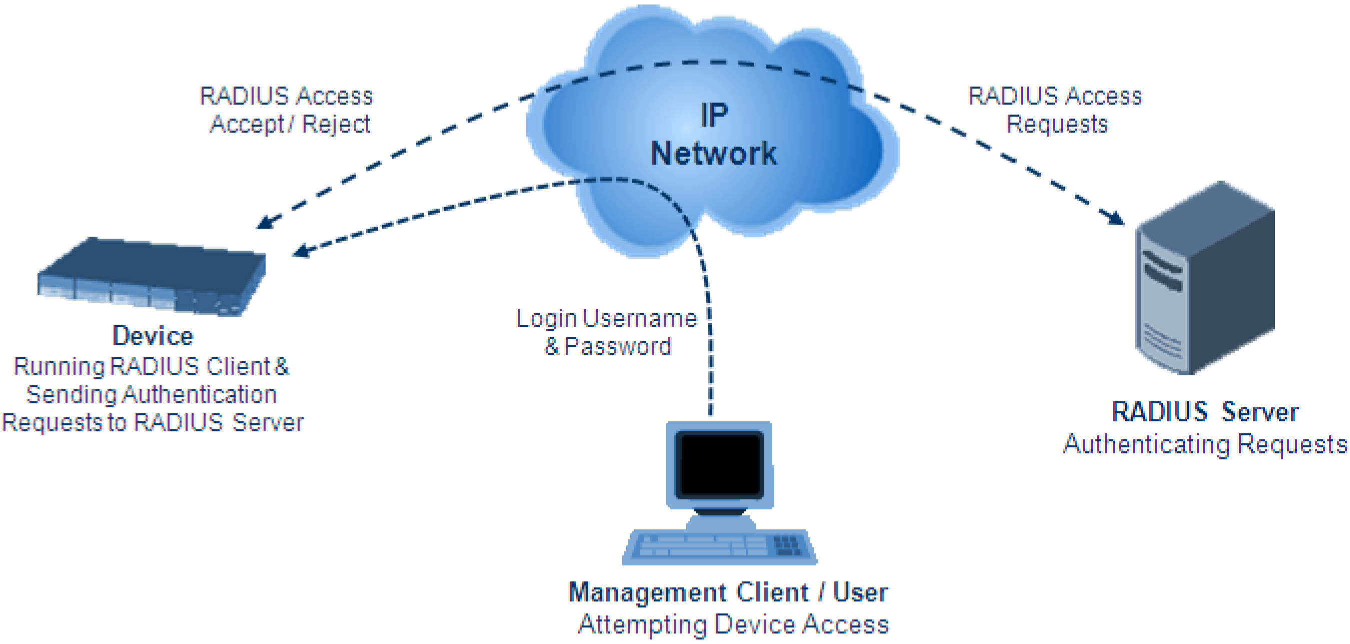
You can enhance security for your device by implementing Remote Authentication Dial-In User Service (RADIUS - RFC 2865) for authenticating multiple management user accounts of the device's embedded Web and Telnet (CLI) servers. Thus, RADIUS also prevents unauthorized access to your device.

When RADIUS authentication is not used, the user's login username and password are locally authenticated by the device in its Web Users table (database). However, the Web Users table can be used as a fallback mechanism in case the RADIUS server does not respond. For configuring local user accounts, see "Configuring Web User Accounts" on page 64.

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 19-3: 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 206
- Configure the device as a RADIUS client for communication with the RADIUS server - see "Configuring RADIUS Authentication" on page 208

19.2.1 Setting Up a Third-Party RADIUS Server

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

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

1. Define the 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 64.

   ```
   #
   # 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.

19.2.2 Configuring RADIUS Authentication

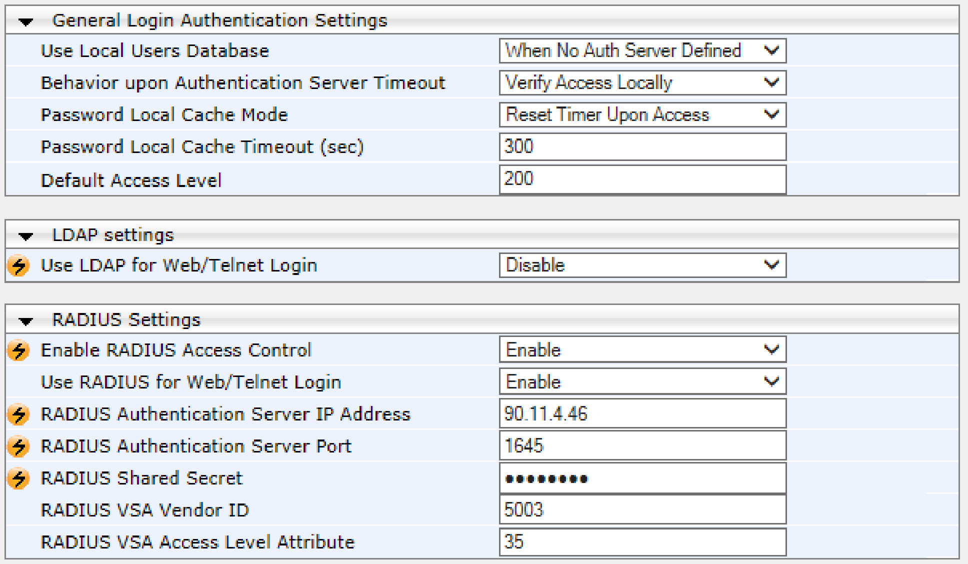
The following procedure describes how to configure the RADIUS feature in the Web interface. For a detailed description of the RADIUS parameters, see "RADIUS Parameters" on page 1007.

> **Note:** By default, the device communicates with the RADIUS server through the OAMP network interface. To specify a WAN interface for RADIUS communication, use the following CLI command:
>
> ```
> (config-system)# radius
> (radius)# source data interface <interface name, e.g., gigabitethernet 0/0>
> - or -
> (radius)# source data source-address interface <IP address of interface>
> ```
>
> To return to the OAMP interface, use the no command:
>
> ```
> (radius)# no source data interface <source data interface>
> ```

➢ **To configure RADIUS:**

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

Figure 19-4: Authentication Settings Page - RADIUS Configuration



▼ General Login Authentication Settings

Use Local Users Database	When No Auth Server Defined
Behavior upon Authentication Server Timeout	Verify Access Locally
Password Local Cache Mode	Reset Timer Upon Access
Password Local Cache Timeout (sec)	300
Default Access Level	200

▼ LDAP settings

Use LDAP for Web/Telnet Login	Disable

▼ RADIUS Settings

Enable RADIUS Access Control	Enable
Use RADIUS for Web/Telnet Login	Enable
RADIUS Authentication Server IP Address	90.11.4.46
RADIUS Authentication Server Port	1645
RADIUS Shared Secret	••••••••
RADIUS VSA Vendor ID	5003
RADIUS VSA Access Level Attribute	35

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

RADIUS server.

6. When implementing Web user access levels, do one of the following:
   - **If the RADIUS server response includes the access level attribute:** In the 'RADIUS VSA Access Level Attribute' field, enter the code that indicates the access level attribute in the VSA section of the received RADIUS packet. For defining the RADIUS server with access levels, see "Setting Up a Third-Party RADIUS Server" on page 206.
   - **If the RADIUS server response does not include the access level attribute:** In the 'Default Access Level' field, enter the default access level that is applied to all users authenticated by the RADIUS server.
7. Configure RADIUS timeout handling:
   a. From the 'Behavior upon Authentication Server Timeout' drop-down list, select the option if the RADIUS server does not respond within five seconds:
      - **Deny Access**: device denies user login access.
      - **Verify Access Locally**: device checks the username and password configured locally for the user (in the Web User Accounts page or Web Users table), and if correct, allows access.

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

   c. From the 'Password Local Cache Mode' drop-down list, select the option for the local RADIUS password cache timer:
      - **Reset Timer Upon Access**: upon each access to a Web page, the timer resets (reverts to the initial value configured in the previous step).
      - **Absolute Expiry Timer**: when you access a Web page, the timer doesn’t reset, but continues its count down.
8. Configure when the Web Users table must be used to authenticate login users. From the 'Use Local Users Database' drop-down list, select one of the following:
   - **When No Auth Server Defined (default):** When no RADIUS server is configured (or as fallback if the server is inaccessible).
   - **Always:** Always, but if not found, use the RADIUS server to authenticate the user.
9. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

19.2.3 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**).

19.2.4 Authenticating RADIUS in the URL

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

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

19.3 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 216). 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 228.

  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 220.

  If connection with the LDAP server disconnects (broken), the device sends the SNMP alarm, acLDAPLostConnection. Upon successful reconnection, the alarm clears. If connection with the LDAP server is disrupted during the search, all search requests are dropped and an alarm indicating a failed status is sent to client applications.

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

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

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

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

The device then assigns the user the access level configured for that group (in "Configuring Access Level per Management Groups Attributes" on page 217). 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 216.
- 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 ($) sign to represent the username, for example, (sAMAcountName=$). For configuring the search filter, see "Configuring the LDAP Search Filter Attribute" on page 217.
- 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 212).

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

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

- Search method for searching DN object records between LDAP servers and within each LDAP server (see "Configuring LDAP Search Methods" on page 220).
- 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 223.

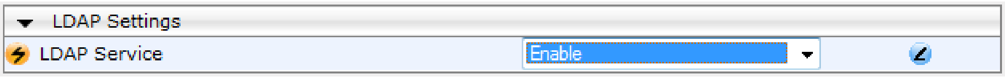
19.3.1 Enabling the LDAP Service

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 19-5: Enabling LDAP on the LDAP Settings Page



LDAP Settings

LDAP Service: Enable

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

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

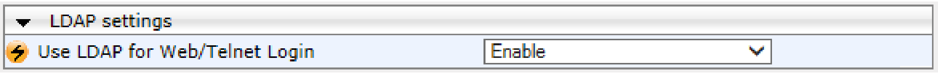
The LDAP service can be used for authenticating and authorizing device management users (Web and CLI), based on the user's login username and password (credentials). At the same, it can also be used to determine users' management access levels (privileges).

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 19-6: Authentication Settings Page - Enabling LDAP-based Login



LDAP settings

Use LDAP for Web/Telnet Login: Enable

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

19.3.3 Configuring LDAP Servers

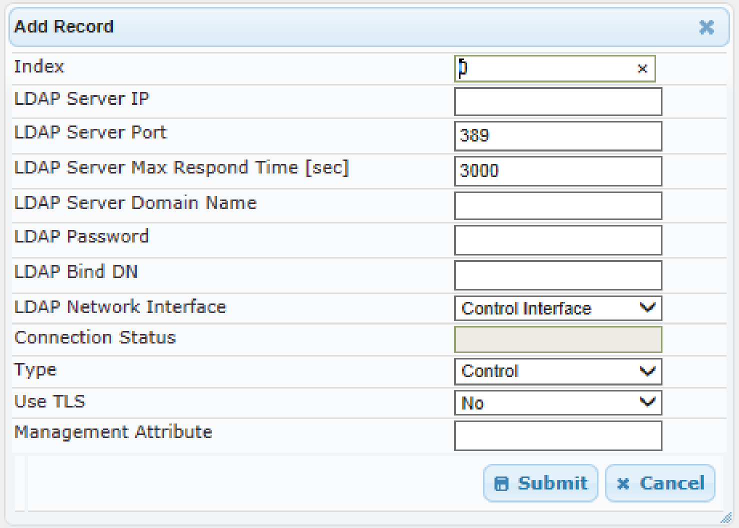
The LDAP Configuration table lets you configure up to four LDAP servers. This table defines the address and connectivity settings of the LDAP server. The LDAP server can be configured for SIP-related queries (e.g., routing and manipulation) or LDAP-based management user login authentication and authorization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

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

➢ To configure an LDAP server:

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

Figure 19-7: LDAP Configuration Table - Add Record



Add Record

Index	0
LDAP Server IP	
LDAP Server Port	389
LDAP Server Max Respond Time [sec]	3000
LDAP Server Domain Name	
LDAP Password	
LDAP Bind DN	
LDAP Network Interface	Control Interface
Connection Status	
Type	Control
Use TLS	No
Management Attribute	

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

Table 19-2: LDAP Configuration Table Parameter Descriptions

Parameter	Description
Index **[LdapConfiguration_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
LDAP Server IP CLI: server-ip **[LdapConfiguration_LdapConfServerIp]**	Defines the IP address of the LDAP server (in dotted-decimal notation, e.g., 192.10.1.255). By default, no IP address is defined. **Note:** If you want to use an FQDN for the LDAP server, leave this parameter undefined and configure the FQDN in the 'LDAP Server Domain Name' parameter (see below).
LDAP Server Port CLI: server-port **[LdapConfiguration_LdapConfServerPort]**	Defines the port number of the LDAP server. The valid value range is 0 to 65535. The default port number is 389.
LDAP Server Max Respond Time CLI: max-respond-time **[LdapConfiguration_LdapConfServerMaxRespondTime]**	Defines the time (in seconds) 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 223.

Parameter	Description
LDAP Server Domain Name CLI: domain-name **[LdapConfiguration_LdapConfServerDomainName]**	Defines the domain name (FQDN) of the LDAP server. The device tries to connect to the LDAP server according to the IP address listed in the received DNS query. If there is no connection to the LDAP server or the connection to the LDAP server fails, the device tries to connect to the LDAP server with the next IP address in the DNS query list. **Note:** The 'LDAP Server IP' parameter takes precedence over this parameter. Thus, if you want to use an FQDN, leave the 'LDAP Server IP' parameter undefined.
LDAP Password CLI: password **[LdapConfiguration_LdapConfPassword]**	Defines the user password for accessing the LDAP server during connection and binding operations. ▪ LDAP-based SIP queries: The parameter is the password used by the device to authenticate itself, as a client, to obtain LDAP service from the LDAP server. ▪ LDAP-based user login authentication: The parameter represents the login password entered by the user during a login attempt. You can use the $ (dollar) sign in this value to enable the device to automatically replace the $ sign with the user's login password in the search filter, which it sends to the LDAP server for authenticating the user's username-password combination. For example, $. **Note:** By default, the device sends the password in clear-text format. You can enable the device to encrypt the password using TLS (see the 'Use SSL' parameter below).
LDAP Bind DN CLI: bind-dn **[LdapConfiguration_LdapConfBindDn]**	Defines the LDAP server's bind Distinguished Name (DN) or username. ▪ LDAP-based SIP queries: The DN is used as the username during connection and binding to the LDAP server. The DN is used to uniquely name an AD object. Below are example parameter settings: ✓ cn=administrator,cn=Users,dc=domain,dc=com ✓ administrator@domain.com ✓ domain\administrator ▪ LDAP-based user login authentication: This parameter represents the login username entered by the user during a login attempt. You can use the $ (dollar) sign in this value to enable the device to automatically replace the $ sign with the user's login username in the search filter, which it sends to the LDAP server for authenticating the user's username-password combination. An example configuration for this parameter is $@sales.local, where the device replaces the $ with the entered username, for example, JohnD@sales.local. The username can also be configured with the domain name of the LDAP server. **Note:** By default, the device sends the username in clear-text format. You can enable the device to encrypt the username using TLS (see the 'Use SSL' parameter below).

Parameter	Description
LDAP Network Interface CLI: interface-type **[LdapConfiguration_LdapConfInterfaceType]**	Assigns one of the device's IP network interfaces for communicating with the LDAP server. ▪ **[0]** Control Interface (default) = The top-most IP network interface row in the IP Interfaces table that is configured for a Control application (may be combined with other applications such as OAMP and Media) is used. ▪ **[1]** OAM Interface = The OAMP interface (may be combined with other applications such as Control and Media) in the IP Interfaces table is used. For configuring IP network interfaces, see "Configuring IP Network Interfaces" on page 130.
Type CLI: type **[LdapConfiguration_Type]**	Defines whether the LDAP server is used for SIP-related queries or management login authentication-related queries. ▪ **[0]** Control (Default) ▪ **[1]** Management **Note:** If you use the same LDAP server for both management and SIP (Control) related applications, the device establishes different LDAP sessions for each application.
Management Attribute CLI: mgmt-attr **[LdapConfiguration_MngmAuthAtt]**	Defines the LDAP attribute name to query, which contains a list of groups to which the user is a member. For Active Directory, this attribute is typically "memberOf". The attribute's values (groups) are used to determine the user's management access level; the group's corresponding access level is configured in "Configuring Access Level per Management Groups Attributes" on page 217. **Notes:** ▪ This parameter is applicable only to LDAP-based login authentication and authorization (i.e., the 'Type' parameter is set to **Management**). ▪ If this functionality is not used, the device assigns the user the configured default access level. For more information, see "Configuring Access Level per Management Groups Attributes" on page 217.
Use SSL CLI: **[LdapConfiguration_useTLS]**	Enables the device to encrypt the username and password (for Control and Management related queries) using TLS when sending them to the LDAP server. ▪ **[0]** No = (Default) Username and password are sent in clear-text format. ▪ **[1]** Yes
Connection Status CLI: connection-status **[LdapConfiguration_ConnectionStatus]**	(Read-only) Displays the connection status with the LDAP server. ▪ "Not Applicable" ▪ "LDAP Connection Broken" ▪ "Connecting" ▪ "Connected" **Note:** For more information about a disconnected LDAP connection, see your Syslog messages generated by the device.

19.3.4 Configuring LDAP DNs (Base Paths) per LDAP Server

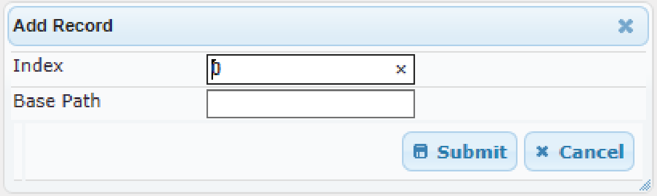
The LDAP Search DN Table table lets you configure LDAP base paths. The table is a "child" of the LDAP Configuration table (see "Configuring LDAP Servers" on page 212) and configuration is done per LDAP server. For the device to run a search using the LDAP service, the base path to the directory's subtree, 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 DNs per LDAP server in the Web interface. You can also configure this using the table ini file parameter, LdapServersSearchDNs or CLI command, configure voip/ldap/ldap-servers-search-dns.

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

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

Figure 19-8: LDAP Search DN Table - Add Record



Add Record

Index: 0

Base Path:

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

Table 19-3: LDAP Search DN Table Parameter Descriptions

Parameter	Description
Index CLI: set internal-index **[LdapServersSearchDNs_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Base Path CLI: set base-path **[LdapServersSearchDNs_Base_Path]**	Defines the full path (DN) to the objects in the AD where the query is done. The valid value is a string of up to 256 characters. For example: OU=NY,DC=OCSR2,DC=local. In this example, the DN path is defined by the LDAP names, OU (organizational unit) and DC (domain component).

19.3.5 Configuring the LDAP Search Filter Attribute

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

- **Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"):** The DN defines the location in the directory from which the LDAP search begins and is configured in "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 216.
- **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 ($) sign to represent the username. For example, the filter can be configured as "(sAMAcountName=$)", where if the user attempts to log in with the username "SueM", the LDAP search is done only for the attribute sAMAcountName 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 212).

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

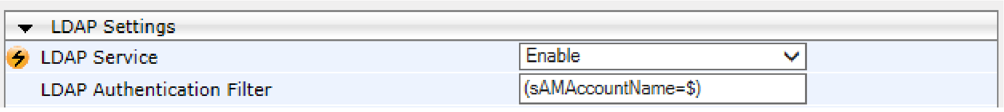
Notes:

- 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 19-9: LDAP Settings Page - LDAP Search Filter



LDAP Settings

LDAP Service	Enable
LDAP Authentication Filter	(sAMAccountName=$)

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

19.3.6 Configuring Access Level per Management Groups Attributes

The Management LDAP Groups table lets you configure LDAP group objects and their corresponding management user access level. The table is a "child" of the LDAP Configuration table (see "Configuring LDAP Servers" on page 212) 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 212).

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 64.

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 DNs (Base Paths) per LDAP Server" on page 216.
- 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,D
C=com
```

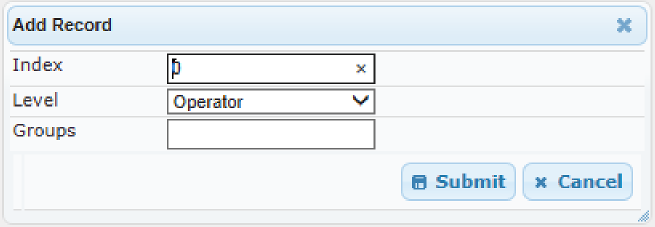
The device searches this LDAP response for the group names that you configured in the Management LDAP Groups table in order to determine the user's access level. If the device finds a group name, the user is assigned the corresponding access level and login is permitted; otherwise, login is denied. Once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

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

➢ To configure management groups and corresponding access level:

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

Figure 19-10: Management LDAP Groups Table - Add Record



Add Record

Index	0
Level	Operator
Groups	

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

Table 19-4: Management LDAP Groups Table Parameter Descriptions

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

19.3.7 Configuring LDAP Search Methods

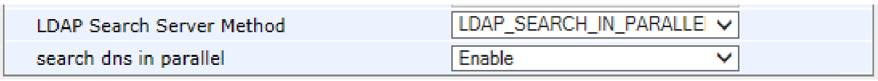
You can configure the device's method for searching the LDAP server(s) for the configured DN objects:

- **DN Search Method between Two LDAP Servers:** When two LDAP servers are implemented, the device runs an LDAP query to search for DN object records on both LDAP servers. You can configure how the device queries the DN object record between the two LDAP servers:
  - **Parallel Search:** The device queries the LDAP servers simultaneously.
  - **Sequential Search:** The device first queries one of the LDAP servers, and if the DN object is not found, it queries the second LDAP server.
- **DN Search Method within an LDAP Server:** You can configure how the device queries the DN object record within each LDAP server:
  - **Parallel Search:** The device queries all DN objects simultaneously. For example, a search for the DN object record "JohnD" is done at the same time in the "Marketing", "Sales" and "Administration" DN objects.
  - **Sequential Search:** The device queries each DN object, one by one, until a result is found. For example, a search for the DN object record "JohnD" is first run in DN object "Marketing" and if a result is not found, it searches in "Sales", and if not found, it searches in "Administration", and so on.

➢ **To configure LDAP search methods:**

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

Figure 19-11: LDAP Settings Page - Search Methods



LDAP Search Server Method	LDAP_SEARCH_IN_PARALLE
search dns in parallel	Enable

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

19.3.8 Configuring the Device's LDAP Cache

The device can optionally store recent LDAP queries and responses with an LDAP server in its local cache. The cache is used for subsequent queries and/or in case of LDAP server failure.

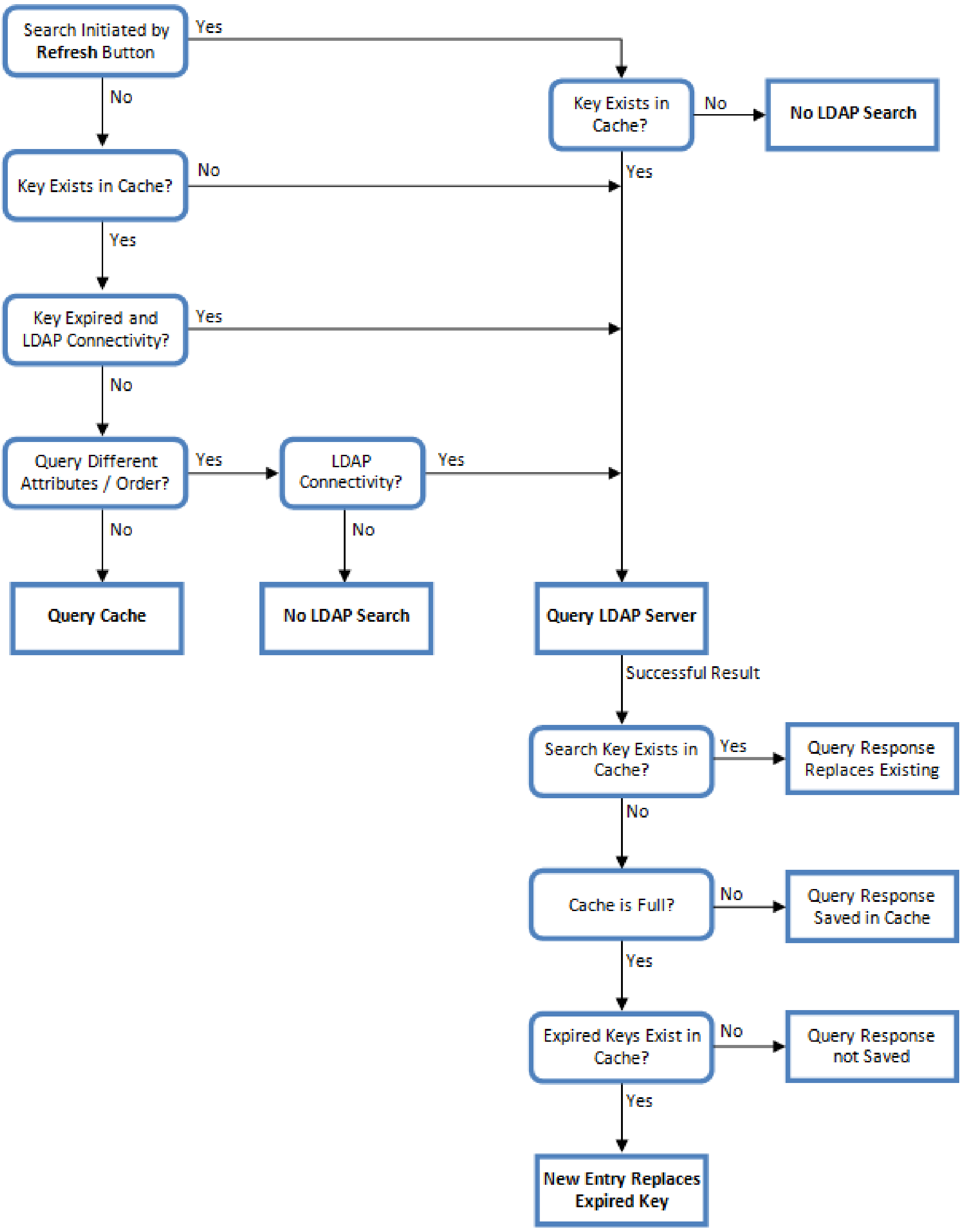
> **Note:** The LDAP Cache feature is applicable only to LDAP-based SIP queries (Control).

The advantage of enabling this feature includes the following:

- Improves routing decision performance by using local cache for subsequent LDAP queries
- Reduces number of queries performed on an LDAP server and corresponding bandwidth consumption
- Provides partial survivability in case of intermittent LDAP server failure (or network isolation)

The handling of LDAP queries with the LDAP cache is shown in the flowchart below:

Figure 19-12: LDAP Query Process with Local LDAP Cache


Search Initiated by Refresh Button
Yes
No
Key Exists in Cache?
No
No LDAP Search
Yes
Key Exists in Cache?
No
Yes
Key Expired and LDAP Connectivity?
Yes
No
Query Different Attributes / Order?
Yes
LDAP Connectivity?
Yes
No
No
Query Cache
No LDAP Search
Query LDAP Server
Successful Result
Search Key Exists in Cache?
Yes
Query Response Replaces Existing
No
Cache is Full?
No
Query Response Saved in Cache
Yes
Expired Keys Exist in Cache?
No
Query Response not Saved
Yes
New Entry Replaces Expired Key

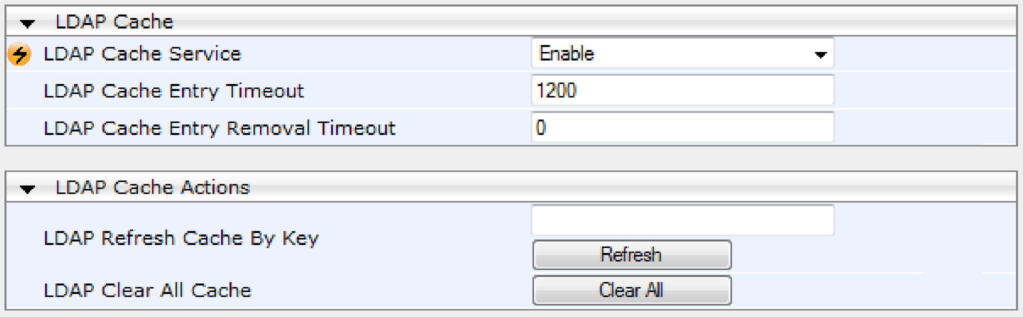
Note: If for the first LDAP query, the result fails for at least one attribute and is successful for at least one, the partial result is cached. However, for subsequent queries, the device does not use the partially cached result, but does a new query with the LDAP server again.

The following procedure describes how to configure the device's LDAP cache in the Web interface. For a full description of the cache parameters, see "LDAP Parameters" on page 1009.

➢ To configure the LDAP cache:

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

Figure 19-13: LDAP Settings Page - Cache Parameters



LDAP Cache

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

LDAP Cache Actions

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

The LDAP Settings page also provides you with the following buttons:

- **LDAP Refresh Cache by Key:** Refreshes a saved LDAP entry response in the cache of a specified LDAP search key. If a request with the specified key exists in the cache, the request is resent to the LDAP server.
- **LDAP Clear All Cache:** Removes all LDAP entries in the cache.

19.3.9 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 64.

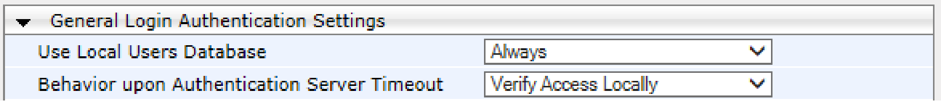
Notes:

- 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 19-14: Authentication Settings Page - Local Database for Login Authentication



General Login Authentication Settings

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

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

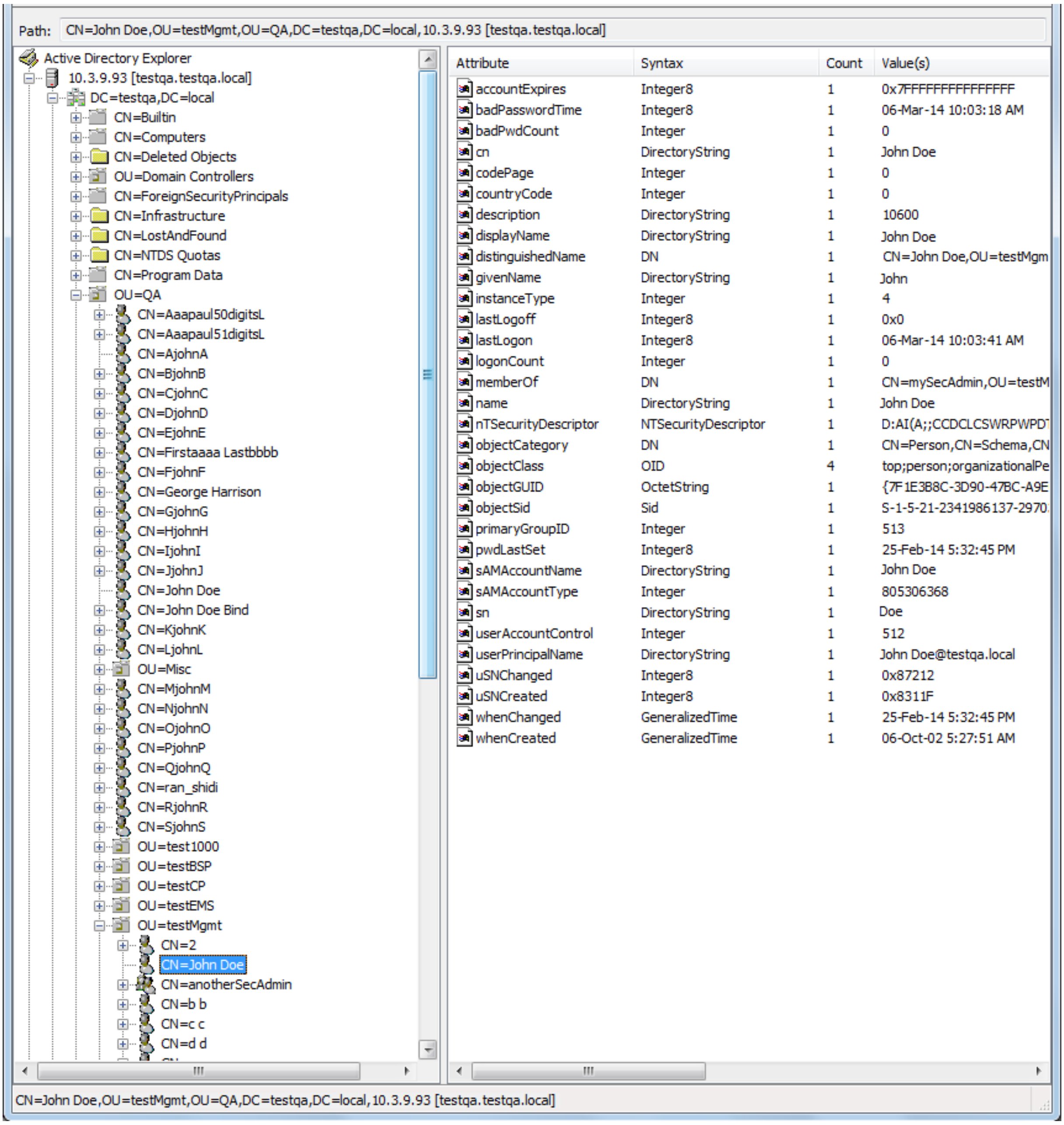
19.3.10 LDAP-based Login Authentication Example

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

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

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

Figure 19-15: 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]

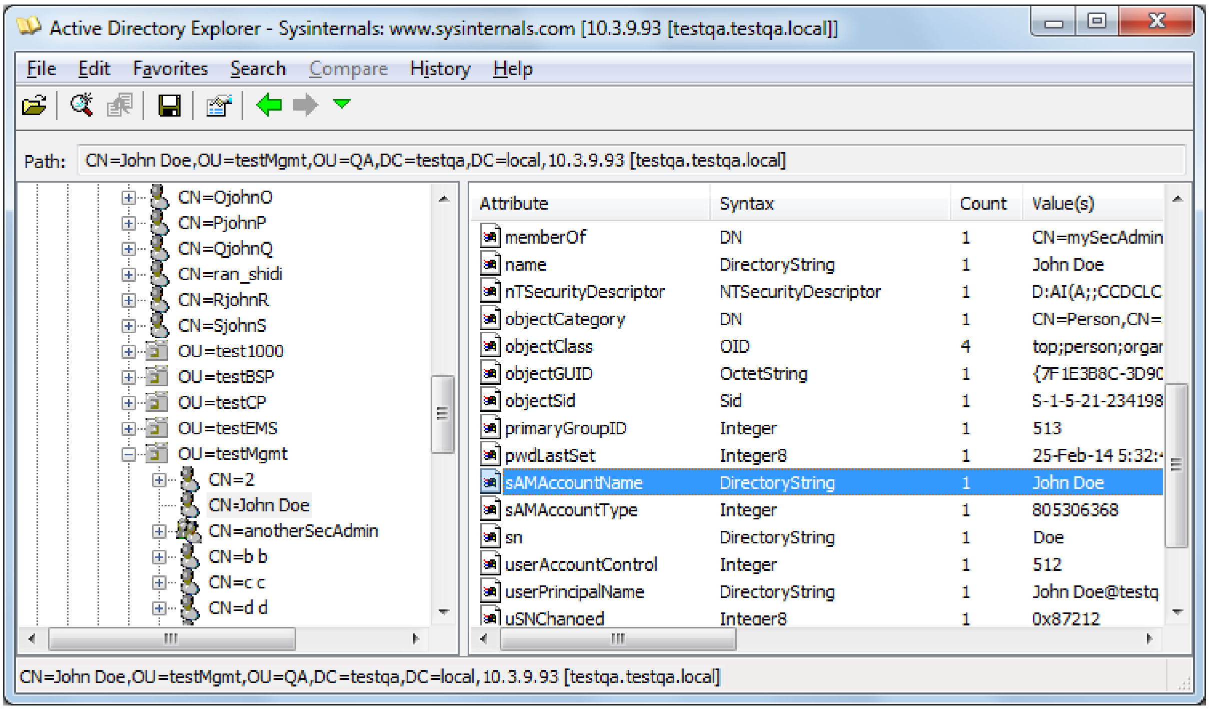
```
Active Directory Explorer
  10.3.9.93 [testqa.testqa.local]
    DC=testqa,DC=local
      CN=Builtin
      CN=Computers
      CN=Deleted Objects
      OU=Domain Controllers
      CN=ForeignSecurityPrincipals
      CN=Infrastructure
      CN=LostAndFound
      CN=NTDS Quotas
      CN=Program Data
      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;;CCDCLCSWRPWPDT
objectCategory	DN	1	CN=Person,CN=Schema,CN
objectClass	OID	4	top;person;organizationalPe
objectGUID	OctetString	1	{7F1E3B8C-3D90-47BC-A9E
objectSid	Sid	1	S-1-5-21-2341986137-2970
primaryGroupID	Integer	1	513
pwdLastSet	Integer8	1	25-Feb-14 5:32:45 PM
sAMAccountName	DirectoryString	1	John Doe
sAMAccountType	Integer	1	805306368
sn	DirectoryString	1	Doe
userAccountControl	Integer	1	512
userPrincipalName	DirectoryString	1	John Doe@testqa.local
uSNChanged	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 19-16: Username Found using sAMAccount Attribute Search Filter


Active Directory Explorer - Sysinternals: www.sysinternals.com [10.3.9.93 [testqa.testqa.local]]
File Edit Favorites Search Compare History Help

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

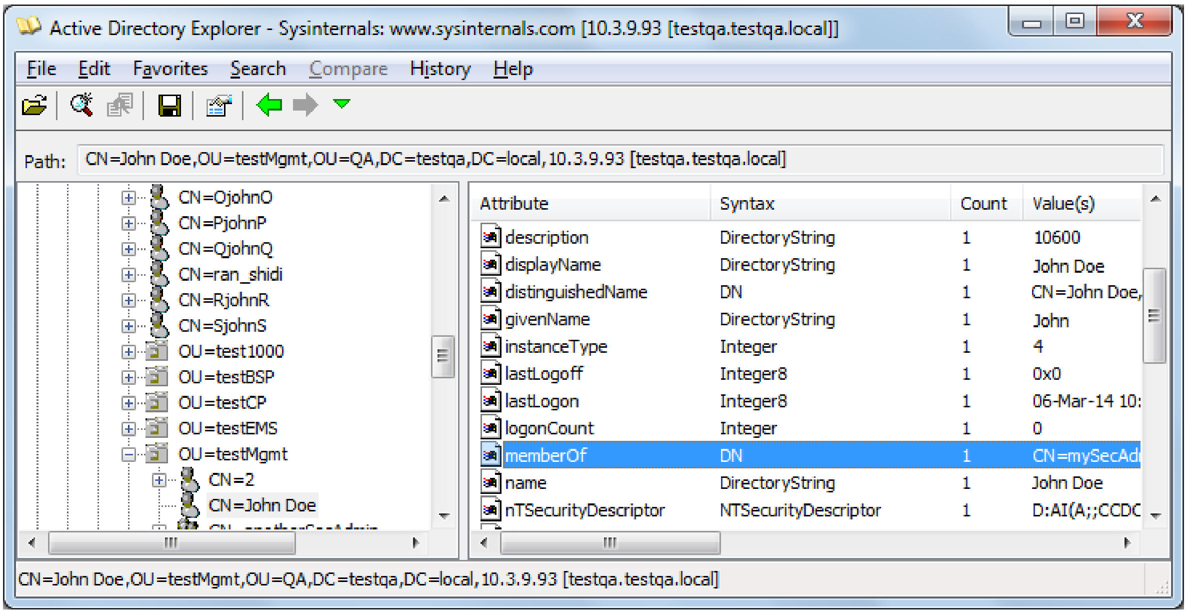
CN=OjohnO
CN=PjohnP
CN=QjohnQ
CN=ran_shidi
CN=RjohnR
CN=SjohnS
OU=test1000
OU=testBSP
OU=testCP
OU=testEMS
OU=testMgmt
CN=2
CN=John Doe
CN=anotherSecAdmin
CN=b b
CN=c c
CN=d d

Attribute	Syntax	Count	Value(s)
memberOf	DN	1	CN=mySecAdmin
name	DirectoryString	1	John Doe
nTSecurityDescriptor	NTSecurityDescriptor	1	D:AI(A;;CCDCLC
objectCategory	DN	1	CN=Person,CN=
objectClass	OID	4	top;person;orgar
objectGUID	OctetString	1	{7F1E3B8C-3D9C
objectSid	Sid	1	S-1-5-21-234198
primaryGroupID	Integer	1	513
pwdLastSet	Integer8	1	25-Feb-14 5:32:4
sAMAccountName	DirectoryString	1	John Doe
sAMAccountType	Integer	1	805306368
sn	DirectoryString	1	Doe
userAccountControl	Integer	1	512
userPrincipalName	DirectoryString	1	John Doe@testq
uSNChanged	Integer8	1	0x87212

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

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

Figure 19-17: User's memberOf Attribute


Active Directory Explorer - Sysinternals: www.sysinternals.com [10.3.9.93 [testqa.testqa.local]]
File Edit Favorites Search Compare History Help

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

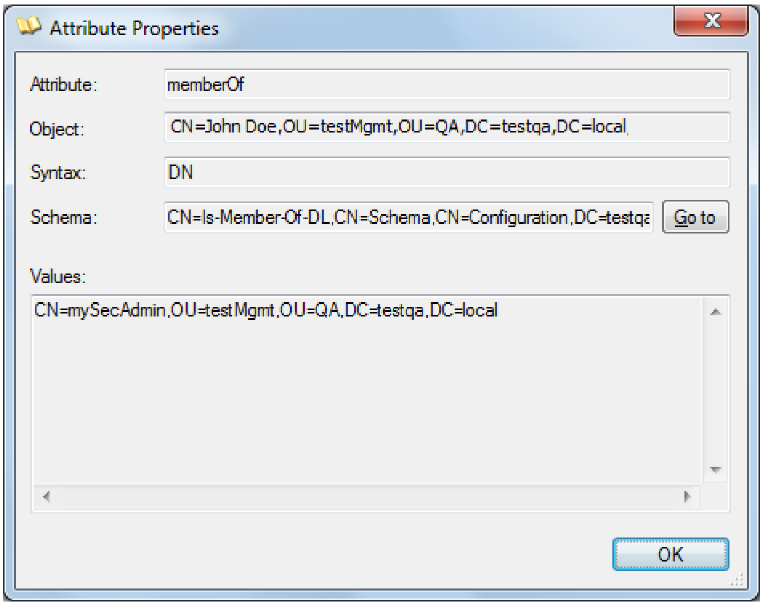
CN=OjohnO
CN=PjohnP
CN=QjohnQ
CN=ran_shidi
CN=RjohnR
CN=SjohnS
OU=test1000
OU=testBSP
OU=testCP
OU=testEMS
OU=testMgmt
CN=2
CN=John Doe

Attribute	Syntax	Count	Value(s)
description	DirectoryString	1	10600
displayName	DirectoryString	1	John Doe
distinguishedName	DN	1	CN=John Doe,
givenName	DirectoryString	1	John
instanceType	Integer	1	4
lastLogoff	Integer8	1	0x0
lastLogon	Integer8	1	06-Mar-14 10:
logonCount	Integer	1	0
memberOf	DN	1	CN=mySecAd
name	DirectoryString	1	John Doe
nTSecurityDescriptor	NTSecurityDescriptor	1	D:AI(A;;CCDC

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

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

Figure 19-18: User's mySecAdmin Group in memberOf Management Attribute



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

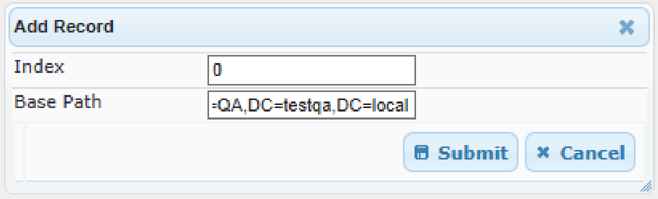
Values:

CN=mySecAdmin,OU=testMgmt,OU=QA,DC=testqa,DC=local

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

- The DN is configured in the LDAP Configuration table (see "Configuring LDAP Servers" on page 212):

Figure 19-19: Configuring DN

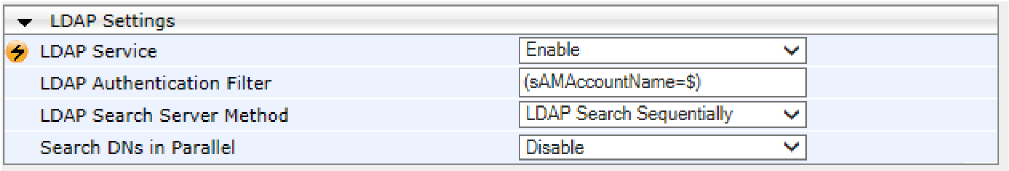


Index: 0

Base Path: =QA,DC=testqa,DC=local

- 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 217):

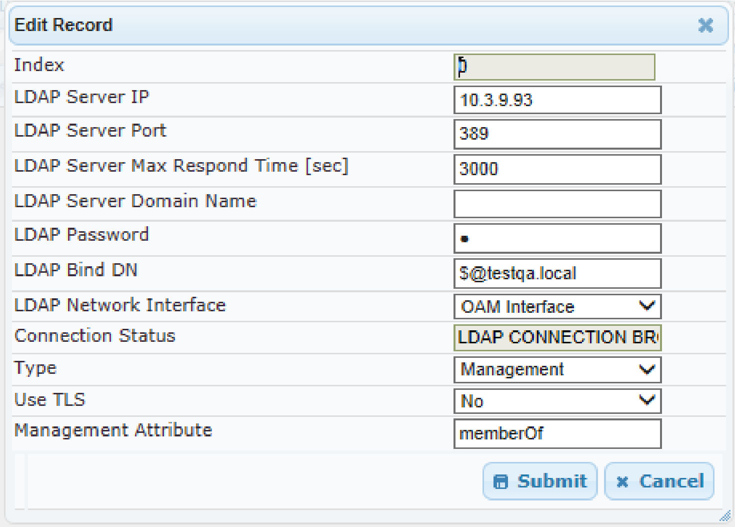
Figure 19-20: Configuring Search Attribute Filter



LDAP Service	Enable
LDAP Authentication Filter	(sAMAccountName=$)
LDAP Search Server Method	LDAP Search Sequentially
Search DNs in Parallel	Disable

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

Figure 19-21: Configuring Management Attribute

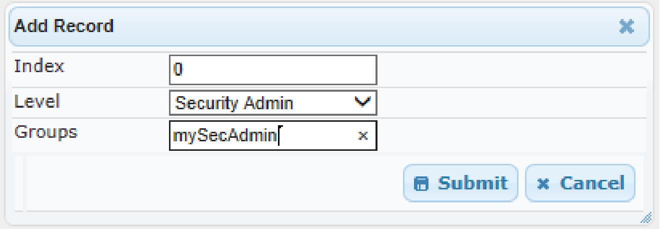


Edit Record

Index	0
LDAP Server IP	10.3.9.93
LDAP Server Port	389
LDAP Server Max Respond Time [sec]	3000
LDAP Server Domain Name	
LDAP Password	•
LDAP Bind DN	$@testqa.local
LDAP Network Interface	OAM Interface
Connection Status	LDAP CONNECTION BR
Type	Management
Use TLS	No
Management Attribute	memberOf

- 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 217):

Figure 19-22: Configuring Management Group Attributes for Determining Access Level



Add Record

Index	0
Level	Security Admin
Groups	mySecAdmin

19.3.11 Active Directory-based Routing for Microsoft Lync

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

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

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

19.3.11.1 Querying the AD and Routing Priority

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

Table 19-5: Parameters for Configuring Query Attribute Key

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

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

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

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

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

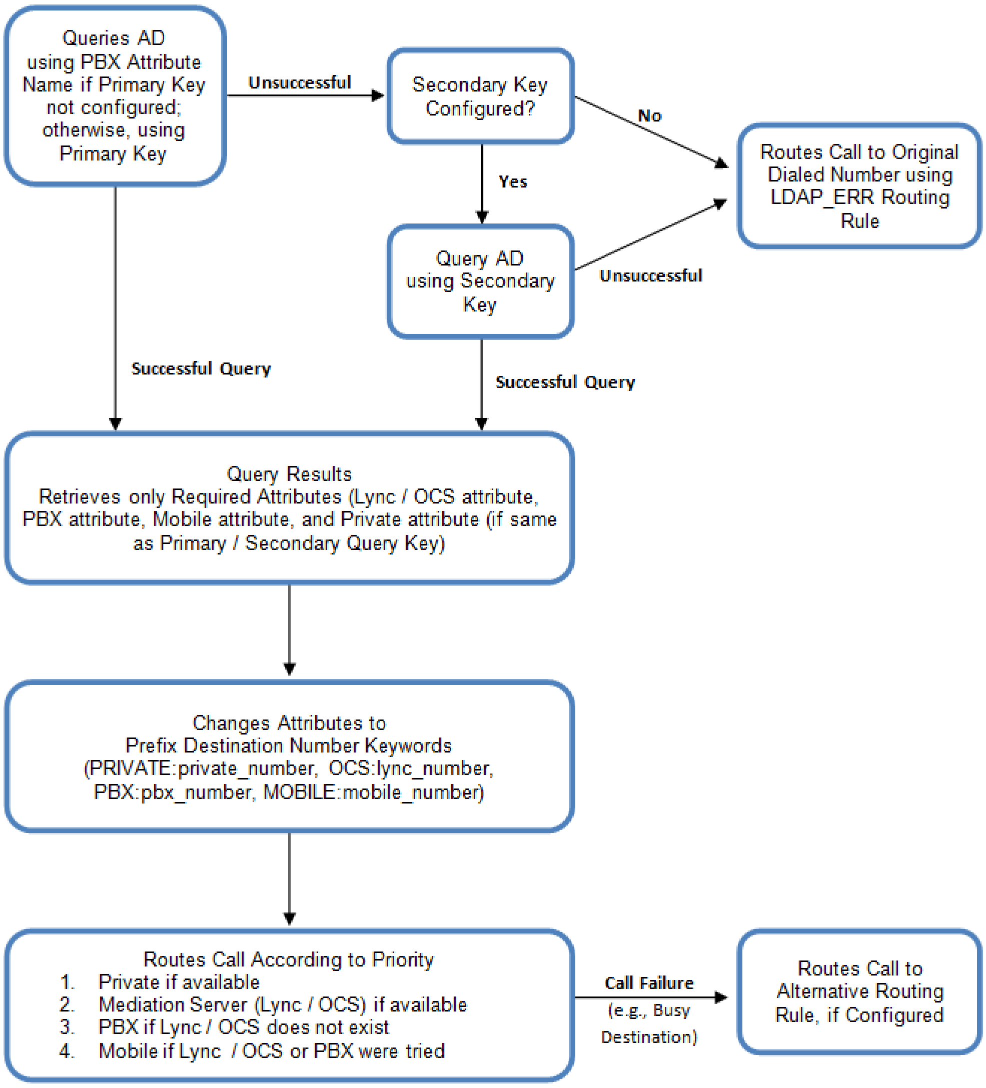
7. The device uses the Outbound IP Routing table to route the call based on the LDAP query result. The device routes the call according to the following priority:
    1. **Private line:** If the query is done for the private attribute and it's found, the device routes the call according to this attribute.
    2. **Mediation Server SIP address (Lync):** If the private attribute does not exist or is not queried, the device routes the call to the Mediation Server (which then routes the call to the Lync client).
    3. **PBX / IP PBX:** If the Lync client is not found in the AD, it routes the call to the PBX / IP PBX.
    4. **Mobile number:** If the Lync client (or Mediation Server) is unavailable (e.g., SIP response 404 "Not Found" upon INVITE sent to Lync client), and the PBX / IP PBX is also unavailable, the device routes the call to the user's mobile number (if exists in the AD).

5. **Alternative route:** If the call routing to all the above fails (e.g., due to unavailable destination - call busy), the device can route the call to an alternative destination if an alternative routing rule is configured.
6. **"Redundant" route:** If the query failed (i.e., no attribute found in the AD), the device uses the routing rule matching the "LDAP_ERR" prefix destination number value.

Note: For Enterprises implementing a PBX / IP PBX system, but yet to migrate to Lync Server, if the PBX / IP PBX system is unavailable or has failed, the device uses the AD query result for the user's mobile phone number, routing the call through the PSTN to the mobile destination.

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

Figure 19-23: LDAP Query Flowchart


Queries AD using PBX Attribute Name if Primary Key not configured; otherwise, using Primary Key
Unsuccessful
Secondary Key Configured?
No
Routes Call to Original Dialed Number using LDAP_ERR Routing Rule
Yes
Query AD using Secondary Key
Unsuccessful
Successful Query
Successful Query
Query Results
Retrieves only Required Attributes (Lync / OCS attribute, PBX attribute, Mobile attribute, and Private attribute (if same as Primary / Secondary Query Key)
Changes Attributes to
Prefix Destination Number Keywords
(PRIVATE:private_number, OCS:lync_number, PBX:pbx_number, MOBILE:mobile_number)
Routes Call According to Priority
1. Private if available
2. Mediation Server (Lync / OCS) if available
3. PBX if Lync / OCS does not exist
4. Mobile if Lync / OCS or PBX were tried
Call Failure
(e.g., Busy Destination)
Routes Call to Alternative Routing Rule, if Configured

Note: If you are using the device's local LDAP cache, see "Configuring the Device's LDAP Cache" on page 220 for the LDAP query process.

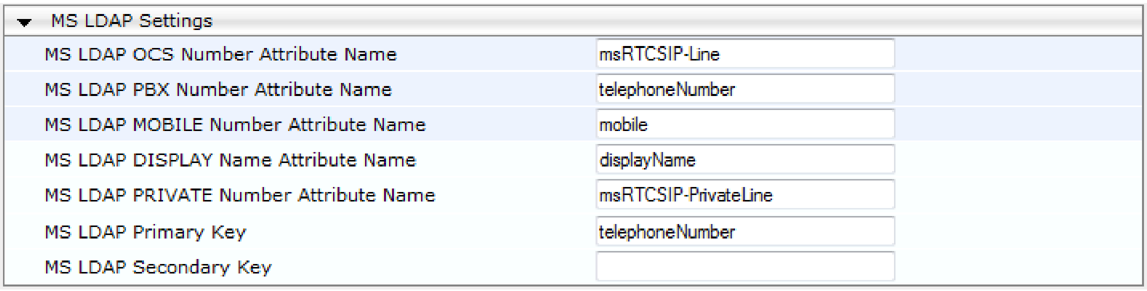
19.3.11.2 Configuring AD-Based Routing Rules

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

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

1. Configure the LDAP server parameters, as described in "Configuring LDAP Servers" on page 212.
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 19-24: 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. Gateway application: Configure AD-based Tel-to-IP routing rules:
   a. Open the Outbound IP Routing table (Configuration tab > VoIP menu > GW and IP to IP > Routing > Tel to IP Routing). For more information, see Configuring Outbound IP Routing Table on page 397.
   b. 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:
      - 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 Tel call to the LDAP server, using the value "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.
4. SBC application: Configure AD-based IP-to-IP routing rules:
   a. Open the IP-to-IP Routing Table page (Configuration tab > VoIP menu > SBC > Routing SBC > IP-to-IP Routing Table). For more information, see Configuring SBC IP-to-IP Routing Rules on page 552.
   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 Tel-to-IP routing rules in the Outbound IP Routing table:

Table 19-6: AD-Based Tel-to-IP Routing Rule Configuration Examples

Index	Dest. Phone Prefix	Dest. IP Address
1	PRIVATE:	10.33.45.60
2	PBX:	10.33.45.65
3	OCS:	10.33.45.68
4	MOBILE:	10.33.45.100
5	LDAP_ERR	10.33.45.80
6	*	LDAP
7	*	10.33.45.72

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

Table 19-7: 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 Tel-to-IP Release Reason (see Alternative Routing for Tel-to-IP Calls on page 412) or SBC Alternative Routing Reason (see Configuring SIP Response Codes for Alternative Routing Reasons on page 561) 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.

19.3.11.3 Querying the AD for Calling Name

The device can retrieve the calling name (display name) from an LDAP-compliant server (for example, Microsoft Active Directory / AD) for Tel-to-IP calls that are received without a calling name.

The device uses the calling number (PBX or mobile number) for the LDAP query. Upon an incoming INVITE, the device queries the AD based on the Calling Number search key (tries to match the calling number with the appropriate "telephoneNumber" or "mobile" number AD attribute entry). It then searches for the corresponding calling name attribute, configured by the MSLDAPDisplayNameAttributeName parameter (e.g., “displayName”). The device uses the resultant calling name as the display name parameter in the SIP From header of the outgoing INVITE message.

To configure this feature, the following keywords are used in the Calling Name Manipulation Table for Tel-to-IP Calls table for the 'Prefix/Suffix to Add' fields, which can be combined with other characters:

- "$LDAP-PBX": LDAP query using the MSLDAPPBXAttrName parameter as the search key
- "$LDAP-MOBILE": LDAP query using MSLDAPMobileAttrName parameter as the search key

If the source (calling) number of the Tel-to-IP call matches the PBX / MOBILE (e.g., "telephoneNumber" and "mobile") number in the AD server, the device uses the resultant Display Name instead of the keyword(s).

For example, assume the following configuration in the Calling Name Manipulation Table for Tel-to-IP Calls:

- 'Source Prefix' field is set to "4".
- 'Prefix to Add' field is set to "$LDAP-PBX Office".

If the calling number is 4046 and the resultant LDAP query display name is "John Doe", the device sends the INVITE message with the following From header:

```
From: John Doe <sip:4064@company.com>
```

Notes:

- The Calling Name Manipulation Table for Tel-to-IP Calls table uses the numbers before manipulation, as inputs.
- The LDAP query uses the calling number after source number manipulation, as the search key value.

19.4 Least Cost Routing

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

19.4.1 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 Outbound IP Routing table (Gateway calls) or IP-to-IP Routing table (SBC calls). The device searches this routing table for matching routing rules, and then selects the rule with the lowest call cost. If two routing rules have identical costs, then the rule appearing higher up in the table is used (i.e., first-matched rule). If a selected route is unavailable, the device selects the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched rules with Cost Groups. This is determined according to the settings of the Default Cost parameter in the Routing Rule Groups table.

The Cost Group defines a fixed connection cost (*connection cost*) and a charge per minute (*minute cost*). Cost Groups can also be configured with time segments (*time bands*), which define connection cost and minute cost based on specific days of the week and time of day (e.g., from Saturday through Sunday, between 6:00 and 18:00). If multiple time bands are configured per Cost Group and a call spans multiple time bands, the call cost is calculated using only the time band in which the call was initially established.

In addition to Cost Groups, the device can calculate the call cost using an optional, user-defined average call duration value. The logic in using this option is that a Cost Group may be cheap if the call duration is short, but due to its high minute cost, may prove very expensive if the duration is lengthy. Thus, together with Cost Groups, the device can use this option to determine least cost routing. The device calculates the Cost Group call cost as follows: Total Call Cost = Connection Cost + (Minute Cost * Average Call Duration).

The below table shows an example of call cost when taking into consideration call duration. This example shows four defined Cost Groups and the total call cost if the average call duration is 10 minutes:

Table 19-8: Call Cost Comparison between Cost Groups for different Call Durations

Cost Group	Connection Cost	Minute Cost	Total Call Cost per Duration	
			1 Minute	10 Minutes
A	1	6	7	61
B	0	10	10	100
C	0.3	8	8.3	80.3
D	6	1	7	**16**

If four matching routing rules are located in the routing table and each one is assigned a different Cost Group as listed in the table above, then the rule assigned Cost Group "D" is selected. Note that for one minute, Cost Groups "A" and "D" are identical, but due to the average call duration, Cost Group "D" is cheaper. Therefore, average call duration is an important factor in determining the cheapest routing role.

Below are a few examples of how you can implement LCR:

- **Example 1:** This example uses two different Cost Groups for routing local calls and international calls:

  Two Cost Groups are configured as shown below:

Cost Group	Connection Cost	Minute Cost
1. "Local Calls"	2	1
2. "International Calls"	6	3

The Cost Groups are assigned to routing rules for local and international calls:

Routing Index	Dest Phone Prefix	Destination IP	Cost Group ID
1	2000	x.x.x.x	1 "Local Calls"
2	00	x.x.x.x	2 "International Calls"

- **Example 2:** This example shows how the device determines the cheapest routing rule in the Outbound IP Routing table:

  The Default Cost parameter (global) in the Routing Rule Groups table is set to **Min**, meaning that if the device locates other matching LCR routing rules (with Cost Groups assigned), the routing rule without a Cost Group is considered the lowest cost route.

  - The following Cost Groups are configured:

Cost Group	Connection Cost	Minute Cost
1. "A"	2	1
2. "B"	6	3

- The Cost Groups are assigned to routing rules:

Routing Index	Dest Phone Prefix	Destination IP	Cost Group
1	201	x.x.x.x	"A'
2	201	x.x.x.x	"B"
3	201	x.x.x.x	0
4	201	x.x.x.x	"B"

The device calculates the optimal route in the following index order: 3, 1, 2, and then 4, due to the following logic:

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

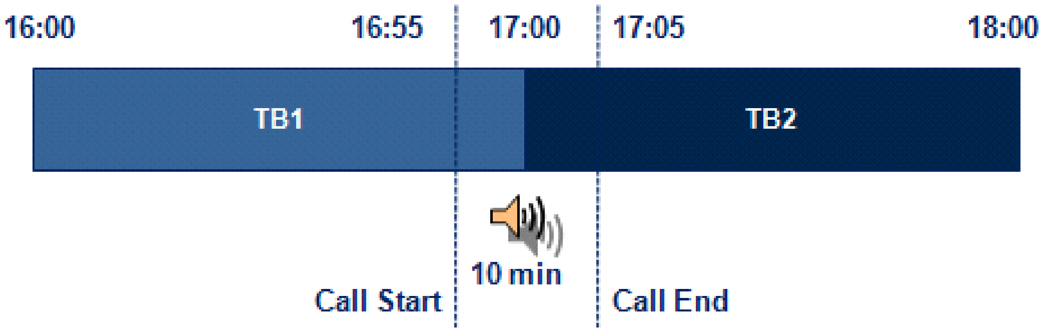
- **Example 3:** This example shows how the cost of a call is calculated if the call spans over multiple time bands:

  Assume a Cost Group, "CG Local" is configured with two time bands, as shown below:

Cost Group	Time Band	Start Time	End Time	Connection Cost	Minute Cost
CG Local	TB1	16:00	17:00	2	1
	TB2	17:00	18:00	7	2

Assume that the call duration is 10 minutes, occurring between 16:55 and 17:05. In other words, the first 5 minutes occurs in time band "TB1" and the next 5 minutes occurs in "TB2", as shown below:

Figure 19-25: 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

19.4.2 Configuring LCR

The following main steps need to be done to configure LCR:

1. Enable the LCR feature and configure the average call duration and default call connection cost - see "Enabling LCR and Configuring Default LCR" on page 236.
2. Configure Cost Groups - see "Configuring Cost Groups" on page 238.
3. Configure Time Bands for a Cost Group - see "Configuring Time Bands for Cost Groups" on page 239.
4. Assign Cost Groups to outbound IP routing rules - see "Assigning Cost Groups to Routing Rules" on page 240.

19.4.2.1 Enabling the LCR Feature

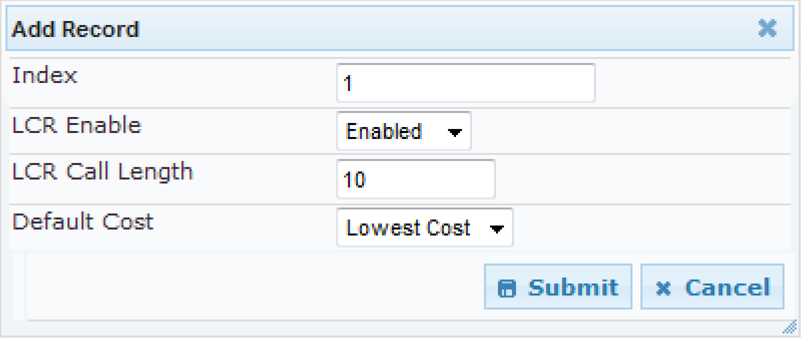
The Routing Rule Groups table lets you enable the LCR feature. This also includes configuring the average call duration and default call cost for routing rules that are not assigned Cost Groups in the Outbound IP Routing table.

The following procedure describes how to enable LCR in the Web interface. You can also do this using the table ini file parameter, RoutingRuleGroups or CLI command, configure voip > services least-cost-routing routing-rule-groups.

➢ **To enable LCR:**

1. Open the Routing Rule Groups Table page (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Routing Rule Groups Table**).
2. Click **Add**; the following dialog box appears:

Figure 19-26: Routing Rule Groups Table - Add Record



Add Record

Index	1
LCR Enable	Enabled
LCR Call Length	10
Default Cost	Lowest Cost

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

Table 19-9: Routing Rule Groups Table Parameter Descriptions

Parameter	Description
Index **[RoutingRuleGroups_Index]**	Defines an index number for the new table record. **Note:** Only one index entry can be configured.
LCR Enable CLI: lcr-enable **[RoutingRuleGroups_LCREnable]**	Enables the LCR feature: ▪ **[0]** Disabled (default) ▪ **[1]** Enabled
LCR Call Length CLI: lcr-call-length **[RoutingRuleGroups_LCRAverageCallLength]**	Defines the average call duration (in minutes) and is used to calculate the variable portion of the call cost. This is useful, for example, when the average call duration spans over multiple time bands. The LCR is calculated as follows: cost = call connect cost + (minute cost * average call duration) The valid value range is 0-65533. The default is 1. For example, assume the following Cost Groups: ▪ "Weekend A": call connection cost is 1 and charge per minute is 6. Therefore, a call of 1 minute cost 7 units. ▪ "Weekend_ B": call connection cost is 6 and charge per minute is 1. Therefore, a call of 1 minute cost 7 units. Therefore, for calls under one minute, "Weekend A" carries the lower cost. However, if the average call duration is more than one minute, then "Weekend B" carries the lower cost.

Parameter	Description
Default Cost CLI: lcr-default-cost **[RoutingRuleGroups_LCRDefaultCost]**	Determines whether routing rules in the Outbound IP Routing table without an assigned Cost Group are considered a higher cost or lower cost route compared to other matched routing rules that are assigned Cost Groups. ▪ **[0]** Lowest Cost = If the device locates other matching LCR routing rules, this routing rule is considered the lowest cost route and therefore, it is selected as the route to use (default.) ▪ **[1]** Highest Cost = If the device locates other matching LCR routing rules, this routing rule is considered as the highest cost route and therefore, is not used or used only if the other cheaper routes are unavailable. **Note:** If more than one valid routing rule without a defined Cost Group exists, the device selects the first-matched rule.

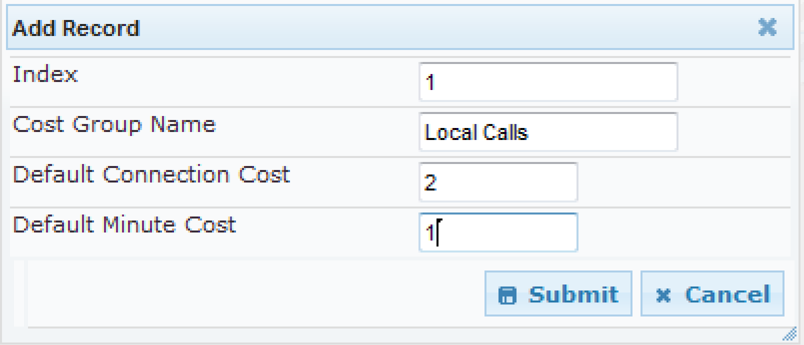
19.4.2.2 Configuring Cost Groups

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

The following procedure describes how to configure Cost Groups in the Web interface. You can also configure this using the table ini file parameter, CostGroupTable or CLI command, configure voip > services least-cost-routing cost-group.

➢ To configure a Cost Group:

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Cost Group Table**).
2. Click **Add**; the following dialog box appears:



Add Record

Index: 1

Cost Group Name: Local Calls

Default Connection Cost: 2

Default Minute Cost: 1

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

Table 19-10: Cost Group Table Parameter Descriptions

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

Parameter	Description
Cost Group Name CLI: cost-group-name **[CostGroupTable_CostGroupName]**	Defines an arbitrary name for the Cost Group. The valid value is a string of up to 30 characters. **Note:** Each Cost Group must have a unique name.
Default Connection Cost CLI:default-connection-cost **[CostGroupTable_DefaultConnectionCost]**	Defines the call connection cost (added as a fixed charge to the call) for a call outside the time bands. The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default connection cost is used.
Default Minute Cost CLI: default-minute-cost **[CostGroupTable_DefaultMinuteCost]**	Defines the call charge per minute for a call outside the time bands. The valid value range is 0-65533. The default is 0. **Note:** When calculating the cost of a call, if the current time of the call is not within a time band configured for the Cost Group, then this default charge per minute is used.

19.4.2.3 Configuring Time Bands for Cost Groups

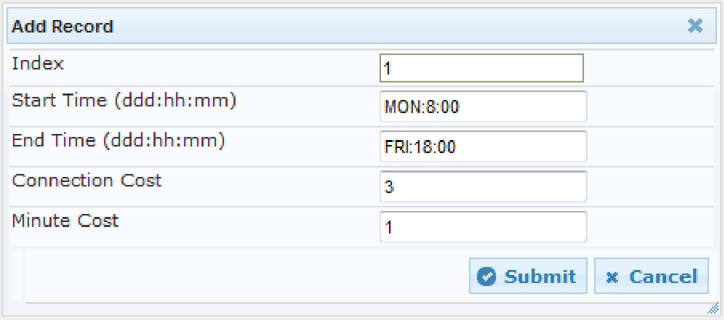
The Time Band table lets you configure Time Bands per Cost Group. A Time Band defines a day and time range (e.g., from Saturday 05:00 to Sunday 24:00), as well as the fixed call connection charge and call rate per minute for this interval. You can configure up to 70 Time Bands, where up to 21 Time Bands can be assigned to each Cost Group.

Note: You cannot configure overlapping Time Bands.

The following procedure describes how to configure Time Bands per Cost Group in the Web interface. You can also configure this using the table ini file parameter, CostGroupTimebands or CLI command, configure voip >services least-cost-routing cost-group-time-bands.

➢ **To configure a Time Band per Cost Group:**

1. Open the Cost Group Table page (**Configuration** tab > **VoIP** menu > **Services** > **Least Cost Routing** > **Cost Group Table**).
2. Select a Cost Group for which you want to assign Time Bands, and then click the **Time Band** link located below the table; the Time Band table for the selected Cost Group appears.
3. Click **Add**; the following dialog box appears:



Add Record

Index	1
Start Time (ddd:hh:mm)	MON:8:00
End Time (ddd:hh:mm)	FRI:18:00
Connection Cost	3
Minute Cost	1

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

Table 19-11: Time Band Table Description

Parameter	Description
Index CLI: timeband-index **[CostGroupTimebands_TimebandIndex]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Start Time CLI: start-time **[CostGroupTimebands_StartTime]**	Defines the day and time of day from when this time band is applicable. The format is DDD:hh:mm, where: ▪ *DDD* is the day of the week, represented by the first three letters of the day in upper case (i.e., SUN, MON, TUE, WED, THU, FRI, or SAT). ▪ *hh* and *mm* denote the time of day, where *hh* is the hour (00-23) and *mm* the minutes (00-59) For example, SAT:22:00 denotes Saturday at 10 pm.
End Time CLI: end-time **[CostGroupTimebands_EndTime]**	Defines the day and time of day until when this time band is applicable. For a description of the valid values, see the parameter above.
Connection Cost CLI: connection-cost **[CostGroupTimebands_ConnectionCost]**	Defines the call connection cost during this time band. This is added as a fixed charge to the call. The valid value range is 0-65533. The default is 0. **Note:** The entered value must be a whole number (i.e., not a decimal).
Minute Cost CLI: minute-cost **[CostGroupTimebands_MinuteCost]**	Defines the call cost per minute charge during this timeband. The valid value range is 0-65533. The default is 0. **Note:** The entered value must be a whole number (i.e., not a decimal).

19.4.2.4 Assigning Cost Groups to Routing Rules

To use your configured Cost Groups, you need to assign them to routing rules:

- Gateway application: Outbound IP Routing table - see Configuring Outbound IP Routing Table on page 397
- SBC application: IP-to-IP Routing table - see Configuring SBC IP-to-IP Routing Rules on page 552

19.5 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 can be configured for any call direction (SBC, Tel-to-IP, or IP-to-Tel). 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 552
- Tel-to-IP routing rules - see Configuring Outbound IP Routing Table on page 397
- IP-to-Tel routing rules - see Configuring Inbound IP Routing Table on page 406

If an incoming call matches the characteristics of a routing rule, the device **first** runs the assigned Call Setup Rules Set ID. The device uses the routing rule to route the call, depending on the result of the Call Setup Rules Set ID:

- **Rule's condition is met:** The device performs the rule's action and then runs the next rule in the Set ID until the last rule or until a rule with an **Exit** Action Type. If the **Exit** rule is configured with a "True" Action Value, the device uses the current routing rule. If the **Exit** rule is configured with a "False" Action Value, the device moves to the next routing rule. If an **Exit** Action Type is not configured and the device has run all the rules in the Set ID, the default Action Value of the Set ID is "True" (i.e., use the current routing rule).
- **Rule's condition is not met:** The device runs the next rule in the Set ID. When the device reaches the end of the Set ID and no **Exit** was performed, the Set ID ends with a "True" result.

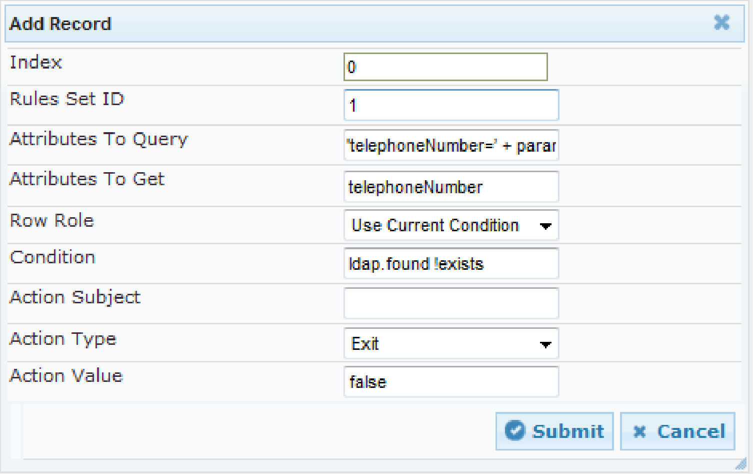
> **Note:** If the source and/or destination numbers are manipulated by the Call Setup rules, they revert to their original values if the device moves to the next routing rule.

The following procedure describes how to configure Call Setup Rules in the Web interface. You can also configure Call Setup Rules using the table ini file parameter, CallSetupRules or CLI command, configure voip/services call-setup-rules.

➢ To configure a Call Setup rule:

1. Open the Call Setup Rules table (**Configuration** tab > **VoIP** menu > **Services** > **Call Setup Rules**).
2. Click **Add**; the following dialog box appears:

Figure 19-27: Call Setup Rules Table - Add Record



Add Record

Index	0
Rules Set ID	1
Attributes To Query	'telephoneNumber=' + parar
Attributes To Get	telephoneNumber
Row Role	Use Current Condition
Condition	ldap.found !exists
Action Subject	
Action Type	Exit
Action Value	false

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

Table 19-12: Call Setup Rules Parameter Descriptions

Parameter	Description
Index **[CallSetupRules_Index]**	Defines an index number for the new table record. **Note:** Each rule must be configured with a unique index.
Rules Set ID CLI: rules-set-id	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

Parameter	Description
[CallSetupRules_RulesSetID]	up to 10 rules. The Set ID is used to assign the Call Setup rules to a routing rule in the routing table. The valid value is 0 to 9. The default is 0.
Attributes To Query CLI: attr-to-query **[CallSetupRules_AttributesToQuery]**	Defines the query string that the device sends to the LDAP server. The valid value is a string of up to 100 characters. Combined strings and values can be configured like in the Message Manipulations table, using the '+' operator. Single quotes (') can be used for specifying a constant string (e.g., '12345'). For example: ▪ 'mobile=' + param.call.dst.user (searches for the AD attribute, "mobile" that has the value of the destination user part of the incoming call) ▪ 'telephoneNumber=' + param.call.redirect + '*' (searches for the AD attribute, "telephoneNumber" that has a redirect number)
Attributes To Get CLI: attr-to-get **[CallSetupRules_AttributesToGet]**	Defines the attributes of the queried LDAP record that the device must handle (e.g., retrieve value). The valid value is a string of up to 100 characters. Up to five attributes can be defined, each separated by a comma (e.g., msRTCSIP-PrivateLine,msRTCSIP-Line,mobile). **Note:** The device saves the retrieved attributes' values for future use in other rules, until the next LDAP query or until the call is connected. Thus, the device does not need to re-query the same attributes.
Row Role CLI: row-role **[CallSetupRules_RowRole]**	Determines which condition must be met in order for this rule to be performed. ▪ **[0]** Use Current Condition = The Condition configured for this rule must be matched in order to perform the configured action (default). ▪ **[1]** Use Previous Condition = The Condition configured for the rule located directly above this rule in the Call Setup table must be matched in order to perform the configured action. This option lets you configure multiple actions for the same Condition.
Condition CLI: condition **[CallSetupRules_Condition]**	Defines the condition that must exist for the device to perform the action. The valid value is a string of up to 200 characters (case-insensitive). Regular Expression (regex) can also be used, for example: ▪ ldap.attr.mobile exists (attribute "mobile" exists in AD) ▪ param.call.dst.user == ldap.attr.msRTCSIP-PrivateLine (called number is the same as the number in the attribute "msRTCSIP-PrivateLine") ▪ ldap.found !exists (LDAP record not found) ▪ ldap.err exists (LDAP error exists)

Parameter	Description
Action Subject CLI: action-subject **[CallSetupRules_ActionSubject]**	Defines the element (header, parameter, or body) upon which you want to perform the action. The valid value is a string of up to 100 characters (case-insensitive). Examples: ▪ header.from contains '1234' (SBC calls only) ▪ param.call.dst.user (called number) ▪ param.call.src.user (calling number) ▪ param.call.src.name (calling name) ▪ param.call.redirect (redirect number) ▪ param.call.src.host (source host) ▪ param.call.dst.host (destination host)
Action Type CLI: action-type **[CallSetupRules_ActionType]**	Defines the type of action to perform. ▪ **[0]** Add (default) = Adds new message header, parameter or body elements. ▪ **[1]** Remove = Removes message header, parameter, or body elements. ▪ **[2]** Modify = Sets element to the new value (all element types). ▪ **[3]** Add Prefix = Adds value at the beginning of the string (string element only). ▪ **[4]** Add Suffix = Adds value at the end of the string (string element only). ▪ **[5]** Remove Suffix = Removes value from the end of the string (string element only). ▪ **[6]** Remove Prefix = Removes value from the beginning of the string (string element only). ▪ **[20]** Exit = Stops the Rule Set ID and returns a result ("True" or "False"). ▪ **[21** Run Rules Set = Performs a different Rule Set ID, specified in the 'Action Value' parameter (below).
Action Value CLI: action-value **[CallSetupRules_ActionValue]**	Defines a value that you want to use in the action. The valid value is a string of up to 300 characters (case-insensitive). Examples: ▪ '+9723976'+ldap.attr.alternateNumber ▪ '9764000' ▪ ldap.attr.displayName ▪ true (if the 'Action Type' is set to **Exit**) ▪ false (if the 'Action Type' is set to **Exit**)

19.5.1 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)

20 Quality of Experience

This chapter describes how to configure the Quality of Experience feature.

20.1 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*.

20.1.1 Configuring the SEM Server

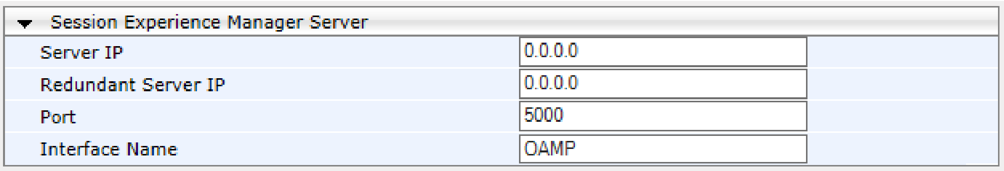
The device can be configured to report QoE voice metrics to a single SEM server or to two SEM/EMS servers deployed in a Geographic Redundancy, High-Availability (HA) mode. Geographic Redundancy is when each SEM/EMS server is located in a different network subnet and has its own IP address. Thus, for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

For regular HA mode, when both 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.

➢ **To configure the SEM server to where the device sends voice metrics:**

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

Figure 20-1: Session Experience Manager Server Page



▼ Session Experience Manager Server

Server IP	0.0.0.0
Redundant Server IP	0.0.0.0
Port	5000
Interface Name	OAMP

2. In the 'Server IP' field, enter the primary SEM server's IP address.
3. If Geographical-Redundancy HA mode exists, in the 'Redundant Server IP' field, enter the secondary SEM server's IP address.
4. In the 'Port' field, enter the port number for the SEM server.
5. In the 'Interface Name' field, enter the device's IP network interface on which the device sends the reports to the SEM server.
6. Click **Submit**, and then save ("burn") your settings to flash memory.

20.1.2 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 121.

20.1.3 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 689. For configuring what to report to the SEM, see "Configuring Quality of Experience Profiles" on page 248.

20.2 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 270), Media Realms (see "Configuring Media Realms" on page 259), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 262).

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 SEM Server" on page 247.
- 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 255.
- 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 561).

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

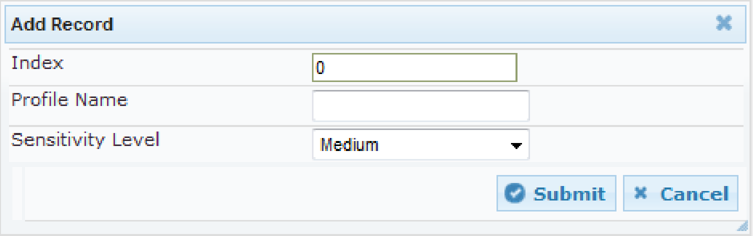
The following procedure describes how to configure Quality of Experience Profiles in the Web interface. You can also configure Quality of Experience Profiles using other management platforms:

- **Quality of Experience Profile table:** Table *ini* file parameter, QoEProfile or CLI command, configure voip/qoe qoe-profile
- **Quality of Experience Color Rules table:** Table *ini* file parameter, QOEColorRules or CLI command, configure voip/qoe qoe-profile qoe-color-rules

➢ To configure a QoE Profile:

1. Open the Quality of Experience Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Quality of Experience Profile**).
2. Click **Add**; the following dialog box appears:

Figure 20-2: Quality of Experience Profile - Add Record



Add Record

Index	0
Profile Name	
Sensitivity Level	Medium

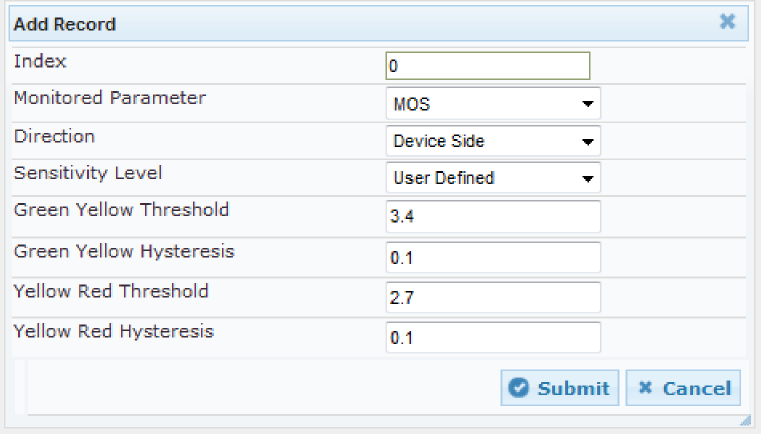
3. Configure a QoE Profile according to the parameters described in the table below.
4. Click **Submit**.

Table 20-1: Quality of Experience Profile Table Parameter Descriptions

Parameter	Description
Index **[QOEProfile_Index]**	Defines an index number for the new table record.
Profile Name CLI: name **[QOEProfile_Name]**	Defines an arbitrary name to easily identify the QoE Profile. The valid value is a string of up to 20 characters.
Sensitivity Level CLI: sensitivity-level **[QOEProfile_SensitivityLevel]**	Defines the pre-configured threshold profile to use. ▪ **[0]** User Defined = Need to define thresholds per monitored parameter in the Quality of Experience Color Rules table. ▪ **[1]** Low = Pre-configured low sensitivity thresholds. ▪ **[2]** Medium = (Default) Pre-configured medium sensitivity thresholds. ▪ **[3]** High = Pre-configured high sensitivity thresholds. Reporting is done for small fluctuations in parameter values.

1. 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.
2. Click **Add**; the following dialog box appears:

Figure 20-3: Quality of Experience Page - Add Record Dialog Box



Add Record

Index: 0
Monitored Parameter: MOS
Direction: Device Side
Sensitivity Level: User Defined
Green Yellow Threshold: 3.4
Green Yellow Hysteresis: 0.1
Yellow Red Threshold: 2.7
Yellow Red Hysteresis: 0.1

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.

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

Table 20-2: Quality of Experience Color Rules Table Parameter Descriptions

Parameter	Description
Index CLI: index **[QOEColorRules_ColorRuleIndex]**	Defines an index number for the new table record.

Parameter	Description
Monitored Parameter CLI: monitored-parameter **[QOEColorRules_monitoredParam]**	Defines the parameter to monitor and report. - **[0]** MOS (default) - **[1]** Delay - **[2]** Packet Loss - **[3]** Jitter - **[4]** RERL [Echo]
Direction CLI: direction **[QOEColorRules_direction]**	Defines the monitoring direction. - **[0]** Device Side (default) - **[1]** Remote Side
Sensitivity Level CLI: sensitivity-level **[QOEColorRules_profile]**	Defines the sensitivity level of the thresholds. - **[0]** User Defined = Need to define the thresholds in the parameters described below. - **[1]** Low = 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 CLI: green-yellow-threshold **[QOEColorRules_GreenYellowThreshold]**	Defines the parameter threshold values between Green (good quality) and Yellow (medium quality) states. The valid threshold values are as follows: - MOS values are in multiples of 10. For example, to denote a MOS of 3.2, the value 32 (i.e., 3.2*10) must be entered. - Delay values are in msec. - Packet Loss values are in percentage (%). - Jitter is in msec. - Echo measures the Residual Echo Return Loss (RERL) in dB.
Green Yellow Hysteresis CLI: green-yellow-hysteresis **[QOEColorRules_GreenYellowHysteresis]**	Defines the fluctuation (change) from the value configured for the Green-Yellow threshold. When the threshold is exceeded by this hysteresis, the device sends a report to the SEM indicating this change. **Note:** If the monitored parameter crosses two thresholds at once (e.g., from Green to Red), the device ignores the hysteresis value and reports the call state change to the SEM.

Parameter	Description
Yellow Red Threshold CLI: yellow-red-threshold **[QOEColorRules_YellowRedThreshold]**	Defines the parameter threshold values between Yellow (medium quality) and Red (poor quality) states. The valid threshold values are as follows: ▪ MOS values are in multiples of 10. For example, to denote a MOS of 3.2, the value 32 (i.e., 3.2*10) must be entered. ▪ Delay values are in msec. ▪ Packet Loss values are in percentage (%). ▪ Jitter is in msec. ▪ Echo measures the Residual Echo Return Loss (RERL) in dB.
Yellow Red Hysteresis CLI: yellow-red-hysteresis **[QOEColorRules_YellowRedHysteresis]**	Defines the fluctuation (change) from the value configured for the Yellow-Red threshold. When the threshold is exceeded by this hysteresis value, the device sends a report to the SEM indicating this change. **Note:** If the monitored parameter crosses two thresholds at once (e.g., from Green to Red), the device ignores the hysteresis value and reports the call state change to the SEM.

20.3 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 270), Media Realms (see "Configuring Media Realms" on page 259), and Remote Media Subnets (see "Configuring Remote Media Subnets" on page 262).

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 255.
- 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 561).
- Send an SNMP alarm (acMediaRealmBWThresholdAlarm). The device clears the alarm when bandwidth utilization returns to normal (within the thresholds).

The thresholds of Bandwidth Profiles use the same color-coding as the Quality of Experience Profile:

- **Green-Yellow threshold:** Lower threshold that indicates that the bandwidth exceeded a user-defined percentage of the configured threshold. This is referred to as a "Warning" alarm (i.e., warning you that bandwidth is nearing the threshold). When bandwidth goes over the threshold, the device considers it as a Yellow state; when it goes below the threshold, it considers it as a Green state.
- **Yellow-Red threshold:** Indicates that bandwidth has exceeded the configured threshold. When bandwidth goes over the threshold, the device considers it as a Red state; when it goes below the threshold, it considers it as a Yellow state.

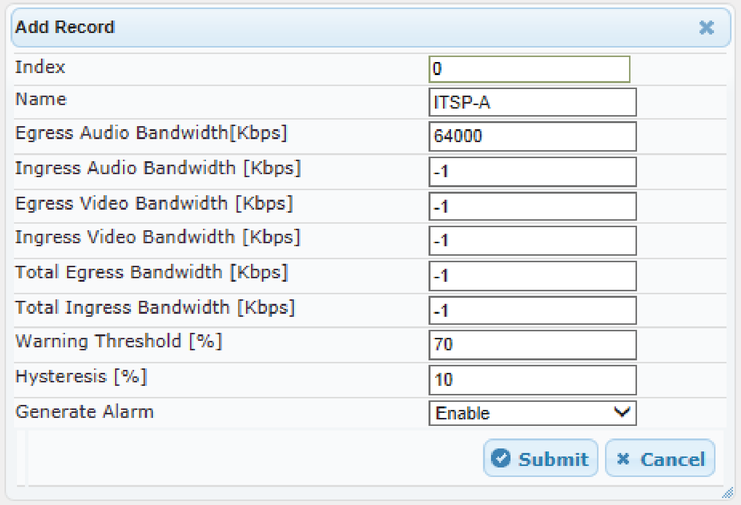
Hysteresis is also used to configure the threshold. This defines the amount of fluctuation from a threshold in order for the threshold to be considered as crossed (i.e., change in color state). Hysteresis is used to avoid false reports.

The following procedure describes how to configure Bandwidth Profiles in the Web interface. You can also configure Bandwidth Profiles using the table *ini* file parameter, BWProfile or CLI command, configure voip/qoe bw-profile.

➢ To configure Bandwidth Profiles:

1. Open the Bandwidth Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Bandwidth Profile**).
2. Click **Add**; the following dialog box appears:

Figure 20-4: Bandwidth Profile Page - Add Record



Add Record

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

The figure above shows a configuration example where if the outgoing voice traffic threshold of 64,000 increases by 80% (70% warning threshold plus 10% hysteresis) to 115,200 (64,000 plus 51,200), a Yellow state occurs and an alarm is sent. If the threshold increases by 10%, a Red state occurs and an alarm is sent.

3. Configure a Bandwidth Profile according to the parameters described in the table below.
4. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

Table 20-3: Bandwidth Profile Table Parameter Descriptions

Parameter	Description
Index **[BWProfile_Index]**	Defines the index of the table row entry.
Name CLI: name **[BWProfile_Name]**	Defines an arbitrary name to easily identify the Bandwidth Profile. The valid value is a string of up to 20 characters.
Egress Audio Bandwidth CLI: egress-audio-bandwidth **[BWProfile_EgressAudioBandwidth]**	Defines the outgoing audio traffic threshold (in Kbps).
Ingress Audio Bandwidth CLI: ingress-audio-bandwidth **[BWProfile_IngressAudioBandwidth]**	Defines the incoming audio traffic threshold (in Kbps).

Parameter	Description
Egress Video Bandwidth CLI: egress-video-bandwidth **[BWProfile_EgressVideoBandwidth]**	Defines the outgoing video traffic threshold (in Kbps).
Ingress Video Bandwidth CLI: ingress-video-bandwidth **[BWProfile_IngressVideoBandwidth]**	Defines the incoming video traffic threshold (in Kbps).
Total Egress Bandwidth CLI: total-egress-bandwidth **[BWProfile_TotalEgressBandwidth]**	Defines the total (video and audio) outgoing bandwidth threshold (in Kbps).
Total Ingress Bandwidth CLI: total-ingress-bandwidth **[BWProfile_TotalIngressBandwidth]**	Defines the total (video and audio) incoming bandwidth threshold (in Kbps).
Warning Threshold CLI: warning-threshold **[BWProfile_WarningThreshold]**	Defines the threshold (in percentage) of the bandwidth thresholds that if exceeded is considered a Warning alarm (Green-Yellow threshold). This applies to any of the configured bandwidth thresholds. The Hysteresis is also added to this Warning threshold. For example, if set to 70% and the Hysteresis to 10%, when the current outgoing voice traffic exceeds 80% of the configured threshold, the Yellow state occurs and a Warning threshold alarm is sent if 'Generate Alarm' is set to **Enable**.
Hysteresis CLI: hysteresis **[BWProfile_hysteresis]**	Defines the bandwidth fluctuation (change) from the bandwidth threshold value (in percentage). The threshold is considered crossed if bandwidth exceeds the configured threshold plus this hysteresis, and a Red state occurs. For example, assume this parameter is set to 10% and the configured bandwidth threshold is set to 64000 Kbps. If current bandwidth reaches 70,400 Kbps (additional 10%), the threshold is considered crossed.
Generate Alarm CLI: generate-alarms **[BWProfile_GenerateAlarms]**	Enables the generation of an SNMP alarm if the threshold (with the hysteresis) is crossed. ▪ **[0]** Disable (default) ▪ **[1]** Enable If enabled, an alarm is sent if one of the following scenarios occurs: ▪ Warning threshold is exceeded (Warning severity - Yellow threshold). ▪ Any configured bandwidth threshold is exceeded (Major severity - Red threshold).

20.4 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 248) and bandwidth utilization (configured in "Configuring Bandwidth Profiles" on page 252). These profiles contain color-coded thresholds that are used to trigger access control and/or media enhancements.

The Media Enhancement Profile table lets you configure any one of the following actions when a specific color-coded threshold (Green-Yellow or Yellow-Red) is crossed for a specific monitored voice metrics (e.g., MOS) or bandwidth (e.g., Egress Audio Bandwidth):

- Reject new calls until the voice metrics or bandwidth returns to below the threshold. This can be used, for example, to reject new calls when bandwidth threshold is exceeded.
- Use a different IP Profile. For example, if packet loss is detected, the IP Group (to which the Media Enhancement Rule is later assigned) can switch to an IP Profile configured with a higher RTP redundancy level. The ability to use a different IP Profile when call quality or bandwidth thresholds are crossed provides a wide range of options for media enhancement and traffic shaping. For example, it may be used to:
    - switch to a low bit-rate coder,
    - negotiate different p-time (and perform transrating if required),
    - increase RTP redundancy level,
    - or block video calls.
- Accept calls

A Media Enhancement Profile can later be assigned to an IP Group (in the IP Group table). However, when the device analyzes the call and determines whether Media Enhancement Profile should be applied or not, it searches for the "most relevant" Quality of Experience Profile or Bandwidth Profile in the following order: 1) Remote Media Subnet, 2) Media Realm, and then 3) IP Group. Thus, a Media Enhancement Profile associated with a specific IP Group may actually "respond" to Quality of Experience or bandwidth thresholds crossed at the Media Realm or Remote Media Subnet level.

Notes:

- The color-coded threshold is first calculated for the IP Group and only then for the Media Realm. The device uses the "worst" color-coded threshold crossing. For example, if a Media Realm crossed a Green-Yellow threshold and an IP Group a Yellow-Red threshold, the action defined for the Red color state is used.
- The device applies Media Enhancements Profiles on new calls **only**, based on the information gathered from previous and/or currently established calls.

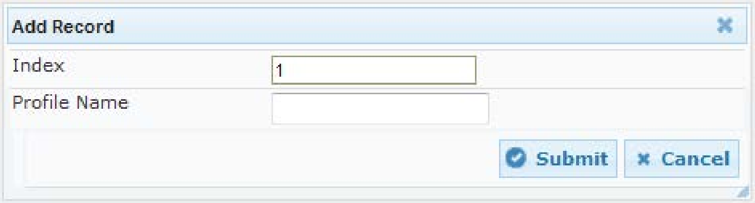
The following procedure describes how to configure Media Enhancement Profiles in the Web interface. You can also configure Media Enhancement Profiles using other management platforms:

- **Media Enhancement Profile table:** Table *ini* file parameter, MediaEnhancementProfile or CLI command, configure voip/qoe media-enhancement
- **Media Enhancement Rules table:** Table *ini* file parameter, MediaEnhancementRules or CLI command, configure voip/qoe media-enhancement-rules

➢ To configure a Media Enhancement Profile:

1. Open the Media Enhancement Profile page (**Configuration** tab > **VoIP** menu > **Quality of Experience** > **Media Enhancement Profile**).
2. Click **Add**; the following dialog box appears:

Figure 20-5: Media Enhancement Profile Table - Add Record



Add Record

Index: 1

Profile Name:

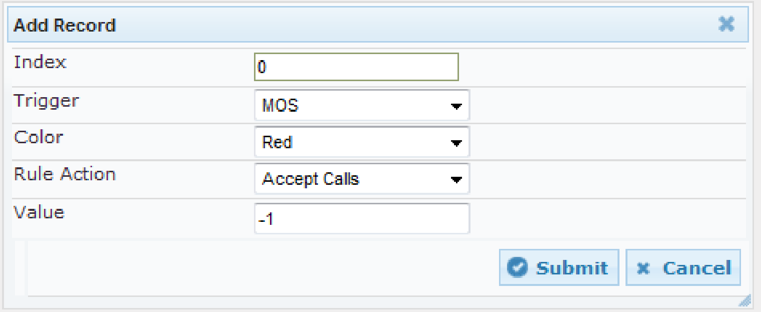
3. Configure a Media Enhancement Profile according to the parameters described in the table below.
4. Click **Submit**.

Table 20-4: Media Enhancement Profile Table Parameter Descriptions

Parameter	Description
Index **[MediaEnhancementProfile_Index]**	Defines the index of the table row entry.
Name CLI: profile-name **[MediaEnhancementProfile_ProfileName]**	Defines an arbitrary name to easily identify the Media Enhancement Profile. The valid value is a string of up to 20 characters.

5. In the Media Enhancement Profile table, select the required Media Enhancement Profile index row, and then click the **Media Enhancement Rules** link located below the table; the Media Enhancement Rules page appears.
6. Click **Add**; the following dialog box appears:

Figure 20-6: Media Enhancement Rules - Add Record



Add Record

Index: 0

Trigger: MOS

Color: Red

Rule Action: Accept Calls

Value: -1

7. Configure a Media Enhancement Rule according to the parameters described in the table below.
8. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

Table 20-5: Media Enhancement Rules Table Parameter Descriptions

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

Parameter	Description
Trigger CLI: trigger **[MediaEnhancementRules_Trigger]**	Defines the monitored metrics parameter or bandwidth associated with this rule. ▪ **[0]** MOS (default) ▪ **[1]** Delay ▪ **[2]** Packet Loss ▪ **[3]** Jitter ▪ **[4]** Bandwidth
Color CLI: color **[MediaEnhancementRules_Color]**	Defines the color-coded threshold change of the monitored metrics or bandwidth (configured in the 'Trigger' parameter) for which this rule is done. ▪ **[0]** Red (default) = Yellow-to-Red threshold is crossed. ▪ **[1]** Yellow = Green-to-Yellow threshold is crossed.
Rule Action CLI: action-rule **[MediaEnhancementRules_ActionRule]**	Defines the action that the device performs when the color-coded threshold is crossed: ▪ **[0]** Accept Calls (default) ▪ **[1]** Reject Calls ▪ **[2]** Alternative IP Profile = An alternative IP Profile ID is used, as configured in the 'Value' field (below). **Notes:** ▪ If this parameter is set to a restrictive action (i.e., **Reject Calls** or **Alternative IP Profile**) for Yellow and no action is set for Red, the device also applies the Yellow action to Red, if this color-coded threshold occurs. ▪ If this parameter is set to a permissive action (i.e., **Accept Calls**) for Red and no action is set for Yellow, the device applies the same action to Yellow, if this color-coded threshold occurs.
Value CLI: value **[MediaEnhancementRules_ActionValue]**	Defines an alternative IP Profile ID for the IP Group that is associated with this rule, if this rule is applied. This parameter is applicable only if the 'Rule Action' parameter is set to **Alternative IP Profile**.

This page is intentionally left blank.

21 Control Network

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

21.1 Configuring Media Realms

The Media Realm table lets you configure a pool of up to 64 SIP media interfaces, termed *Media Realms*. Media Realms allow you to divide a Media-type interface (configured in the Interface table) into several realms, where each realm is specified by a UDP port range. Media Realms also define the maximum number of permitted media sessions. Media Realms can later be assigned to IP Groups (see "Configuring IP Groups" on page 270) and SRDs (see "Configuring SRDs" on page 264).

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 248.
- **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 252.

You can also configure remote destination subnets per Media Realm and assign each subnet a Quality of Experience Profile and Bandwidth Profile. For configuring Remote Media Subnets, see "Configuring Remote Media Subnets" on page 262.

Notes:

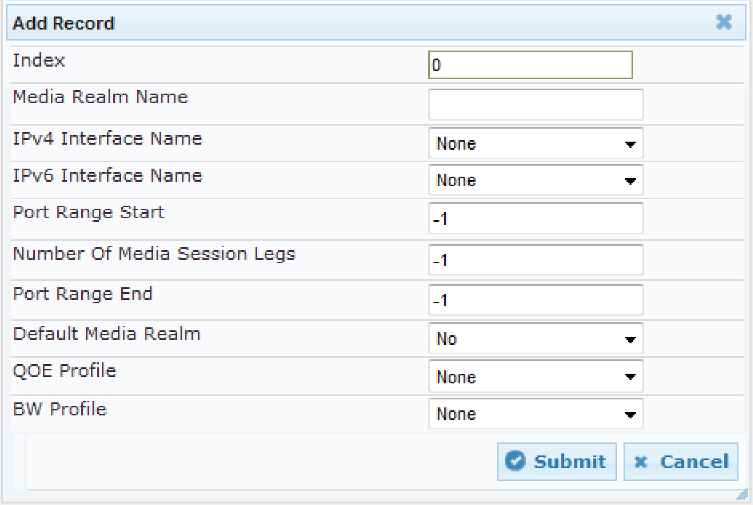
- If an IP Group is associated with an SRD and different Media Realms are assigned to the IP Group and SRD, the IP Group's Media Realm takes precedence.
- If you modify a Media Realm currently being used by a call, the device does not perform Quality of Experience for the call. If you delete the Media Realm during the call, the device maintains the call until the call parties end the call.

The following procedure describes how to configure Media Realms in the Web interface. You can also configure Media Realms using the table ini file parameter, CpMediaRealm or CLI command, configure voip/voip-network realm.

➢ To configure a Media Realm:

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

Figure 21-1: Media Realm Page - Add Record Dialog Box



Add Record

Index	0
Media Realm Name	
IPv4 Interface Name	None
IPv6 Interface Name	None
Port Range Start	-1
Number Of Media Session Legs	-1
Port Range End	-1
Default Media Realm	No
QOE Profile	None
BW Profile	None

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

Table 21-1: Media Realm Table Parameter Descriptions

Parameter	Description
Index **[CpMediaRealm_Index]**	Defines an index number for the new table record. The valid value is 0 to 63.
Media Realm Name CLI: name **[CpMediaRealm_MediaRealmName]**	Defines an arbitrary name to easily identify the Media Realm. The valid value is a string of up to 40 characters. **Notes:** ▪ This parameter is mandatory. ▪ The name assigned to the Media Realm must be unique.
IPv4 Interface Name CLI: ipv4 **[CpMediaRealm_IPv4IF]**	Assigns an IPv4 network interface to the Media Realm. This is the name of the interface as configured in the 'Interface Name' field of the Interface table.
IPv6 Interface Name CLI: ipv6if [CpMediaRealm_IPv6IF]	Assigns an IPv6 network interface to the Media Realm. This is the name of the interface as configured for the 'Interface Name' field of the Interface table.
Port Range Start CLI: port-range-start **[CpMediaRealm_PortRangeStart]**	Defines the starting port for the range of Media interface UDP ports. **Notes:** ▪ You must either configure all Media Realms with port ranges or all without; not some with and some without. ▪ The available UDP port range is according to the BaseUDPport parameter. For more information, see "Configuring RTP Base UDP Port" on page 194. ▪

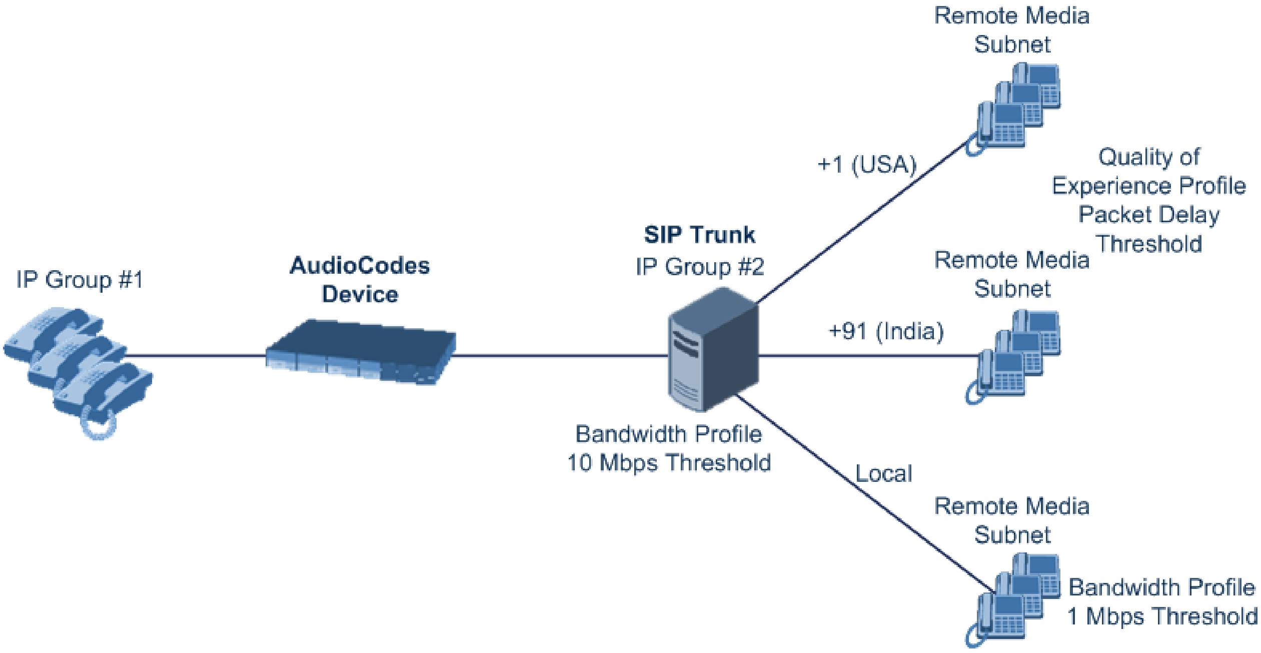
Parameter	Description
Number of Media Session Legs CLI: session-leg **[CpMediaRealm_MediaSessionLeg]**	Defines the number of media sessions associated with the range of ports. This is the number of media sessions available in the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.
Port Range End CLI: port-range-end **[CpMediaRealm_PortRangeEnd]**	(Read-only field) Displays the ending port for the range of media interface UDP ports. This field is calculated by adding the 'Media Session Leg' field (multiplied by the port chunk size) to the 'Port Range Start' field. A value appears once a row has been successfully added to the table.
Default Media Realm CLI: is-default **[CpMediaRealm_IsDefault]**	Defines the Media Realm as the default Media Realm. This default Media Realm is used when no Media Realm is configured for an IP Group or SRD for a specific call. ▪ **[0]** No (default) ▪ **[1]** Yes **Notes:** ▪ This parameter can be set to Yes for only **one** defined Media Realm. ▪ If the parameter is not configured, then the first Media Realm in the table is used as default. ▪ If the table is not configured, the default Media Realm includes all the configured media interfaces.
QoE Profile CLI: qoe-profile **[CpMediaRealm_QoeProfile]**	Assigns a QoE Profile to the Media Realm. For configuring QoE Profiles, see "Configuring Quality of Experience Profiles" on page 248.
BW Profile CLI: bw-profile **[CpMediaRealm_BWProfile]**	Assigns a Bandwidth Profile to the Media Realm. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 252.

21.2 Configuring Remote Media Subnets

Remote Media Subnets define destination subnets for media (RTP/SRTP) traffic on a specific Media Realm. Each Remote Media Subnet can be assigned different call quality (Quality of Experience Profile) and bandwidth utilization (Bandwidth Profile) profiles. These profiles are configured in "Configuring Quality of Experience Profiles" on page 248 and "Configuring Bandwidth Profiles" on page 252, 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 21-2: Remote Media Subnets Example



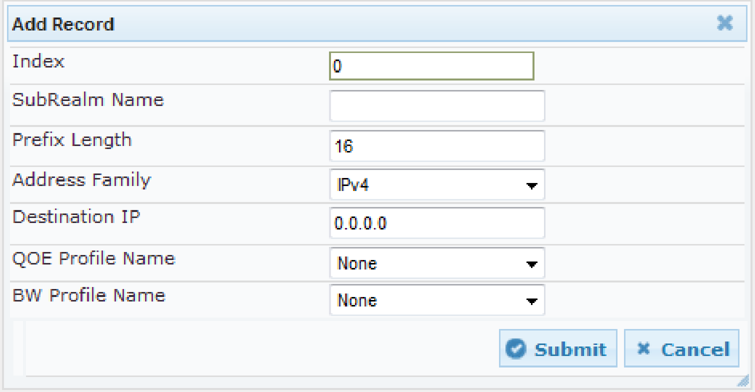
The following procedure describes how to configure Remote Media Subnets in the Web interface. You can also configure Remote Media Subnets using the table *ini* file parameter, RemoteMediaSubnet or CLI command, configure voip > voip-network realm remotemediasubnet.

➢ **To configure a Remote Media Subnet:**

1. Open the Media Realm Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **Media Realm Configuration**).
2. Select the Media Realm index row for which you want to add Remote Media Subnets, and then click the **Remote Media Subnet** link located below the table; the Remote Media Subnet table appears.

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

Figure 21-3: Remote Media Subnet - Add Record



Index	0
SubRealm Name	
Prefix Length	16
Address Family	IPv4
Destination IP	0.0.0.0
QOE Profile Name	None
BW Profile Name	None

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

Table 21-2: Remote Media Subnet Table Parameter Descriptions

Parameter	Description
Index **[RemoteMediaSubnet_RemoteMediaSubnetIndex]**	Defines an index number for the new table record.
Sub-Realm Name CLI: name **[RemoteMediaSubnet_RemoteMediaSubnetName]**	Defines an arbitrary name to easily identify the Remote Media Subnet. The valid value is a string of up to 20 characters.
Prefix Length CLI: prefix-length **[RemoteMediaSubnet_PrefixLength]**	Defines the subnet mask in Classless Inter-Domain Routing (CIDR) notation. For example, 16 denotes 255.255.0.0. The default is 16.
Address Family CLI: address-family **[RemoteMediaSubnet_AddressFamily]**	Defines the IP address protocol. ▪ **[2]** IPv4 Manual (default) ▪ **[10]** IPv6 Manual
Destination IP CLI: dst-ip-address **[RemoteMediaSubnet_DstIPAddress]**	Defines the IP address of the destination. The default is 0.0.0.0.
QOE Profile Name CLI: qoe-profile **[RemoteMediaSubnet_QOEProfileName]**	Assigns a Quality of Experience Profile to the Remote Media Subnet.
BW Profile Name CLI: bw-profile **[RemoteMediaSubnet_BWProfileName]**	Assigns a Bandwidth Profile to the Remote Media Subnet.

21.3 Configuring SRDs

The SRD table lets you configure up to 32 signaling routing domains (SRD). An SRD represents a logical VoIP network. Each logical or physical connection requires an SRD. For example, if the device interfaces with both the LAN and WAN, you would need to configure an SRD for each one.

The SRD is composed of the following:

- SIP Interface: The SIP Interface defines a listening port and transport type (e.g., TLS) for SIP signaling traffic on a specific logical IP network interface of the device.
- Media Realm: The Media Realm defines a UDP port range for RTP (media) traffic on a specific logical IP network interface of the device.

An SRD is a set of definitions together creating multiple, virtual multi-service IP gateways:

- Multiple and different SIP signaling interfaces (SRD associated with a SIP Interface) and RTP media (associated with a Media Realm) for multiple Layer-3 networks. Due to the B2BUA nature of the SBC application, different interfaces can be assigned to each leg of the call, and between the LAN and WAN side.
- Can operate with multiple gateway customers that may reside either in the same or in different Layer-3 networks as the device. This allows separation of signaling traffic between different customers. In such a scenario, the device is configured with multiple SRD's.

Typically, one SRD is defined for each SIP entity (e.g. proxies, IP phones, application servers, gateways, and softswitches) that communicate with each other. This provides these entities with VoIP services that reside on the same Layer-3 network (must be able to communicate without traversing NAT devices and must not have overlapping IP addresses). Routing from one SRD to another is possible, whereby each routing destination (IP Group or destination address) indicates the SRD to which it belongs.

Once configured, you can use the SRD as follows:

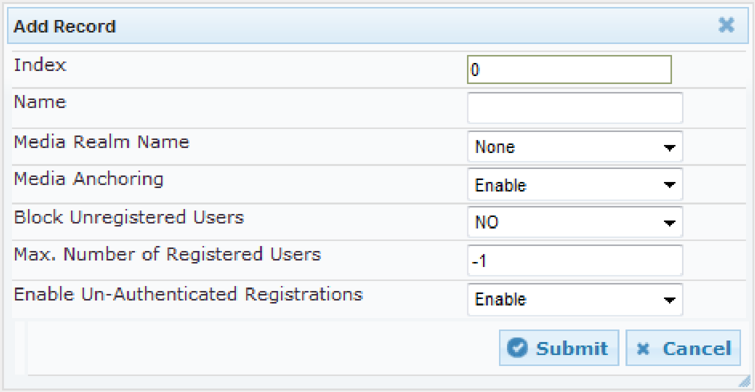
- Associate it with a SIP Interface (see "Configuring SIP Interfaces" on page 267)
- Associate it with an IP Group (see "Configuring IP Groups" on page 270)
- Associate it with a Proxy Set (see "Configuring Proxy Sets" on page 282)
- Associated it with an Admission Control rule (see Configuring Admission Control Table on page 539)
- Define it as a Classification rule for incoming SIP requests (see "Configuring Classification Rules" on page 545)
- Use it as a destination IP-to-IP routing rule (see Configuring Outbound IP Routing Table on page 397 for the Gateway application and Configuring SBC IP-to-IP Routing Rules on page 552 for the SBC application)

The following procedure describes how to configure SRDs in the Web interface. You can also configure this using the table ini file parameter, SRD or CLI command, configure voip > voip-network srd.

➢ **To configure an SRD:**

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

Figure 21-4: SRD Settings Page



Add Record

Index	0
Name	
Media Realm Name	None
Media Anchoring	Enable
Block Unregistered Users	NO
Max. Number of Registered Users	-1
Enable Un-Authenticated Registrations	Enable

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

Table 21-3: SRD Table Parameter Descriptions

Parameter	Description
Index **[SRD_Index]**	Defines an index for the new table record.
SRD Name CLI: name **[SRD_Name]**	Defines an arbitrary name to easily identify the SRD. The valid value can be a string of up to 21 characters. **Note:** This parameter is mandatory.
Media Realm Name CLI: media-realm **[SRD_MediaRealm]**	Assigns a Media Realm to the SRD. The listed Media Realms are the identifiable names that you configured for the Media Realms in the 'Media Realm Name' field of the Media Realm table (see "Configuring Media Realms" on page 259). **Note:** If the Media Realm is later deleted from the Media Realm table, this value becomes invalid in the SRD table.

Parameter	Description
Media Anchoring CLI: intra-srd-media-anchoring [SRD_IntraSRDMediaAnchoring]	Enables the Media Anchoring feature (Anti-Tromboning) per SRD, whereby RTP (media) flows directly between the call parties (i.e., does not traverse the device). ▪ [0] Enable = (Default) RTP traverses the device and each leg uses a different coder or coder parameters. ▪ [1] Disable = The RTP packet flow does not traverse the device; instead, the two SIP UAs establish a direct RTP/SRTP (media) flow between one another. **Notes:** ▪ If this parameter is enabled and the two call endpoints belong to the same SRD, calls cannot be established if the following scenario exists: a. One of the endpoints is defined as a foreign user (for example, “follow me service”) b. and one endpoint is located on the WAN and the other on the LAN. The reason for this is that in Media Anchoring, the device does not interfere in the SIP signaling such as manipulation of IP addresses, which is necessary for calls between LAN and WAN. ▪ When the global parameter SBCDirectMedia is disabled, Media Anchoring can only be enabled for calls between endpoints belonging to the same SRD. ▪ For more information on Media Anchoring, see No Media Anchoring (Anti-Tromboning) on page 511.
Block Unregistered Users CLI: block-un-reg-users [SRD_BlockUnRegUsers]	Determines whether the device blocks (rejects) incoming calls (INVITE requests) from unregistered users (pertaining to User-type IP Groups) for the SRD. ▪ [0] No = Calls from unregistered users are not blocked (default). ▪ [1] Yes = Blocks calls from unregistered users. **Note:** When the call is blocked, the device sends a SIP 500 "Server Internal Error" response to the remote end.
Max. Number of Registered Users CLI: max-reg-users [SRD_MaxNumOfRegUsers]	Maximum number of users belonging to this SRD that can register with the device. By default, no limitation exists for registered users
Enable Un-Authenticated Registrations CLI: enable-un-auth-registrs [SRD_EnableUnAuthenticatedRegistrations]	Determines whether the device blocks REGISTER requests from new users (i.e., users not registered in the device's registration database) when the destination is a User-type IP Group. ▪ [0] No = The device sends REGISTER requests to the SIP proxy server and only if authenticated by the server does the device add the user registration to its database. ▪ [1] Yes = The device adds REGISTER requests to its database even if the requests are not authenticated by a SIP proxy (default).

21.4 Configuring SIP Interfaces

The SIP Interface table lets you configure up to 32 SIP Interfaces. A SIP Interface defines a listening port and type (UDP, TCP, or TLS) for SIP signaling traffic on a specific logical IP network interface (configured in the Interface table). The SIP Interface can be configured for a specific application (i.e., Gateway\IP-to-IP, SAS, SBC) and associated with an SRD. For each SIP Interface, you can assign a SIP message policy rule, assign SIP message manipulation rules, enable TLS mutual authentication, enable TCP keepalive, assign a SSL/TLS certificate (TLS Context), and configure the SIP response sent upon classification failure.

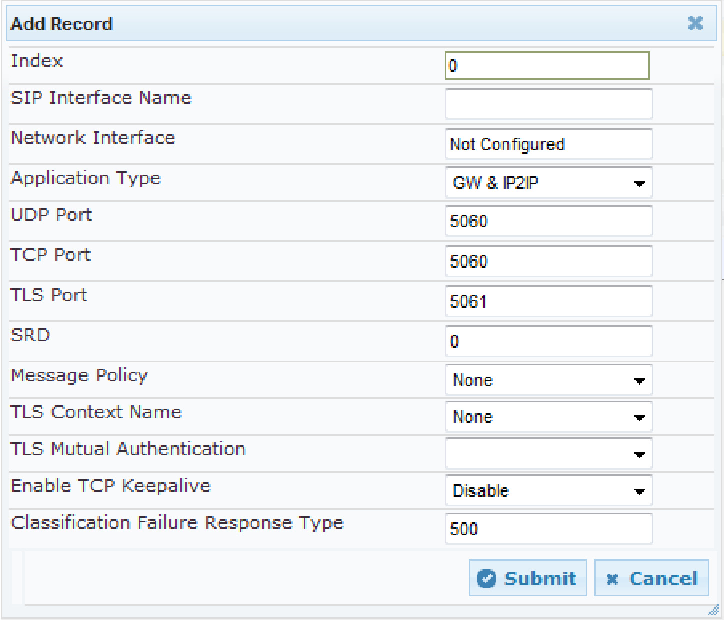
SIP Interfaces can be used, for example, for the following:

- Using SIP signaling interfaces per call leg (i.e., each SIP entity communicates with a specific SRD).
- Using different SIP listening ports for a single or for multiple IP network interfaces.
- Differentiating between applications by creating SIP Interfaces per application.
- Separating signaling traffic between networks (e.g., different customers) to use different routing tables, manipulations, SIP definitions, and so on.

The following procedure describes how to configure SIP interfaces in the Web interface. You can also configure this using the table ini file parameter, SIPInterface or the CLI command, configure voip > voip-network sip-interface.

➢ **To configure a SIP Interface:**

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



Add Record

Index	0
SIP Interface Name	
Network Interface	Not Configured
Application Type	GW & IP2IP
UDP Port	5060
TCP Port	5060
TLS Port	5061
SRD	0
Message Policy	None
TLS Context Name	None
TLS Mutual Authentication	
Enable TCP Keepalive	Disable
Classification Failure Response Type	500

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

Table 21-4: SIP Interface Table Parameter Descriptions

Parameter	Description
Index **[SIPInterface_Index]**	Defines an index for the new table record.
Interface Name CLI: interface-name **[SIPInterface_InterfaceName]**	Defines an arbitrary name to easily identify the SIP Interface. The valid value is a string of up to 21 characters.
Network Interface CLI: network-interface **[SIPInterface_NetworkInterface]**	Assigns a Control-type IP network interface to the SIP Interface. This string value must be identical (case-sensitive) to that configured in the 'Interface Name' field of the Interface table (see "Configuring IP Network Interfaces" on page 130). By default, no value is defined. **Note:** To create a SIP interface on the WAN interface, configure this parameter with the string value, "WAN". This WAN interface is selected in the Interface table (or use the WANInterfaceName parameter), which is the WAN interface address as defined in WAN Access Settings. If VLANs are defined for the WAN interface and one of the VLANs is selected as the VoIP WAN interface, then the defined SIP Interface uses this interface.
Application Type CLI: application-type **[SIPInterface_ApplicationType]**	Defines the application type associated with the SIP Interface. ▪ [0] GW/IP2IP (default) = Gateway / IP-to-IP application. ▪ [1] SAS = Stand-Alone Survivability (SAS) application. ▪ [2] SBC = SBC application.
UDP Port CLI: udp-port **[SIPInterface_UDPPort]**	Defines the listening and source UDP port. The valid range is 1 to 65534. The default is 5060. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
TCP Port CLI: tcp-port **[SIPInterface_TCPPort]**	Defines the listening TCP port. The valid range is 1 to 65534. The default is 5060. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).
TLS Port CLI: tls-port **[SIPInterface_TLSPort]**	Defines the listening TLS port. The valid range is 1 to 65534. The default is 5061. **Notes:** ▪ This port must be outside of the RTP port range. ▪ Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping).

Parameter	Description
SRD CLI: srd **[SIPInterface_SRD]**	Assigns an SRD ID to the SIP Interface (configured in "Configuring SRDs" on page 264). The default is 0. **Notes:** ▪ You can assign the same SRD ID to up to two SIP Interfaces of the same application type. ▪ For the SAS application, use only SRD ID 0. ▪ Each SIP Interface of the same application type (e.g., SBC) that is assigned to the same SRD must be configured with the same IP version (IPv4 or IPv6). ▪ All the SIP Interfaces that are assigned to the same SRD must have the same network interface (assigned in the 'Network Interface' parameter, above).
Message Policy CLI: message-policy **[SIPInterface_MessagePolicy]**	Assigns a SIP message policy to the SIP interface (configured in "Configuring SIP Message Policy Rules").
TLS Context Name CLI: tls-context-name **[SIPInterface_TLSContext]**	Assigns a TLS Context (SSL/TLS certificate) to the SIP Interface. The TLS Context is assigned by name, as configured in the 'Name' field of the TLS Contexts table. ▪ Incoming calls: This TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call or classification to an IP Group based on Proxy Set fails. ▪ Outgoing calls: This TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call. For more information about how certificates are associated with calls and for configuring TLS Contexts, see "Configuring SSL/TLS Certificates" on page 109.
TLS Mutual Authentication CLI: tls-mutual-auth **[SIPInterface_TLSMutualAuthentication]**	Enables TLS mutual authentication for the SIP Interface (when device acts as a server). ▪ **[-1]** Not Configured = (Default) The SIPSRequireClientCertificate global parameter setting is applied. ▪ **[0]** Disable = Device does not request the client certificate for TLS connection on this SIP Interface. ▪ **[1]** Enable = Device requires receipt and verification of the client certificate to establish the TLS connection on this SIP Interface.
Enable TCP Keepalive CLI: tcp-keepalive-enable **[SIPInterface_TCPKeepaliveEnable]**	Enables the TCP Keep-Alive mechanism with the IP entity on this SIP Interface. TCP keep-alive can be used, for example, to keep a NAT entry open for clients located behind a NAT server, or simply to check that the connection to the IP entity is available. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For configuring TCP keepalive, use the following ini file parameters: TCPKeepAliveTime, TCPKeepAliveInterval, and TCPKeepAliveRetry.

Parameter	Description
Classification Failure Response Type CLI: classification_fail_response_type [SIPInterface_ClassificationFailureResponseType]	Defines the SIP response code that the device sends if a received SIP request (OPTIONS, REGISTER, or INVITE) 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:** This parameter is applicable only if the device is set to reject unclassified calls. This is configured using the 'Unclassified Calls' parameter on the General Settings page (Configuration tab > VoIP menu > SBC > General Settings).
Web: Pre Classification ManSet CLI: preclassification-manset [SIPInterface_PreClassificationManipulationSet]	Assigns a Message Manipulation Set ID to the SIP Interface. This lets you apply SIP message manipulation rules on incoming SIP initiating-dialog request messages (not in-dialog), received on this SIP Interface, prior to the Classification process. By default, no Message Manipulation Set ID is defined. For configuring Message Manipulation Sets, see Configuring SIP Message Manipulation on page 299. **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. ▪ This parameter is applicable only to SBC calls.

21.5 Configuring IP Groups

The IP Group table lets you configure up to 50 IP Groups. An IP Group represents a SIP entity in the network with which the device communicates. This can be a server (e.g., IP PBX or ITSP) or it can be a group of users (e.g., LAN IP phones). For servers, the IP Group is typically used to define the server's IP address by associating it with a Proxy Set (see Configuring Proxy Sets on page 282).

IP Groups can be used for the following:

- SBC application: The IP Group table can be used to classify incoming SIP dialog-initiating requests (e.g., INVITE messages) to specific IP Groups. This classification is based on the Proxy Set ID associated with the IP Group. If the source address of the incoming SIP dialog is defined for this Proxy Set, the device assigns the SIP dialog to the associated IP Group. This feature is configured using the 'Classify by Proxy Set' parameter in the IP Group table.

Note: It is highly recommended to use the Classification table for classifying incoming SIP dialogs to IP Groups (see Configuring Classification Rules on page 545).

- SBC application: IP Groups are used for configuring IP-to-IP routing rules where they represent the source and destination of the call (see Configuring SBC IP-to-IP Routing Rules on page 552).
- 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 291).
- Gateway application: Call routing rules:
  - Outgoing IP calls (IP-to-IP or Tel-to-IP): The IP Group identifies the source of the call and is used as the destination of the outgoing IP call (configured in the Outbound IP Routing table). For Tel-to-IP calls, the IP Group (Serving IP Group) can be used as the IP destination to where all SIP dialogs that are initiated from a Trunk Group are sent (configured in Configuring Hunt Group Settings on page 367).
  - Incoming IP calls (IP-to-IP or IP-to-Tel): The IP Group identifies the source of the IP call.
  - The IP Group can be used to associate a number manipulation rule with specific calls identified by IP Group.

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 248.
- **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 252.

Notes:

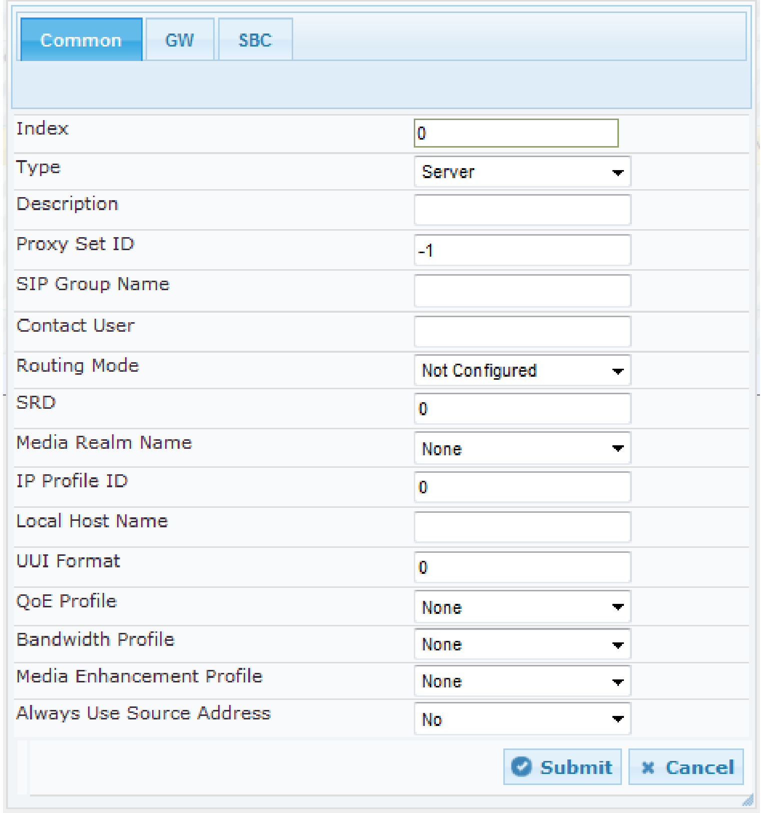
- IP Group ID 0 cannot be used. This IP Group is set to default values and is used by the device when IP Groups are not implemented.
- If different SRDs are configured in the IP Group and Proxy Set tables, the SRD defined for the Proxy Set takes precedence.

The following procedure describes how to configure IP Groups in the Web interface. You can also configure IP Groups using the table ini file parameter, IPGroup or CLI command, configure voip > control-network ip-group.

➢ To configure an IP Group:

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

Figure 21-5: IP Group Table - Add Dialog Box



Index	0
Type	Server
Description	
Proxy Set ID	-1
SIP Group Name	
Contact User	
Routing Mode	Not Configured
SRD	0
Media Realm Name	None
IP Profile ID	0
Local Host Name	
UUI Format	0
QoE Profile	None
Bandwidth Profile	None
Media Enhancement Profile	None
Always Use Source Address	No

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

Table 21-5: IP Group Table Parameter Descriptions

Parameter	Description
Common Parameters	
Index **[IPGroup_Index]**	Defines an index for the new table record.
Type CLI: type [IPGroup_Type]	Defines the type of IP Group: ▪ [0] Server = Used when the destination address, configured by the Proxy Set, of the IP Group (e.g., ITSP, Proxy, IP-PBX, or Application server) is known. ▪ [1] User = Represents a group of users such as IP phones and softphones where their location is dynamically obtained by the device when REGISTER requests and responses traverse (or are terminated) by the device. These users are considered remote (far-end) users. Typically, this IP Group is configured with a Serving IP Group that represents an IP-PBX, Application or Proxy server that serves this User-type IP Group. Each SIP

Parameter	Description
	request sent by a user of this IP Group is proxied to the Serving IP Group. For registrations, the device updates its internal database with the AOR and contacts of the users. Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group. The device forwards these responses directly to the SIP users. To route a call to a registered user, a rule must be configured in the Outbound IP Routing table or SBC IP-to-IP Routing table. The device searches the dynamic database (by using the request URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained from this entry, and a SIP request is sent to the destination. The device also supports NAT traversal for the SIP clients located behind NAT. In this case, the device must be defined with a global IP address. ▪ [2] Gateway = This is applicable only to the SBC application in scenarios where the device receives requests to and from a gateway representing multiple users. This IP Group type is necessary as the other IP Group types are not suitable: ✓ The IP Group cannot be defined as a Server since its destination address is unknown during configuration. ✓ The IP Group cannot be defined as a User since the SIP Contact header of the incoming REGISTER does not represent a specific user. The Request-URI user part can change and therefore, the device is unable to identify an already registered user and therefore, adds an additional record to the database. The IP address of the Gateway IP Group is obtained dynamically from the host part of the Contact header in the REGISTER request received from the IP Group. Therefore, routing to this IP Group is possible only once a REGISTER request is received. If a REGISTER refresh request arrives, the device updates the new location (i.e., IP address) of the IP Group. If the REGISTER fails, no update is performed. If an UN-REGISTER request arrives, the IP address associated with the IP Group is deleted and therefore, no routing to the IP Group is done. **Note:** This field is applicable only to the SBC and IP-to-IP application.
Description CLI: description **[IPGroup_Description]**	Defines a brief description for the IP Group. The valid value is a string of up to 29 characters. The default is an empty field.
Proxy Set ID CLI: proxy-set-id **[IPGroup_ProxySetId]**	Assigns a Proxy Set ID to the IP Group. All INVITE messages destined to this IP Group are sent to the IP address configured for the Proxy Set. **Notes:**

Parameter	Description
	- The Proxy Set is applicable only to Server-type IP Groups. - The SRD configured for this Proxy Set in the Proxy Set table is automatically assigned to this IP Group (see the 'SRD' field below). - To configure Proxy Sets, see "Configuring Proxy Sets" on page 282.
SIP Group Name CLI: sip-group-name **[IPGroup_SIPGroupName]**	Defines the SIP Request-URI host name used in INVITE and REGISTER messages sent to this IP Group, or the host name in the From header of INVITE messages received from this IP Group. The valid value is a string of up to 100 characters. The default is an empty field. **Notes:** - If this parameter is not configured, the value of the global parameter, ProxyName is used instead (see "Configuring Proxy and Registration Parameters" on page 296). - If the IP Group is of User type, this parameter is used internally as a host name in the Request-URI for Tel-to-IP initiated calls. For example, if an incoming call from the device's T1 trunk is routed to a User-type IP Group, the device first creates the Request-URI (<destination_number>@<SIP Group Name>), and then it searches the internal database for a match.
Contact User CLI: contact-user **[IPGroup_ContactUser]**	Defines the user part of the From, To, and Contact headers of SIP REGISTER messages, and the user part of the Contact header of INVITE messages received from this IP Group and forwarded by the device to another IP Group. **Notes:** - This parameter is applicable only to Server-type IP Groups. - This parameter is overridden by the ‘Contact User’ parameter in the ‘Account’ table (see "Configuring Registration Accounts" on page 291).
SRD CLI: srd [IPGroup_SRD]	Assigns an SRD to the IP Group. The default is 0. **Notes:** - For this parameter to take effect, a device reset is required. - To configure SRDs, see Configuring SRDs on page 264. - For Server-type IP Groups, if you assign the IP Group with a Proxy Set ID (in the 'Proxy Set ID' field), the SRD field is automatically set to the SRD value assigned to the Proxy Set in the Proxy Set table.
Media Realm Name CLI: media-realm-name [IPGroup_MediaRealm]	Assigns a Media Realm to the IP Group. The string value must be identical (including case-sensitive) to the Media Realm name defined in the Media Realm table (see Configuring Media Realms on page 259). **Notes:** - For this parameter to take effect, a device reset is

Parameter	Description
	required. - If the Media Realm is deleted from the Media Realm table, this value becomes invalid.
IP Profile ID CLI: ip-profile-id **[IPGroup_ProfileId]**	Assigns an IP Profile to the IP Group. To configure IP Profiles, see "Configuring IP Profiles" on page 314. The default is 0.
Local Host Name CLI: local-host-name **[IPGroup_ContactName]**	Defines the host name (string) that the device uses in the SIP message's Via and Contact headers. This is typically used to define an FQDN as the host name. The device uses this string for Via and Contact headers in outgoing INVITE messages sent to a specific IP Group, and the Contact header in SIP 18x and 200 OK responses for incoming INVITE messages received from a specific IP Group. The Inbound IP Routing table can be used to identify the source IP Group from where the INVITE message was received. If this parameter is not configured (default), these headers are populated with the device's dotted-decimal IP address of the network interface on which the message is sent. **Note:** To ensure proper device handling, this parameter should be a valid FQDN.
QoE Profile CLI: qoe-profile **[IPGroup_QOEProfile]**	Assigns a Quality of Experience Profile rule. For configuring Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 248.
Bandwidth Profile CLI: bandwidth-profile **[IPGroup_BWProfile]**	Assigns a Bandwidth Profile rule. For configuring Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 252.
Media Enhancement Profile CLI: media-enhancement-profile **[IPGroup_MediaEnhancementProfile]**	Assigns a Media Enhancement Profile rule. For configuring Media Enhancement Profiles, see "Configuring Media Enhancement Profiles" on page 255.
Always Use Source Address CLI: always-use-source-addr **[IPGroup_AlwaysUseSourceAddr]**	Enables the device to always send SIP requests and responses, within a SIP dialog, to the source IP address received in the previous SIP message packet. This feature is especially useful in scenarios where the IP Group endpoints are located behind a NAT firewall (and the device is unable to identify this using its regular NAT mechanism). - **[0]** No = (Default) The device sends SIP requests according to the settings of the global parameter, SIPNatDetection. - **[1]** Yes = The device sends SIP requests and responses to the source IP address received in the previous SIP message packet. For information on NAT traversal, see Remote UA behind NAT on page 153.
CLI: Msg-Man-User-Defined-String1 **[IPGroup_MsgManUserDef1]**	Defines a value for the SIP user part that can be used in Message Manipulation rules configured in the Message Manipulations table. The Message Manipulation rule obtains this value from the IP Group, by using the following syntax: param.ipg.<src|dst>.user-defined.<0>. The valid value is a string of up to 30 characters.

Parameter	Description
	For configuring Message Manipulation rules, see "Configuring SIP Message Manipulation" on page 299.
CLI: Msg-Man-User-Defined-String2 **[IPGroup_MsgManUserDef2]**	Defines a value for the SIP user part that can be used in Message Manipulation rules configured in the Message Manipulations table. The Message Manipulation rule obtains this value from the IP Group, by using the following syntax: param.ipg.<src|dst>.user-defined.<1>. The valid value is a string of up to 30 characters. For configuring Message Manipulation rules, see "Configuring SIP Message Manipulation" on page 299.
Gateway Parameters	
Always Use Route Table CLI: always-use-route-table [IPGroup_AlwaysUseRouteTable]	Defines the Request-URI host name in outgoing INVITE messages. ▪ [0] No (default). ▪ [1] Yes = The device uses the IP address (or domain name) defined in the Outbound IP Routing table (see Configuring the Outbound IP Routing Table on page 397) as the Request-URI host name in outgoing INVITE messages, instead of the value configured in the 'SIP Group Name' field. **Note:** This parameter is applicable only to Server-type IP Groups.
Routing Mode CLI: routing-mode [IPGroup_RoutingMode]	Defines the routing mode for outgoing SIP INVITE messages. ▪ [-1] Not Configured = (Default) The routing is according to the selected Serving IP Group. If no Serving IP Group is selected, the device routes the call according to the Outbound IP Routing table (see Configuring Outbound IP Routing Table on page 397). ▪ [0] Routing Table = The device routes the call according to the Outbound IP Routing table. ▪ [1] Serving IP Group = The device sends the SIP INVITE to the selected Serving IP Group. If no Serving IP Group is selected, the default IP Group is used. If the Proxy server(s) associated with the destination IP Group is not alive, the device uses the Outbound IP Routing table (if the parameter IsFallbackUsed is set 1, i.e., fallback enabled - see Configuring Proxy and Registration Parameters on page 296). ▪ [2] Request-URI = The device sends the SIP INVITE to the IP address according to the received SIP Request-URI host name. **Notes:** ▪ This parameter is applicable only to the IP-to-IP application. ▪ This parameter is applicable only to Server-type IP Groups.
SIP Re-Routing Mode CLI: re-routing-mode [IPGroup_SIPReRoutingMode]	Defines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ [-1] Not Configured (Default)

Parameter	Description
	▪ [0] Standard = INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message or Contact header in the 3xx response. ▪ [1] Proxy = Sends a new INVITE to the Proxy. This is applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ [2] Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ When this parameter is set to [1] and the INVITE sent to the Proxy fails, the device re-routes the call according to the Standard mode [0]. ▪ When this parameter is set to [2] and the INVITE fails, the device re-routes the call according to the Standard mode [0]. If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect / Transfer request is rejected. ▪ When this parameter is set to [2], the XferPrefix parameter can be used to define different routing rules for redirected calls. ▪ This parameter is ignored if the parameter AlwaysSendToProxy is set to 1.
Enable Survivability CLI: enable-survivability [IPGroup_EnableSurvivability]	Defines how the device handles registration messages and whether Survivability mode is enabled for User-type IP Groups. ▪ [0] Disable (default). ▪ [1] Enable if Necessary = Survivability mode is enabled only if the Serving IP Group is unavailable. The device saves in its Registration database the registration messages sent by the clients (e.g., IP phones) belonging to the User-type IP Group. If communication with the Serving IP Group (e.g., IP-PBX) fails, the User-type IP Group enters into Survivability mode in which the device uses its database for routing calls between the clients of the User-type IP Group. In Survivability mode, the RTP packets between the clients always traverse through the device, and new registrations can also be processed. When the Serving IP Group is available again, the device returns to normal mode, sending INVITE and REGISTER messages to the Serving IP Group. ▪ [2] Always Enable = Survivability mode is always enabled. The communication with the Serving IP Group is always considered as failed. The device uses its database for routing calls between the clients of the User-type IP Group. ▪ [3] Always Terminate Register = The registration messages received from the clients are saved in the device's registration database without forwarding them to the proxy server. Upon receipt of the registration message, the device returns a SIP 200 OK to the client.

Parameter	Description
	Notes: - This parameter is applicable only to the IP-to-IP application. - This parameter is applicable only to User-type IP Groups.
Serving IP Group ID CLI: serving-ip-group-id [IPGroup_ServingIPGroup]	If configured, INVITE messages initiated from the IP Group are sent to this Serving IP Group (range 1 to 9). In other words, the INVITEs are sent to the address defined for the Proxy Set associated with this Serving IP Group. The Request-URI host name in the INVITE messages are set to the value of the 'SIP Group Name' parameter defined for the Serving IP Group. **Notes:** - This parameter is applicable only to the IP-to-IP application. - If the PreferRouteTable parameter is set to 1, the routing rules in the Outbound IP Routing table take precedence over this 'Serving IP Group ID' parameter. - If this parameter is not configured, the INVITE messages are sent to the default Proxy or according to the 'Outbound IP Routing table.
SBC Parameters	
Classify By Proxy Set CLI: classify-by-proxy-set [IPGroup_ClassifyByProxySet]	Defines whether the incoming INVITE is classified to the IP Group according to its associated Proxy Set. - [0] Disable - [1] Enable (default) This classification occurs only if classification according to the device's database fails to (i.e., received INVITE is not from a registered user). The classification proceeds with checking whether the INVITE's IP address (if host names, then according to the dynamically resolved IP address list) is defined for a Proxy Set ID (in the Proxy Set table). If a Proxy Set ID has such an IP address, the device classifies the INVITE as belonging to the IP Group associated with this Proxy Set. The Proxy Set ID is assigned to the IP Group using the IP Group table's 'Proxy Set ID' parameter (see above). **Notes:** - In cases where multiple IP Groups are associated with the same Proxy Set ID, do not enable this feature. If enabled, the device is unable to correctly classify the incoming INVITEs to their appropriate IP Groups. - To enhance security, it is highly recommended to disable this parameter so that the device can use the Classification table rules to classify the call. If this parameter is enabled, the Classification table is not used if an associated Proxy Set is found. - This parameter is applicable only to Server-type IP Groups.
Max. Number of Registered Users CLI: max-num-of-reg-users [IPGroup_MaxNumOfRegUsers]	Defines the maximum number of users in this IP Group that can register with the device. By default, no limitation exists for registered users.

Parameter	Description
	Note: This field is applicable only to User-type IP Groups.
Inbound Message Manipulation Set CLI: inbound-mesg-manipulation-set [IPGroup_InboundManSet]	Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the inbound message. To configure Message Manipulation rules, see Configuring SIP Message Manipulation on page 299.
Outbound Message Manipulation Set CLI: outbound-mesg-manipulation-set [IPGroup_OutboundManSet]	Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the outbound message. To configure Message Manipulation rules, see Configuring SIP Message Manipulation on page 299.
Registration Mode CLI: registration-mode [IPGroup_RegistrationMode]	Defines the registration mode for the IP Group: ▪ [0] User Initiates Registration (default) ▪ [1] SBC Initiates Registration = Used when the device serves as a client (e.g., with an IP PBX). This functions only with the User Info file. ▪ [2] Registrations not Needed = The device adds users to its database in active state.
Authentication Mode CLI: authentication-mode [IPGroup_AuthenticationMode]	Defines the authentication mode. ▪ [0] User Authenticates = (Default) The device does not handle the authentication, but simply passes the authentication messages between the SIP user agents. ▪ [1] SBC as Client = The device authenticates as a client. It receives the 401/407 response from the proxy requesting for authentication. The device sends the proxy the authorization credentials (i.e., username and password) according to one of the following: 1) account defined in the Account table (only if authenticating Server-type IP Group), 2) global username and password parameters (only if authenticating Server-type IP Group), 3) User Information file, or 4) sends request to users requesting credentials (only if authenticating User-type IP Group). ▪ [2] SBC as Server = The device acts as an Authentication server: ✓ Authenticates SIP clients, using the usernames and passwords in the User Information table (see SBC User Information for SBC User Database on page 618). This is applicable only to User-type IP Groups. ✓ Authenticates SIP severs. This is applicable only to Server-type IP Groups.
Authentication Method List CLI: authentication-method-list [IPGroup_MethodList]	Defines SIP methods received from the IP Group that must be challenged by the device, when the device acts as an Authentication server. If this parameter is not defined (i.e., empty value), no methods are challenged. The default value is null. Multiple entries are separated by a backslash "\", for example, INVITE\REGISTER. **Note:** This parameter is applicable only if the 'Authentication Mode' parameter is set to SBC as Server [2].
SBC Client Forking Mode CLI: enable-sbc-client-forking	Defines call forking of INVITE messages to up to five separate SIP outgoing legs for User-type IP Groups. This occurs if multiple contacts are registered under the same

Parameter	Description
[IPGroup_EnableSBCClientForking]	AOR in the device's registration database. ▪ [0] Sequential = (Default) Sequentially sends the INVITE to each contact. If there is no answer from the first contact, it sends the INVITE to the second contact, and so on until a contact answers. If no contact answers, the call fails or is routed to an alternative destination, if configured. ▪ [1] Parallel = Sends the INVITE simultaneously to all contacts. The call is established with the first contact that answers. ▪ [2] Sequential Available Only = Sequentially sends the INVITE only to available contacts (i.e., not busy). If there is no answer from the first available contact, it sends the INVITE to the second contact, and so on until a contact answers. If no contact answers, the call fails or is routed to an alternative destination, if configured. **Note:** The device can also fork INVITE messages received for a Request-URI of a specific contact (user) registered in the database to all other users located under the same AOR as the specific contact. This is configured using the SBCSendInviteToAllContacts parameter.
Source URI Input CLI: src-uri-input [IPGroup_SourceUriInput]	Defines the SIP header in the incoming INVITE that is used for call matching characteristics based on source URIs. ▪ [-1] Not Configured (default) ▪ [0] From ▪ [1] To ▪ [2] Request-URI ▪ [3] P-Asserted - First Header ▪ [4] P-Asserted - Second Header ▪ [5] P-Preferred ▪ [6] Route ▪ [7] Diversion ▪ [8] P-Associated-URI ▪ [9] P-Called-Party-ID ▪ [10] Contact ▪ [11] Referred-by **Notes:** ▪ This parameter is applicable only when classification is done according to the Classification table. ▪ If the configured SIP header does not exist in the incoming INVITE message, the classification of the message to a source IP Group fails. ▪ If the device receives an INVITE as a result of a REFER request or a 3xx response, then the incoming INVITE is routed according to the Request-URI. The device identifies such INVITEs according to a specific prefix in the Request-URI header, configured by the SBCXferPrefix parameter. Therefore, in this scenario, the device ignores this parameter setting.
Destination URI Input CLI: dst-uri-input [IPGroup_DestUriInput]	Defines the SIP header in the incoming INVITE to use as a call matching characteristic based on destination URIs. The parameter is used for classification and routing purposes.

Parameter	Description
	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. Once classified, the device uses the parameter for routing the call. For example, if set to To, the URI in the To header of the incoming INVITE is used as a matching characteristic for classifying the call to an IP Group in the Classification table. Once classified, the device uses the URI in the To header as the destination. ▪ [-1] Not Configured (default) ▪ [0] From ▪ [1] To ▪ [2] Request-URI ▪ [3] P-Asserted - First Header ▪ [4] P-Asserted - Second Header ▪ [5] P-Preferred ▪ [6] Route ▪ [7] Diversion ▪ [8] P-Associated-URI ▪ [9] P-Called-Party-ID ▪ [10] Contact ▪ [11] Referred-by **Notes:** ▪ The parameter is applicable only when classification is done according to the Classification table. ▪ If the configured SIP header does not exist in the incoming INVITE message, the classification of the message to a source IP Group fails. ▪ If the device receives an INVITE as a result of a REFER request or a 3xx response, the incoming INVITE is routed according to the Request-URI. The device identifies such INVITEs according to a specific prefix in the Request-URI header, configured by the SBCXferPrefix parameter. Therefore, in this scenario, the device ignores this parameter setting.
Username CLI: username [IPGroup_Username]	Defines the shared username for authenticating the IP Group, when the device acts as an Authentication server. The valid value is a string of up to 51 characters. By default, no username is defined. **Notes:** ▪ This parameter is applicable only to Server-type IP Groups and when the 'Authentication Mode' parameter is set to SBC as Server (i.e., authentication of servers). ▪ To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.
Password CLI: password IPGroup_Password]	Defines the shared password for authenticating the IP Group, when the device acts as an Authentication server. The valid value is a string of up to 51 characters. By default, no password is defined.

Parameter	Description
	Notes: ▪ This parameter is applicable only to Server-type IP Groups and when the 'Authentication Mode' parameter is set to SBC as Server (i.e., authentication of servers). ▪ To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.

21.6 Configuring Proxy Sets

The Proxy Sets table lets you configure up to 50 Proxy Sets. A Proxy Set defines the destination address (IP address and/or FQDN) and transport type (e.g., UDP) of a SIP server (e.g., Proxy). Each Proxy Set can be configured with up to 10 addresses configured as an IP address and/or DNS host name (FQDN), enabling you to implement load balancing and redundancy between multiple servers. If you configure the address as an FQDN, you can configure the method (A-record DNS, SRV, or NAPTR) for resolving the domain name to an IP address. The device supports up to 30 DNS-resolved IP addresses. (If the DNS resolution provides more than this number, the device uses the first 30 IP addresses in the received list, and ignores the rest.)

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

Proxy Sets are later assigned to **Server-type** IP Groups, in the IP Group table. When the device sends an INVITE message to an IP Group, it sends it to the address configured for the Proxy Set. You can also enable the classification of incoming SBC SIP dialogs to IP Groups based on Proxy Set. If the source address of the incoming SIP dialog is the same as the address of a Proxy Set that is assigned to an IP Group, the device classifies the SIP dialog as belonging to that IP Group. This feature is configured using the 'Classify by Proxy Set' parameter in the IP Group table. For configuring IP Groups, see "Configuring IP Groups" on page 270.

Note: For classifying incoming SIP dialogs to IP Groups, it is highly recommended to use ONLY the Classification table (see Configuring Classification Rules on page 545).

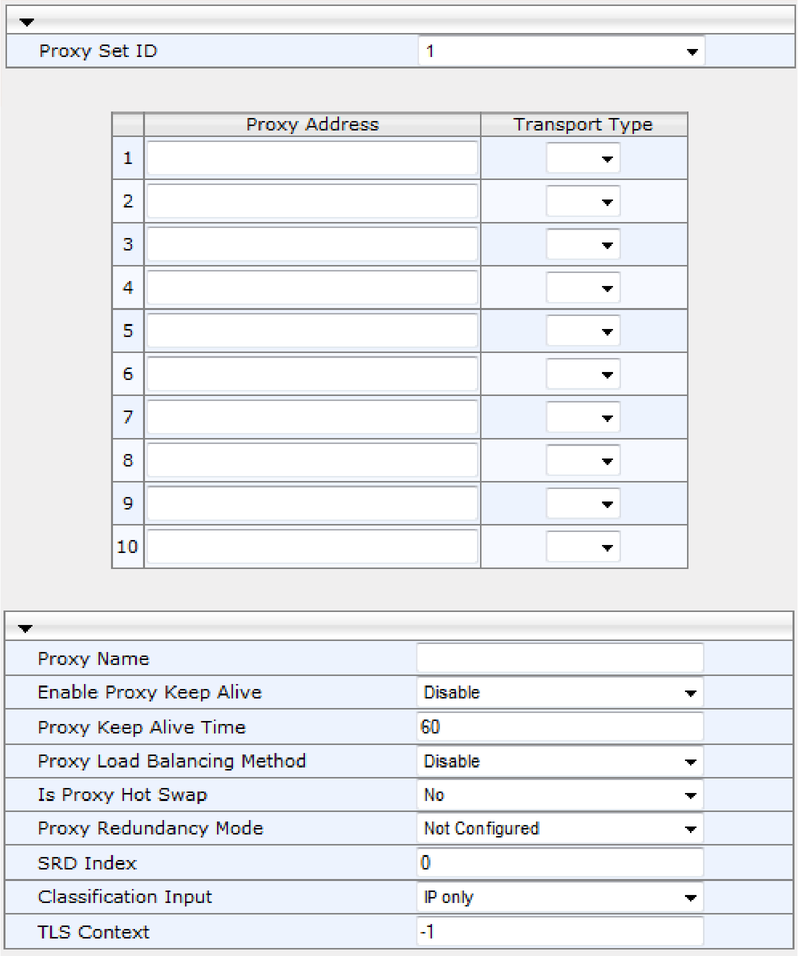
The following procedure describes how to configure Proxy Sets in the Web interface. You can also configure Proxy Sets using the following management tools:

- Proxy Set ID with IP addresses: table ini file parameter, ProxyIP or CLI command, configure voip > voip-network proxy-ip > proxy-set-id
- Attributes for the Proxy Set: table ini file parameter, ProxySet or CLI command, configure voip > voip-network proxy-set

➢ To configure a Proxy Set:

1. Open the Proxy Sets Table page (**Configuration** tab > **VoIP** menu > **VoIP Network** > **Proxy Sets Table**).

Figure 21-6: Proxy Sets Table Page



Proxy Set ID	1

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

Proxy Name	
Enable Proxy Keep Alive	Disable
Proxy Keep Alive Time	60
Proxy Load Balancing Method	Disable
Is Proxy Hot Swap	No
Proxy Redundancy Mode	Not Configured
SRD Index	0
Classification Input	IP only
TLS Context	-1

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

Table 21-6: Proxy Sets Table Parameter Description

Parameter	Description
Web: Proxy Set ID EMS: Index CLI: configure voip > voip-network proxy-set **[ProxySet_Index]**	Defines an index number for the new table record.
Proxy Address CLI: voip-network proxy-ip > proxy-address **[ProxyIp_IpAddress]**	Defines the address of the Proxy server. Up to 10 addresses can be configured per Proxy Set. The address can be defined as an IP address in dotted-decimal notation (e.g., 201.10.8.1) or FQDN. You can also specify the port using the following format: ▪ IPv4 address: <IP address>:<port> (e.g., 201.10.8.1:5060) ▪ IPv6 address: <[IPV6 address]>:<port> (e.g.,

Parameter	Description
	[2000::1:200:200:86:14]:5060)
Transport Type CLI: voip-network proxy-ip > transport-type **[ProxyIp_TransportType]**	Defines the transport type for communicating with the Proxy server. ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS ▪ **[-1]** = Undefined **Note:** If this parameter is not configured, the setting of the global parameter, SIPTransportType is used.
Proxy Name CLI: proxy-name **[ProxySet_ProxyName]**	Defines an arbitrary name to easily identify the Proxy Set. The valid value is a string of up to 20 characters.
DNS Resolve Method CLI: dns-resolve-method **[ProxySet_DNSResolveMethod]**	Defines the DNS query record type for resolving the Proxy server's host name into an IP address. ▪ **[-1]** = DNS resolving is done according to the settings of the global parameter, Proxy DNS Query Type. ▪ **[0]** A-Record = (Default) A-record DNS query. ▪ **[1]** SRV = If the Proxy address is configured with a domain name without a port (e.g., domain.com), an SRV query is done. The SRV query returns the host names (and their weights). The device then performs DNS A-record queries per host name (according to the received weights). If the configured Proxy address contains a domain name with a port (e.g., domain.com:5080), the device performs a regular DNS A-record query. ▪ **[2]** NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the configured Proxy address contains a domain name with a port (e.g., domain.com:5080), the device performs a regular DNS A-record query. If the transport type is configured for the Proxy address, a NAPTR query is not performed. **Note:** An SRV query can return up to four host names. For each host name, the subsequent DNS A-record query can be resolved into up to 15 IP addresses. However, the device supports up to 30 DNS-resolved IP addresses. If the device receives more than this number of DNS-resolved IP addresses, the device uses the first 30 IP addresses in the received list, and ignores the rest.
Web/EMS: Enable Proxy Keep Alive CLI: voip-network proxy-set > proxy-enable-keep-alive **[ProxySet_EnableProxyKeepAlive]**	Enables the device's Proxy Keep-Alive mechanism, which checks communication with the Proxy server. ▪ **[0]** Disable (default). ▪ **[1]** Using Options = Enables the Proxy Keep-Alive mechanism using SIP OPTIONS messages. The device sends these messages every user-defined interval, configured by the 'Proxy Keep Alive Time'

Parameter	Description
	parameter. If the device receives a SIP response code that is also configured in the 'Keep-Alive Failure Responses' parameter (below), the device considers the Proxy as down. You can also configure whether to use the device's IP address or string name ("gateway name") in the OPTIONS message (see the UseGatewayNameForOptions parameter). - **[2]** Using Register = Enables the Proxy Keep-Alive mechanism using SIP REGISTER messages. The device sends the REGISTER message every user-defined interval, configured by the RegistrationTime parameter (Gateway application) or SBCProxyRegistrationTime parameter (SBC application). Any SIP response from the Proxy - success (200 OK) or failure (4xx response) - is considered as if the Proxy is "alive". If the Proxy does not respond to INVITE messages sent by the device, the Proxy is considered as down (offline). If you enable Proxy Keep-Alive mechanism, the device can operate with multiple Proxy servers (addresses) for redundancy and load balancing (configured by the 'Proxy Load Balancing Method' parameter). **Notes:** - For Survivability mode for User-type IP Groups, this parameter must be enabled (1 or 2). - If this parameter is enabled and the Proxy uses the TCP/TLS transport type, you can enable CRLF Keep-Alive mechanism, using the UsePingPongKeepAlive parameter.
Web: Proxy Keep Alive Time EMS: Keep Alive Time CLI: voip-network proxy-set > proxy-keep-alive-time **[ProxySet_ProxyKeepAliveTime]**	Defines the interval (in seconds) between Keep-Alive messages sent by the device when the Keep-Alive mechanism is enabled. The valid range is 5 to 2,000,000. The default is 60. **Note:** This parameter is applicable only if the 'Enable Proxy Keep Alive' parameter is set to **Using Options**.
Web: Keep-Alive Failure Responses CLI: keepalive-fail-resp **[ProxySet_KeepAliveFailureResp]**	Defines SIP response codes that if any is received in response to a keep-alive message using SIP OPTIONS, the device considers the Proxy as down. Up to three response codes can be configured, where each code is separated by a comma (e.g., 407,404). By default, no responses are defined. If no responses are configured or responses received are not those configured, the proxy is considered "alive". **Note:** The SIP 200 response code is not supported by this feature.
Web: Proxy Load Balancing Method EMS: Load Balancing Method CLI: voip-network proxy-set > proxy-load-balancing-method **[ProxySet_ProxyLoadBalancingMethod]**	Enables the Proxy Load Balancing mechanism per Proxy Set. - **[0]** Disable = Load Balancing is disabled (default) - **[1]** Round Robin = A list of all possible Proxy IP addresses is compiled. This list includes all IP addresses per Proxy Set after necessary DNS resolutions (including NAPTR and SRV, if

Parameter	Description
	configured). After this list is compiled, the Proxy Keep-Alive mechanism (according to parameters EnableProxyKeepAlive and ProxyKeepAliveTime) tags each entry as 'offline' or 'online'. Load balancing is only performed on Proxy servers that are tagged as 'online'. All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP addresses list is refreshed according to ProxyIPListRefreshTime. If a change in the order of the entries in the list occurs, all load statistics are erased and balancing starts over again. ▪ **[2]** Random Weights = The outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server, using SRV records. The device sends the requests in such a fashion that each Proxy receives a percentage of the requests according to its' assigned weight. A single FQDN should be configured as a Proxy IP address. Random Weights Load Balancing is not used in the following scenarios: ✓ The Proxy Set includes more than one Proxy IP address. ✓ The only Proxy defined is an IP address and not an FQDN. ✓ SRV is not enabled (DNSQueryType). ✓ The SRV response includes several records with a different Priority value.
Web/EMS: Is Proxy Hot Swap CLI: voip-network proxy-set > is-proxy-hot-swap **[ProxySet_IsProxyHotSwap]**	Enables the Proxy Hot-Swap redundancy mechanism, which provides real-time switching from the primary Proxy server to redundant Proxies when no response is received from the primary. ▪ **[0]** No (default) ▪ **[1]** Yes = The device sends the SIP INVITE/REGISTER message to the first address (Proxy/Registrar server) listed in the Proxy Set. If a SIP response is received and this response code is defined in the 'Keep Alive Failure Response' parameter (above), the device assumes the Proxy as down and sends the message again; otherwise, the device assumes the proxy "alive" and does not send the message again. Each time a defined response code is received, the device re-sends the message. This can occur until a user-defined maximum number of retransmissions, configured by the HotSwapRtx parameter, after which the device sends the same message to the next address (redundant Proxy/Registrar), and so on. If there is no response from any of the Proxies, the device goes through the address list again until a "live" Proxy is located.
Web/EMS: Proxy Redundancy Mode CLI: voip-network proxy-set > proxy-redundancy-mode **[ProxySet_ProxyRedundancyMode]**	Determines whether the device switches from a redundant Proxy to the primary Proxy when it becomes available again. ▪ **[-1]** Not configured = (Default) The global parameter,

Parameter	Description
	ProxyRedundancyMode applies. - **[0]** Parking = The device continues operating with the redundant (now active) Proxy until the next failure, after which it operates with the next redundant Proxy. - **[1]** Homing = The device always attempts to operate with the primary Proxy. The device switches back to the primary Proxy whenever it becomes available. **Notes:** - To enable this functionality, you must also enable the Proxy Keep-Alive mechanism (using the 'Enable Proxy Keep Alive' parameter). - The **Homing** option can only be used if the 'Enable Proxy Keep Alive' parameter is set to **Using Options.**
Web/EMS: SRD Index CLI: voip-network proxy-set > srd-id [ProxySet_ProxySet_SRD]	Assigns an SRD to the Proxy Set ID. The default is SRD 0. **Notes:** - For this parameter to take effect, a device reset is required. - To configure SRDs, see Configuring SRDs on page 264.
Web/EMS: Classification Input CLI: voip-network proxy-set > classification-input [ProxySet_ClassificationInput]	Defines how the device classifies IP calls to the Proxy Set. - [0] IP Only = (Default) The call is classified to the Proxy Set according to its IP address only. - [1] IP + Port + Transport = The call is classified to the Proxy Set according to its IP address, port, and transport type. **Notes:** - This parameter is applicable only if the IP Group table's parameter, 'Classify by Proxy Set' is set to Enable. - This parameter is applicable only to the SBC application.

Parameter	Description
Web/EMS: TLS Context Index CLI: tls-context-index **[ProxySet_TLSContext]**	Assigns a TLS Context (SSL/TLS certificate) to the Proxy Set. The TLS Context is assigned by index number, as configured in the TLS Contexts table. ▪ **Incoming calls:** If the 'Transport Type' parameter (above) is set to **TLS** and the incoming call is successfully classified to an IP Group based on this Proxy Set, this TLS Context is used. If the 'Transport Type' parameter is set to **UDP** or classification to this Proxy Set fails, this TLS Context is not used. Instead, the device uses the TLS Context configured for the SIP Interface (see "Configuring SIP Interfaces" on page 267) used for the call; otherwise, the default TLS Context (ID 0) is used. ▪ **Outgoing calls:** If the 'Transport Type' parameter (above) is set to **TLS** and the outgoing call is sent to an IP Group that is 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 more information about how certificates are associated with calls and for configuring TLS Contexts, see "Configuring SSL/TLS Certificates" on page 109.

21.7 Assign WAN Interface to VoIP Traffic

If you require remote management through the WAN interface, then follow the procedure described in this section.

Once you have configured the WAN IP address, you need to associate it with VoIP traffic (i.e., SIP signaling and media / RTP interfaces). The available WAN interfaces depend on the hardware configuration (e.g., Ethernet or SHDSL) and/or whether VLANs are configured for the WAN interface. If VLANs are configured, then you can select the WAN VLAN on which you want to run the SIP signaling and media interfaces.

Once this association is set, VoIP traffic is sent through the WAN and incoming traffic is identified as coming from the WAN. The device automatically configures the required port forwarding and static NAT rules.

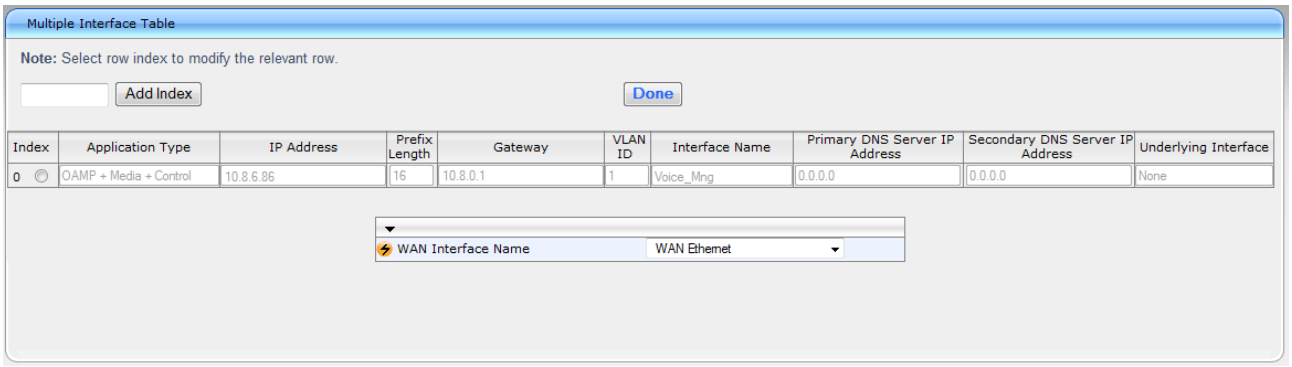
> **Note:** If you do not assign the WAN interface to SIP and media interfaces, then the WAN interface may not be used for VoIP traffic. In such cases, the VoIP traffic can be sent and received within the LAN or sent to the WAN through a third-party LAN router. If a third-party router is used as the interface to the WAN, then you need to configure NAT rules (in the NAT Translation table) to translate the VoIP LAN IP addresses (configured in the Interface table and associated with SIP and media interfaces) into global, public IP addresses.

➢ To assign a WAN interface to VoIP traffic:

1. Select the WAN interface:

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

Figure 21-7: Selecting WAN Interface for VoIP Traffic



Multiple Interface Table

Note: Select row index to modify the relevant row.

Index	Application Type	IP Address	Prefix Length	Gateway	VLAN ID	Interface Name	Primary DNS Server IP Address	Secondary DNS Server IP Address	Underlying Interface
0 ○	OAMP + Media + Control	10.8.6.86	16	10.8.0.1	1	Voice_Mng	0.0.0.0	0.0.0.0	None

WAN Interface Name: WAN Ethernet

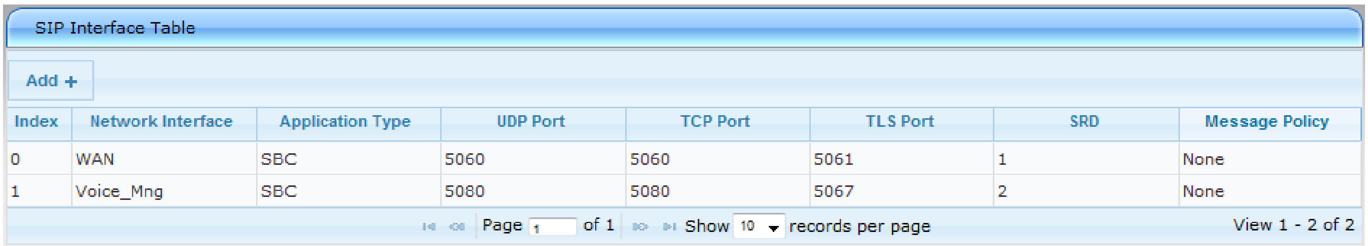
   b. From the 'WAN Interface Name' drop-down list, select the WAN interface for VoIP traffic.

   c. Click **Submit**, and then reset the device for your setting to take effect.

2. Assign the selected WAN interface to SIP signaling and RTP (media) interfaces. This is done in the SIP Interface and Media Realm tables respectively (whereby the WAN interface is denoted as "WAN"):

a. Open the SIP Interface Table page (see "Configuring SIP Interfaces" on page 267) and configure SIP interfaces on the WAN interface.

Figure 21-8: Assigning SIP Interface to WAN



SIP Interface Table

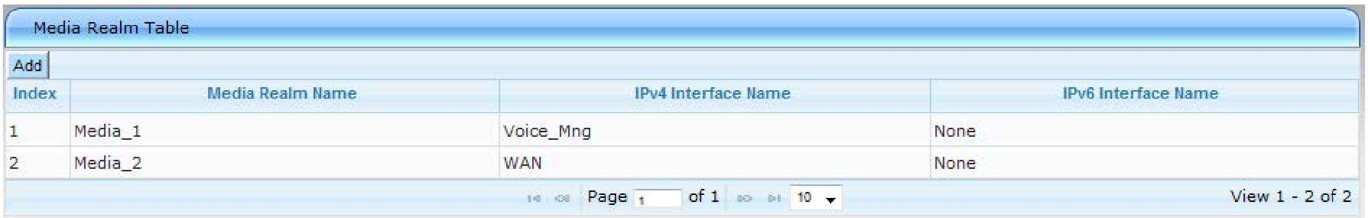
Add +

Index	Network Interface	Application Type	UDP Port	TCP Port	TLS Port	SRD	Message Policy
0	WAN	SBC	5060	5060	5061	1	None
1	Voice_Mng	SBC	5080	5080	5067	2	None

Page 1 of 1 Show 10 records per page View 1 - 2 of 2

b. Open the Media Realm Table page (see "Configuring Media Realms" on page 259) and configure media interfaces on the WAN interface.

Figure 21-9: Assigning WAN Interface to Media Realm



Media Realm Table

Add

Index	Media Realm Name	IPv4 Interface Name	IPv6 Interface Name
1	Media_1	Voice_Mng	None
2	Media_2	WAN	None

Page 1 of 1 10 View 1 - 2 of 2

c. Configure SRDs and associate them with these SIP signaling and media interfaces.

d. Configure other SIP settings as required.

22 SIP Definitions

This section describes configuration of various SIP-related functionalities.

22.1 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 763.

22.2 Configuring Registration Accounts

The Account table lets you configure up to 50 Accounts. An Account defines registration information for registering and authenticating (digest) "served" Trunk Groups or IP Groups (e.g., IP PBX) with a "serving" IP Group (e.g., ITSP). Registration information includes a username, password, host name (AOR), and contact user name (AOR). The device includes this information in the REGISTER message sent to the "serving" IP Group. Up to 10 Accounts can be configured per "served" Trunk Group or IP Group.

A "served" Trunk Group or IP Group can register to more than one "serving" IP Group (e.g., multiple ITSPs). This is done by configuring multiple entries in the Account table for the same "served" Trunk Group or IP Group, but with different "serving" IP Groups, user name/password, host name, and contact user values.

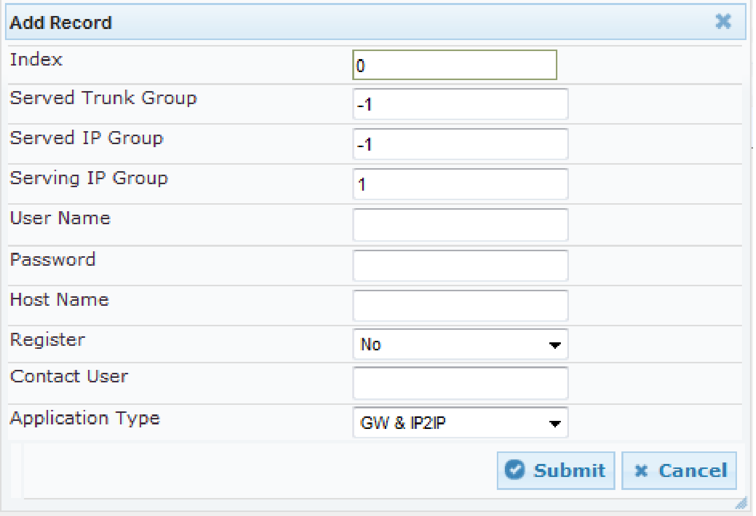
Note: If no match is found in the Account table for incoming or outgoing calls, the username and password is taken from:

- FXS interfaces: Authentication table (see Configuring Authentication on page 477 per Port)
- 'UserName' and 'Password' parameters on the Proxy & Registration page

The following procedure describes how to configure Accounts in the Web interface. You can also configure Accounts using the table ini file parameter, Account or CLI command, configure voip > sip-definition account.

➢ **To configure an Account:**

1. Open the Account Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Account Table**).
2. Click **Add**; the following dialog box appears:



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

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

Once you have configured Accounts, you can register or un-register them, as described below:

➢ To register or un-register an Account:

1. In the table, select the required Account entry row.
2. From the **Action** drop-down list, choose one of the following commands:
   - **Register** to register the Account.
   - **Un-Register** to un-register an Account.

To view Account registration status, see "Viewing Registration Status" on page 685.

If all trunks belonging to the Trunk Group are down, the device un-registers them. If any trunk belonging to the Trunk Group is returned to service, the device registers them again. This ensures, for example, that the Proxy does not send INVITEs to trunks that are out of service.

If registration with an IP Group fails for all accounts of a specific Trunk Group that includes all the channels in the Trunk Group, the Trunk Group is set to Out-Of-Service if the OOSOnRegistrationFail parameter is set to 1 (see Proxy & Registration Parameters on page 296).

Table 22-1: Account Table Parameter Descriptions

Parameter	Description
Index	Defines an index for the new table record.
Served Trunk Group CLI: served-trunk-group [Account_ServedTrunkGroup]	Defines the Trunk Group ID that you want to register and/or authenticate. ▪ For Tel-to-IP calls, the served Trunk Group is the source Trunk Group from where the call originated. ▪ For IP-to-Tel calls, the served Trunk Group is the Trunk Group ID to where the call is sent. **Note:** This parameter is applicable only to the Gateway application.
Served IP Group CLI: served-ip-group [Account_ServedIPGroup]	Defines the IP Group (e.g., IP-PBX) that you want to register and/or authenticate. **Note:** This parameter is applicable only to the SBC and Gateway's IP-to-IP application.
Serving IP Group	Defines the IP Group to where the device sends the SIP REGISTER

Parameter	Description
CLI: serving-ip-group **[Account_ServingIPGroup]**	requests (if enabled) for registration and authentication. ▪ For Tel-to-IP calls, the serving IP Group is the destination IP Group configured in the Trunk Group Settings table or Outbound IP Routing table (see Configuring the Outbound IP Routing Table on page 397). ▪ For IP-to-Tel calls, the serving IP Group is the 'Source IP Group ID' configured in the Inbound IP Routing table (see Configuring the Inbound IP Routing Table on page 406).
User Name CLI: user-name **[Account_Username]**	Defines the digest MD5 Authentication username. The valid value is a string of up to 50 characters.
Password CLI: password **[Account_Password]**	Defines the digest MD5 Authentication password. The valid value is a string of up to 50 characters.
Host Name CLI: host-name **[Account_HostName]**	Defines the Address of Record (AOR) host name. The host name appears in SIP REGISTER From/To headers as ContactUser@HostName. For a successful registration, the host name is also included in the URI of the INVITE From header. The valid value is a string of up to 49 characters. **Note:** If this parameter is not configured or if registration fails, the 'SIP Group Name' parameter value configured in the IP Group table is used instead.
Register CLI: register **[Account_Register]**	Enables registration. ▪ **[0]** No (Default) ▪ **[1]** Regular = Regular registration process. For more information, see "Regular Registration Mode" on page 294. ▪ **[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 294. **Notes:** ▪ Gateway application: To enable registration, you also need to set the 'Registration Mode' parameter to Per Account in the Hunt Group Settings table (see Configuring Hunt Group Settings on page 367). ▪ The account registration is not affected by the IsRegisterNeeded parameter.
Contact User CLI: contact-user **[Account_ContactUser]**	Defines the AOR username. This appears in REGISTER From/To headers as ContactUser@HostName, and in INVITE/200 OK Contact headers as ContactUser@<device's IP address>. **Notes:** ▪ If this parameter is not configured, the 'Contact User' parameter in the IP Group table is used instead. ▪ If registration fails, the user part in the INVITE Contact header contains the source party number.
Application Type CLI: application-type **[Account_ApplicationType]**	Defines the application type: ▪ [0] GW/IP2IP = (Default) Gateway and IP-to-IP application. ▪ [2] SBC = SBC application.

22.2.1 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
```

22.2.2 Single Registration for Multiple Phone Numbers using GIN

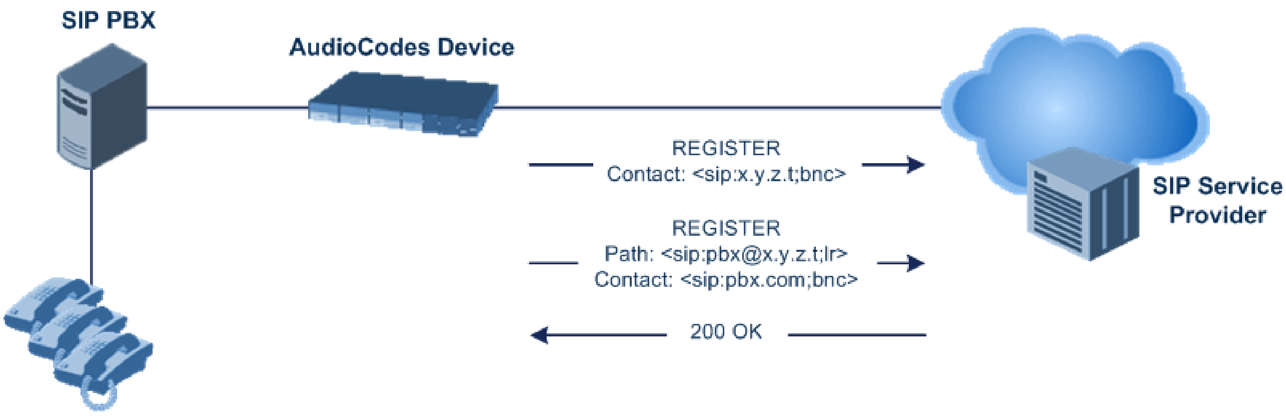
When you configure the registration mode in the Account table to **GIN**, the Global Identifiable Number (GIN) registration method is used, according to RFC 6140. The device performs GIN-based registration of users to a SIP registrar on behalf of a SIP PBX. In effect, the PBX registers with the service provider, just as a directly hosted SIP endpoint would register. However, because a PBX has multiple user agents, it needs to register a contact address on behalf of each of these. Rather than performing a separate registration procedure for each user agents, GIN registration mode does multiple registrations using a single REGISTER transaction.

According to this mechanism, the PBX delivers to the service provider in the Contact header field of a REGISTER request a template from which the service provider can construct contact URIs for each of the AORs assigned to the PBX and thus, can register these contact URIs within its location service. These registered contact URIs can then be used to deliver to the PBX inbound requests targeted at the AORs concerned. The mechanism can be used with AORs comprising SIP URIs based on global E.164 numbers and the service provider's domain name or sub-domain name.

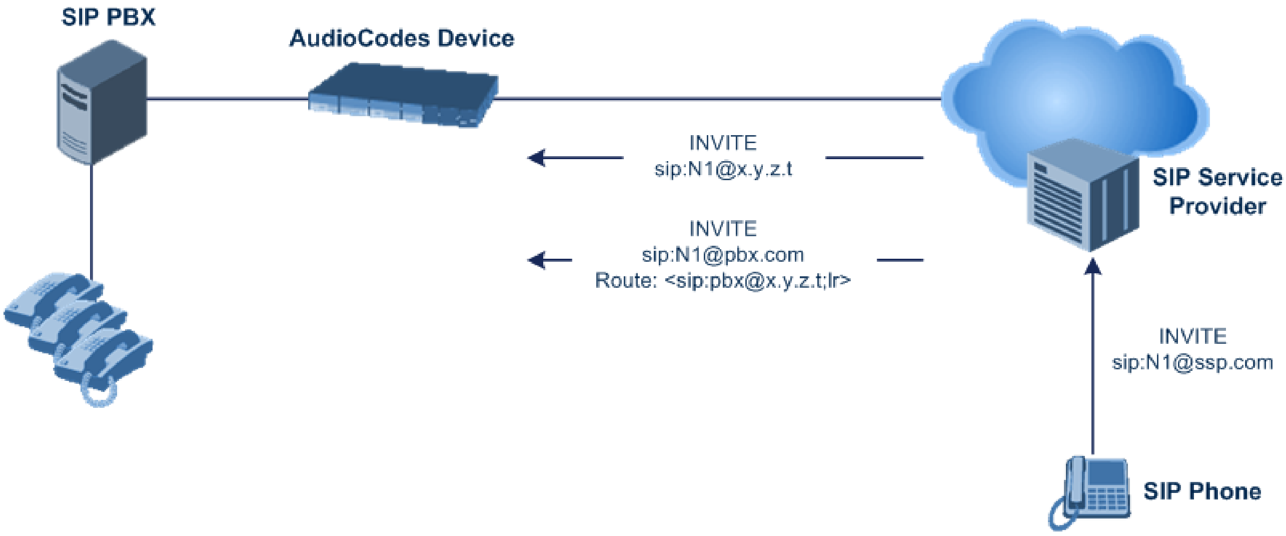
The SIP REGISTER request sent by the device for GIN registration with a SIP server provider contains the Require and Proxy-Require headers. These headers contain the token 'gin'. The Supported header contains the token 'path' and the URI in the Contact header contains the parameter 'bnc' without a user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

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

The figure below illustrates the GIN registration process:


SIP PBX
AudioCodes Device
REGISTER
Contact: <sip:x.y.z.t;bnc>
REGISTER
Path: <sip:pbx@x.y.z.t;lr>
Contact: <sip:pbx.com;bnc>
200 OK
SIP Service Provider

The figure below illustrates an incoming call using GIN:



22.3 Configuring Proxy and Registration Parameters

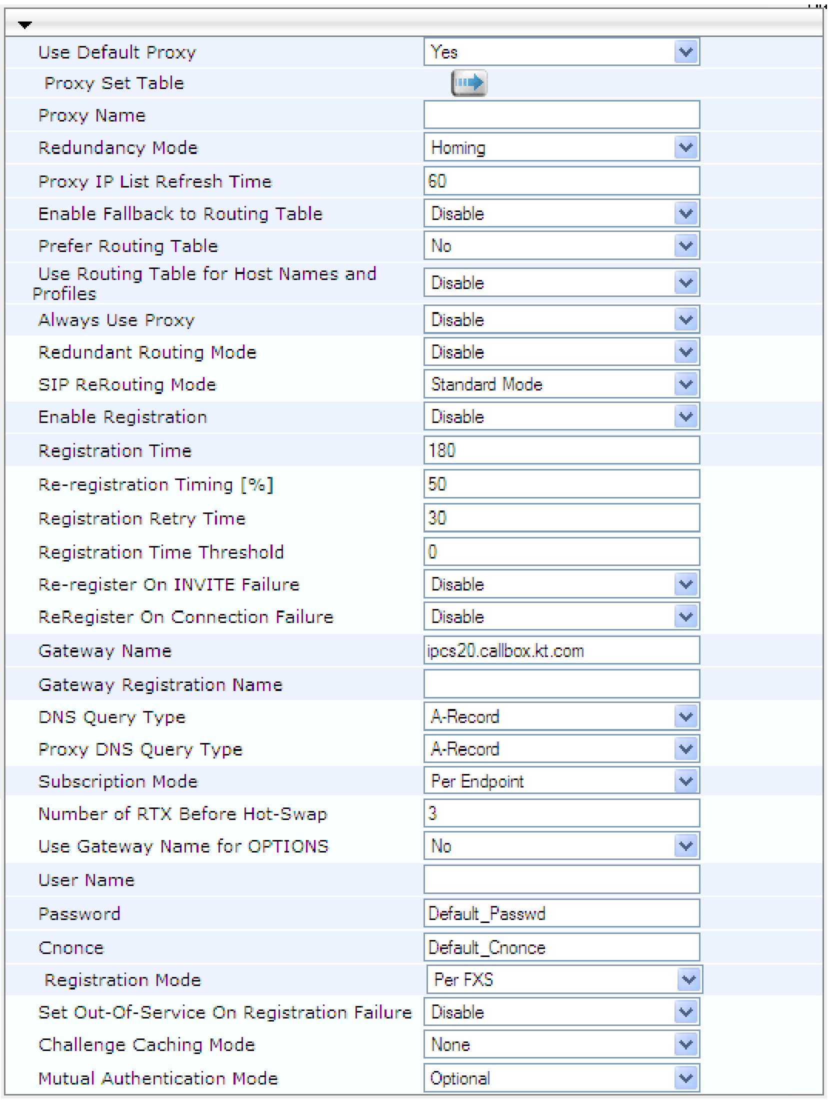
The Proxy & Registration page allows you to configure the Proxy server and registration parameters. For a description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 763.

Note: To view the registration status of endpoints with a SIP Registrar/Proxy server, see "Viewing Registration Status" on page 685.

➢ To configure the Proxy and registration parameters:

1. Open the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Proxy & Registration**).

Figure 22-1: Proxy & Registration Page



Use Default Proxy	Yes
Proxy Set Table	➡
Proxy Name	
Redundancy Mode	Homing
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	Disable
SIP ReRouting Mode	Standard Mode
Enable Registration	Disable
Registration Time	180
Re-registration Timing [%]	50
Registration Retry Time	30
Registration Time Threshold	0
Re-register On INVITE Failure	Disable
ReRegister On Connection Failure	Disable
Gateway Name	ipcs20.callbox.kt.com
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	
Password	Default_Passwd
Cnonce	Default_Cnonce
Registration Mode	Per FXS
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:

-
- FXS/FXO endpoints, BRI endpoints, Trunk Groups - Trunk Group table (see Configuring Trunk Group Table on page 365)
- Accounts - Account table (see "Configuring Registration Accounts" on page 291)

Click the **Proxy Set Table** ➡ button to Open the Proxy Sets Table page to configure groups of proxy addresses. Alternatively, you can open this page from the **Proxy Sets Table** page item (see "Configuring Proxy Sets" on page 282 for a description of this page).

22.3.1 SIP Message Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261 standard. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then re-send the INVITE with a Proxy-Authorization header containing the credentials.

User agents, Redirect or Registrar servers typically use the SIP 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example shows the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to a Registrar/Proxy server for registration:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip: 122@10.1.1.200>;tag=1c17940
To: <sip: 122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Sip-Gateway/Mediant 800B MSBR/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 "a9a031cfddcb10d91c8e7b4926086f7e".
6. Final stage:
    - A1 result: The nonce from the proxy response is "11432d6bce58ddf02e3b5e1c77c010d2".
    - A2 result: The equation to be evaluated is "A1:11432d6bce58ddf02e3b5e1c77c010d2:A2".
    - The MD5 algorithm is run on this equation. The outcome of the calculation is the response needed by the device to register with the Proxy.
    - The response is "b9c45d0234a5abf5ddf5c704029b38cf".

    At this time, a new REGISTER request is issued with the following response:

    ```
    REGISTER sip:10.2.2.222 SIP/2.0
    Via: SIP/2.0/UDP 10.1.1.200
    From: <sip: 122@10.1.1.200>;tag=1c23940
    To: <sip: 122@10.1.1.200>
    Call-ID: 654982194@10.1.1.200
    Server: Audiocodes-Sip-Gateway/Mediant 800B
    MSBR/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
```

22.4 Configuring SIP Message Manipulation

The Message Manipulations table lets you configure up to 100 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:

- SBC application: 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 267).
  - Post-classification Process: Message manipulation can be done on inbound and/or outbound SIP messages after the call has been successfully classified. You configure this by assigning the Manipulation Set ID to the relevant IP Group in the IP Group table (see Configuring IP Groups on page 270).
- Gateway application: Message Manipulation rules are applied to calls as follows:
  - Manipulating Inbound SIP INVITE Messages: Message manipulation can be applied only to all inbound calls (not specific calls). This is done by assigning a Manipulation Set ID to the "global" ini file parameter, GWInboundManipulationSet.
  - Manipulating Outbound SIP INVITE Messages:
    a. Message manipulation can be done for specific calls, by assigning a Manipulation Set ID to an IP Group in the IP Group table, using the 'Outbound Message Manipulation Set' parameter.
    b. Message manipulation can be applied to all outbound calls (except for IP Groups that have been assigned a Manipulation Set ID). This is done by assigning a Manipulation Set ID to the "global" ini file parameter, GWOutboundManipulationSet.

The device also supports a built-in SIP message normalization feature that can be enabled per Message Manipulation rule. The normalization feature removes unknown SIP message elements before forwarding the message. These elements can include SIP headers, SIP header parameters, and SDP body fields.

The SIP message manipulation feature supports the following:

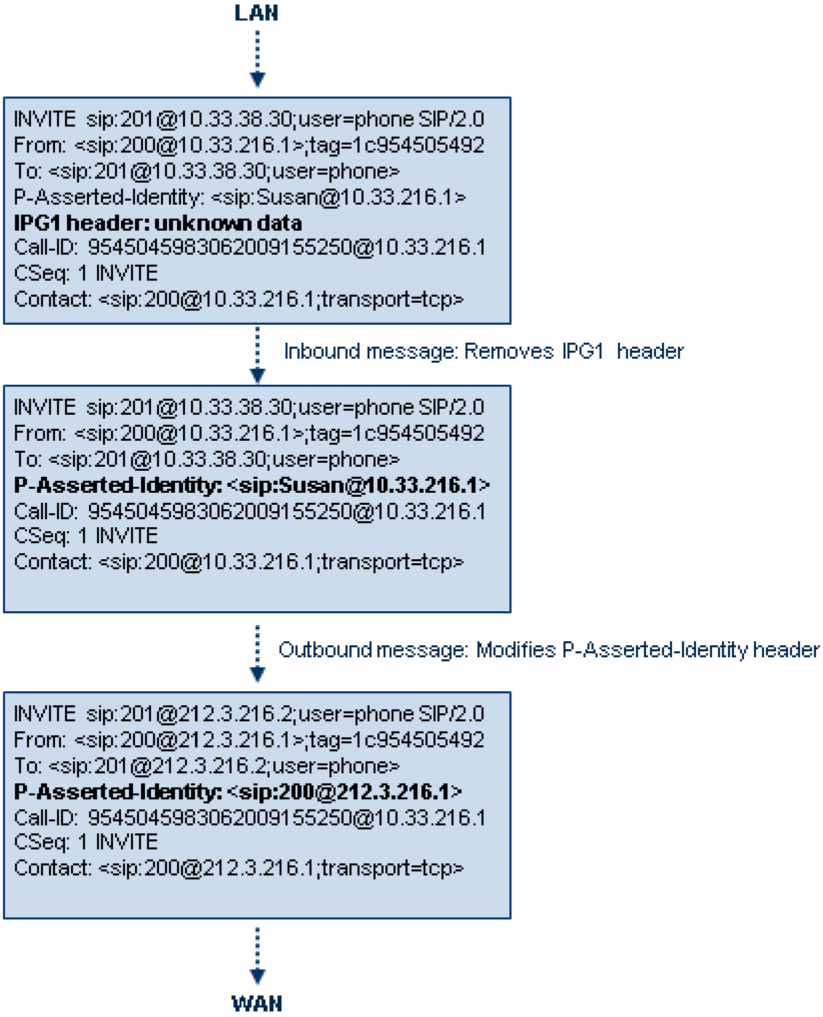
- Manipulation on SIP message type (Method, Request/Response, and Response type)
- Addition of new SIP headers
- Removal of SIP headers ("black list")
- Modification of SIP header components such as values, header values (e.g., URI

value of the P-Asserted-Identity header can be copied to the From header), call's parameter values

- Deletion of SIP body (e.g., if a message body is not supported at the destination network this body is removed)
- Translating one SIP response code to another
- Topology hiding (generally present in SIP headers such as Via, Record Route, Route and Service-Route).
- Configurable identity hiding (information related to identity of subscribers, for example, P-Asserted-Identity, Referred-By, Identity and Identity-Info)
- Apply conditions per rule - the condition can be on parts of the message or call's parameters
- Multiple manipulation rules on the same SIP message

The figure below illustrates a SIP message manipulation example:

Figure 22-2: SIP Header Manipulation Example


LAN
Inbound message: Removes IPG1 header
Outbound message: Modifies P-Asserted-Identity header
WAN

```
INVITE sip:201@10.33.38.30;user=phone SIP/2.0
From: <sip:200@10.33.216.1>;tag=1c954505492
To: <sip:201@10.33.38.30;user=phone>
P-Asserted-Identity: <sip:Susan@10.33.216.1>
IPG1 header: unknown data
Call-ID: 9545045983062009155250@10.33.216.1
CSeq: 1 INVITE
Contact: <sip:200@10.33.216.1;transport=tcp>
```

```
INVITE sip:201@10.33.38.30;user=phone SIP/2.0
From: <sip:200@10.33.216.1>;tag=1c954505492
To: <sip:201@10.33.38.30;user=phone>
P-Asserted-Identity: <sip:Susan@10.33.216.1>
Call-ID: 9545045983062009155250@10.33.216.1
CSeq: 1 INVITE
Contact: <sip:200@10.33.216.1;transport=tcp>
```

```
INVITE sip:201@212.3.216.2;user=phone SIP/2.0
From: <sip:200@212.3.216.1>;tag=1c954505492
To: <sip:201@212.3.216.2;user=phone>
P-Asserted-Identity: <sip:200@212.3.216.1>
Call-ID: 9545045983062009155250@10.33.216.1
CSeq: 1 INVITE
Contact: <sip:200@212.3.216.1;transport=tcp>
```

Notes:

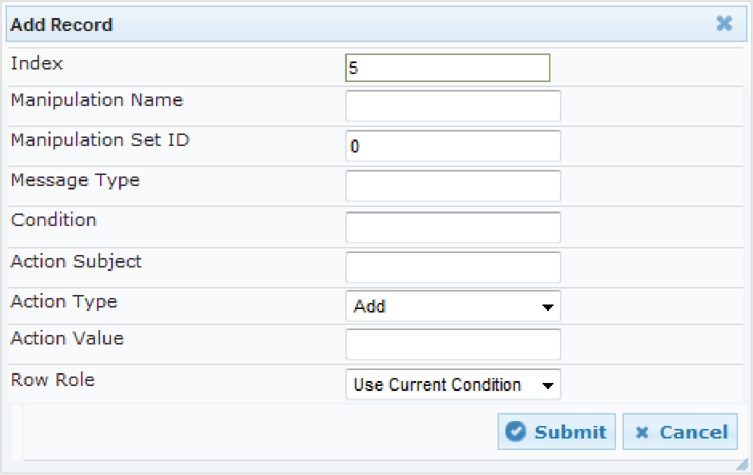
- For a detailed description of the syntax used for configuring Message Manipulation rules, refer to the *SIP Message Manipulations Quick Reference Guide*.
- For the SBC application, Inbound message manipulation is done only after the Classification, inbound/outbound number manipulations, and routing 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 following procedure describes how to configure Message Manipulation rules in the Web interface. You can also configure Message Manipulation rules using the table ini file parameter, MessageManipulations or CLI command, configure voip > sbc manipulations message-manipulations.

➢ To configure SIP message manipulation rules:

1. Open the Message Manipulations page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Msg Policy & Manipulation** > **Message Manipulations**).
2. Click **Add**; the following dialog box appears:

Figure 22-3: Message Manipulations Table - Add Record Dialog Box



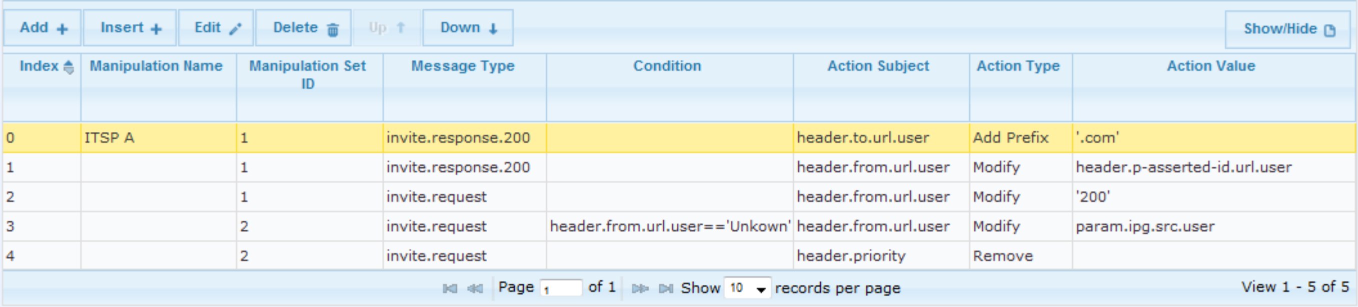
Add Record

Index	5
Manipulation Name	
Manipulation Set ID	0
Message Type	
Condition	
Action Subject	
Action Type	Add
Action Value	
Row Role	Use Current Condition

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

An example of configured message manipulation rules are shown in the figure below:

Figure 22-4: Message Manipulations Page



Index	Manipulation Name	Manipulation Set ID	Message Type	Condition	Action Subject	Action Type	Action Value
0	ITSP A	1	invite.response.200		header.to.url.user	Add Prefix	'.com'
1		1	invite.response.200		header.from.url.user	Modify	header.p-asserted-id.url.user
2		1	invite.request		header.from.url.user	Modify	'200'
3		2	invite.request	header.from.url.user=='Unkown'	header.from.url.user	Modify	param.ipg.src.user
4		2	invite.request		header.priority	Remove	

- 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 22-2: Message Manipulations Parameter Descriptions

Parameter	Description
Index **[MessageManipulations_Index]**	Defines an index number for the new table record. **Note:** Each rule must be configured with a unique index.
Manipulation Name CLI: manipulation-name **[MessageManipulations_ManipulationName]**	Defines an arbitrary name to easily identify the Message Manipulation rule. The valid value is a string of up to 16 characters.
Manipulation Set ID CLI: manipulation-set-id **[MessageManipulations_ManSetID]**	Defines a Manipulation Set ID for the rule. You can define the same Manipulation Set ID for multiple rules to create a group of rules. The Manipulation Set ID is used to assign the manipulation rules to an IP Group (in the IP Group table) for inbound and/or outbound messages. The valid value is 0 to 19. The default is 0.
Matching Characteristics	
Message Type CLI: message-type **[MessageManipulations_MessageType]**	Defines the SIP message type that you want to manipulate. The valid value is a string (case-insensitive) denoting the SIP message. For example: ▪ Empty = rule applies to all messages ▪ Invite = rule applies to all INVITE requests and responses ▪ Invite.Request = rule applies to INVITE requests ▪ Invite.Response = rule applies to INVITE responses ▪ subscribe.response.2xx = rule applies to SUBSCRIBE confirmation responses **Note:** Currently, SIP 100 Trying messages cannot

Parameter	Description
	be manipulated.
Condition CLI: condition **[MessageManipulations_Condition]**	Defines the condition that must exist for the rule to apply. The valid value is a string (case-insensitive). For example: ▪ header.from.url.user== '100' (indicates that the user part of the From header must have the value "100") ▪ header.contact.param.expires > '3600' ▪ header.to.url.host contains 'domain' ▪ param.call.dst.user != '100'
Operation	
Action Subject CLI: action-subject **[MessageManipulations_ActionSubject]**	Defines the SIP header upon which the manipulation is performed. The valid value is a string (case-insensitive).
Action Type CLI: action-type **[MessageManipulations_ActionType]**	Defines the type of manipulation. ▪ **[0]** Add (default) = Adds new header/param/body (header or parameter elements). ▪ **[1]** Remove = Removes header/param/body (header or parameter elements). ▪ **[2]** Modify = Sets element to the new value (all element types). ▪ **[3]** Add Prefix = Adds value at the beginning of the string (string element only). ▪ **[4]** Add Suffix = Adds value at the end of the string (string element only). ▪ **[5]** Remove Suffix = Removes value from the end of the string (string element only). ▪ **[6]** Remove Prefix = Removes value from the beginning of the string (string element only). ▪ **[7]** Normalize = Removes unknown SIP message elements before forwarding the message.
Action Value CLI: action-value **[MessageManipulations_ActionValue]**	Defines a value that you want to use in the manipulation. The default value is a string (case-insensitive) in the following syntax: ▪ string/<message-element>/<call-param> + ▪ string/<message-element>/<call-param> For example: ▪ 'itsp.com' ▪ header.from.url.user ▪ param.call.dst.user ▪ param.call.dst.host + '.com' ▪ param.call.src.user + '<' + header.from.url.user + '@' + header.p-asserted-id.url.host + '>'

Parameter	Description
	Note: Only single quotation marks must be used.
Row Role CLI: row-role **[MessageManipulations_RowRole]**	Determines which condition must be used for the rule of this table row. ▪ **[0]** Use Current Condition = The condition entered in this row must be matched in order to perform the defined action (default). ▪ **[1]** Use Previous Condition = The condition of the rule configured directly above this row must be used in order to perform the defined action. This option allows you to configure multiple actions for the same condition. **Note:** When multiple manipulations rules apply to the same header, the next rule applies to the result string of the previous rule.

22.5 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 267).

Each Message Policy rule can be configured with the following:

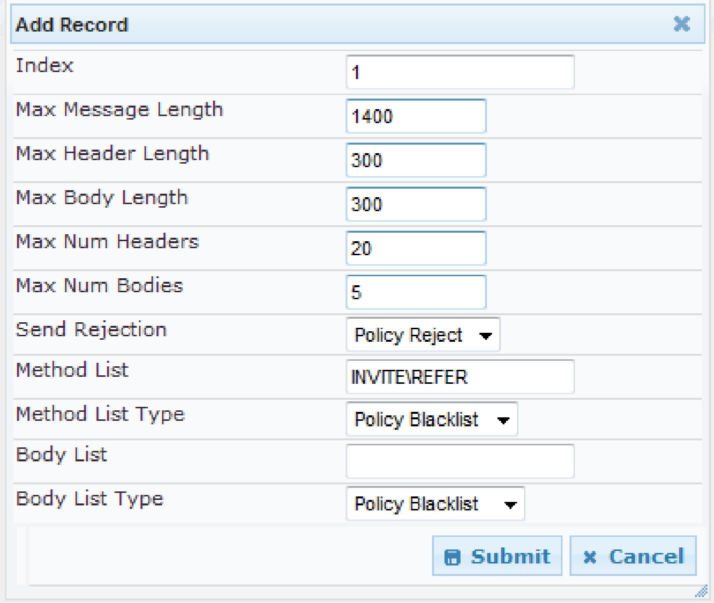
- Maximum message length
- Maximum header length
- Maximum message body length
- Maximum number of headers
- Maximum number of bodies
- Option to send 400 "Bad Request" response if message request is rejected
- Blacklist and whitelist for defined methods (e.g., INVITE)
- Blacklist and whitelist for defined bodies

The following procedure describes how to configure Message Policy rules in the Web interface. You can also configure Message Policy rules using the table ini file parameter, MessagePolicy or the CLI command, configure voip > sbc message-policy.

➢ To configure SIP Message Policy rules:

1. Open the Message Policy Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Msg Policy & Manipulation** > **Message Policy Table**).
2. Click **Add**; the following dialog box appears:

Figure 22-5: Message Policy Table - Add Record Dialog Box



Add Record

Index	1
Max Message Length	1400
Max Header Length	300
Max Body Length	300
Max Num Headers	20
Max Num Bodies	5
Send Rejection	Policy Reject
Method List	INVITE\REFER
Method List Type	Policy Blacklist
Body List	
Body List Type	Policy Blacklist

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

Table 22-3: Message Policy Table Parameter Descriptions

Parameter	Description
Index **[MessagePolicy_Index]**	Defines an index number for the new table record.
Max Message Length CLI: max-message-length **[MessagePolicy_MaxMessageLength]**	Defines the maximum SIP message length. The valid value is up to 32,768 characters. The default is 32,768.
Max Header Length CLI: max-header-length **[MessagePolicy_MaxHeaderLength]**	Defines the maximum SIP header length. The valid value is up to 512 characters. The default is 512.
Max Body Length CLI: max-body-length **[MessagePolicy_MaxBodyLength]**	Defines the maximum SIP message body length. This is the value of the Content-Length header. The valid value is up to 1,024 characters. The default is 1,024.
Max Num Headers CLI: max-num-headers **[MessagePolicy_MaxNumHeaders]**	Defines the maximum number of SIP headers. The valid value is any number up to 32. The default is 32. **Note:** The device supports up to 20 SIP Record-Route headers that can be received in a SIP INVITE request or a 200 OK response. If it receives more than this, it responds with a SIP 513 'Message Too Large' response.

Parameter	Description
Max Num Bodies CLI: max-num-bodies **[MessagePolicy_MaxNumBodies]**	Defines the maximum number of bodies (e.g., SDP) in the SIP message. The valid value is any number up to 8. The default is 8.
Send Rejection CLI: send-rejection **[MessagePolicy_SendRejection]**	Determines whether the device sends a 400 "Bad Request" response if a message request is rejected. ▪ **[0]** Policy Reject = (Default) If the message is a request, the device sends a response to reject the request. ▪ **[1]** Policy Drop = The device ignores the message without sending any response.
SIP Method Blacklist-Whitelist Policy	
Method List CLI: method-list **[MessagePolicy_MethodList]**	Defines SIP methods (e.g., INVITE\BYE) to blacklist or whitelist. Multiple methods are separated by a backslash (\). The method values are case-insensitive.
Method List Type CLI: method-list-type **[MessagePolicy_MethodListType]**	Defines the policy (blacklist or whitelist) for the SIP methods specified in the 'Method List' parameter (above). ▪ **[0]** Policy Blacklist = The specified methods are rejected. ▪ **[1]** Policy Whitelist = (Default) Only the specified methods are allowed; the others are rejected.
SIP Body Blacklist-Whitelist Policy	
Body List CLI: body-list **[MessagePolicy_BodyList]**	Defines the SIP body type (i.e., value of the Content-Type header) to blacklist or whitelist. For example, application/sdp. The values of this parameter are case-sensitive.
Body List Type CLI: body-list-type **[MessagePolicy_BodyListType]**	Defines the policy (blacklist or whitelist) for the SIP body specified in the 'Body List' parameter (above). ▪ **[0]** Policy Blacklist =The specified SIP body is rejected. ▪ **[1]** Policy Whitelist = (Default) Only the specified SIP body is allowed; the others are rejected.

23 Coders and Profiles

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

23.1 Configuring Default Coders

The Coders table lets you configure up to 10 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 310) is assigned to specific calls, using Tel or 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:

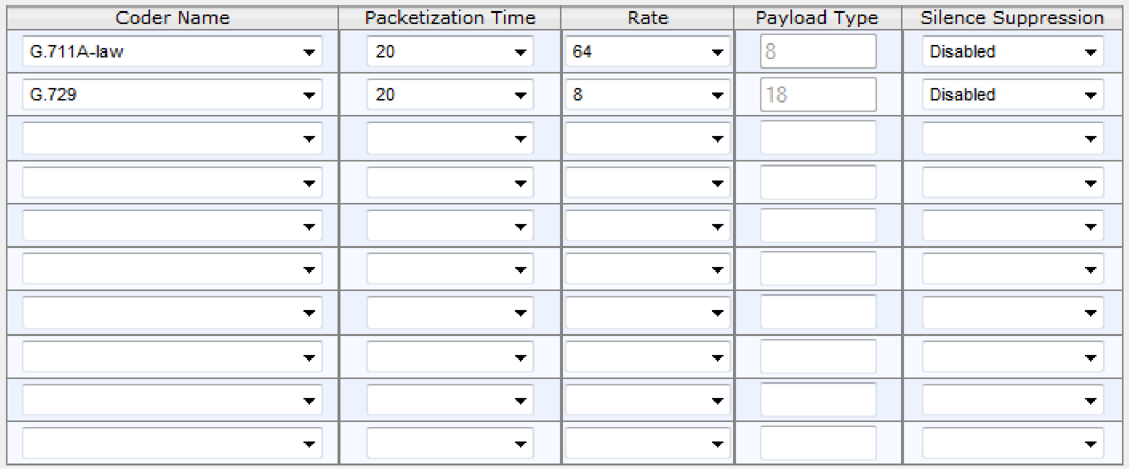
- Only the packetization time of the first coder in the coder list is declared in INVITE/200 OK SDP, even if multiple coders are defined. The device always uses the packetization time requested by the remote side for sending RTP packets. If not specified, the packetization time is assigned the default 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.
- For information on V.152 and implementation of T.38 and VBD coders, see "Supporting V.152 Implementation" on page 189.

The following procedure describes how to configure the Coders table in the Web interface. You can also configure this table using the table ini file parameter, CodersGroup0 or CLI command, configure voip > coders-and-profiles coders-group.

➢ **To configure coders:**

1. Open the Coders page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Coders**).

Figure 23-1: Coders Table Page



Coder Name	Packetization Time	Rate	Payload Type	Silence Suppression
G.711A-law	20	64	8	Disabled
G.729	20	8	18	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 23-1: Coders Table Parameter Descriptions

Parameter	Description
Coder Name CLI: name **[CodersGroup0_Name]**	Defines the coder. **Note:** Each coder type (e.g., G.729) can be configured only once in the table.
Packetization Time CLI: 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 CLI: rate **[CodersGroup0_rate]**	Defines the bit rate (in kbps) for the coder.
Payload Type CLI: 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 CLI: 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 silence suppression is not configured for a coder, the settings of the EnableSilenceCompression parameter is used. ▪ 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. An exception to this logic is when the remote gateway is Cisco equipment (IsCiscoSCEMode). ▪

The table below lists the supported coders:

Table 23-2: 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	Dynamic (0-127); Default 180	N/A
G.711U-law_VBD **[g711UlawVbd]**	10, 20 (default), 30, 40, 50, 60, 80, 100, 120	64	Dynamic (0-127); Default 120	N/A

Coder Name	Packetization Time (msec)	Rate (kbps)	Payload Type	Silence Suppression
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 (0-127); Default 23	▪ **[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
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 (0-127)	▪ [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 (0-127)	▪ [0] Disable ▪ [1] Enable
iLBC [iLBC]	20 (default), 40, 60, 80, 100, 120	15 (default)	Dynamic (0-127); 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]	-	-	-	-

Coder Name	Packetization Time (msec)	Rate (kbps)	Payload Type	Silence Suppression
T.38 Over RTP	N/A	N/A	Dynamic (90 - 127); Default 106	N/A

23.2 Configuring Coder Groups

The Coder Group Settings table lets you configure up to 10 *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 Tel Profiles (see Configuring Tel Profiles on page 311) or IP Profiles (see "Configuring IP Profiles" on page 314). For the SBC application, Coder Groups can be used as Extension Coders and/or Allowed Coders.

Notes:

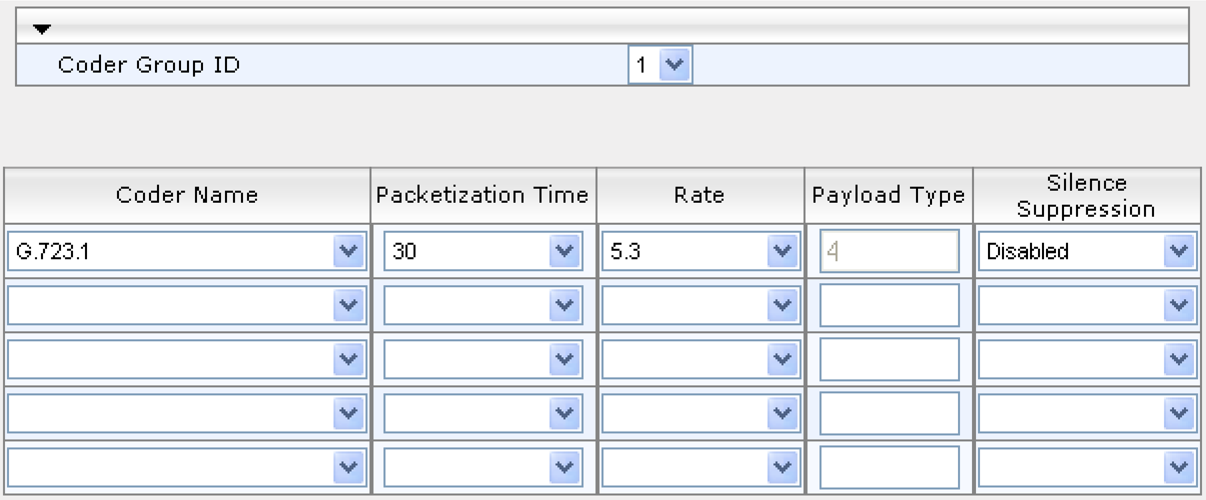
- To define coders for calls that are not assigned a specific Coder Group using Tel Profiles or IP Profiles, see "Configuring Default Coders" on page 307. This group of coders is termed the *Default Coder Group*.
- For a list of supported coders, see "Configuring Default Coders" on page 307.

The following procedure describes how to configure the Coders table in the Web interface. You can also configure this table using the table ini file parameter, CodersGroupX or CLI command, 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 23-2: Coder Group Settings Page



Coder Group ID: 1

Coder Name	Packetization Time	Rate	Payload Type	Silence Suppression
G.723.1	30	5.3	4	Disabled

2. Configure the Coder Group according to the parameters described in the table below.

3. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

Table 23-3: Coder Group Settings Table Parameter Descriptions

Parameter	Description
Coder Group ID **[CodersGroupX_Index]**	Defines an ID for the Coder Group.
Coder Name CLI: 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 CLI: 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 CLI: rate **[CodersGroupX_rate]**	Defines the bit rate (in kbps) for the coder.
Payload Type CLI: 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 CLI: 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. **Notes:** ▪ If silence suppression is not configured for a coder, the settings of the EnableSilenceCompression parameter is used.

23.3 Configuring Tel Profile

The Tel Profile Settings table lets you configure up to nine *Tel Profiles*. A Tel Profile is a set of parameters with specific settings which can be assigned to specific calls. The Tel Profile Settings table includes a wide range of parameters for configuring the Tel Profile. Each of these parameters has a corresponding "global" parameter, which when configured applies to all calls. The main difference, if any, between the Tel Profile parameters and their corresponding global parameters are their default values.

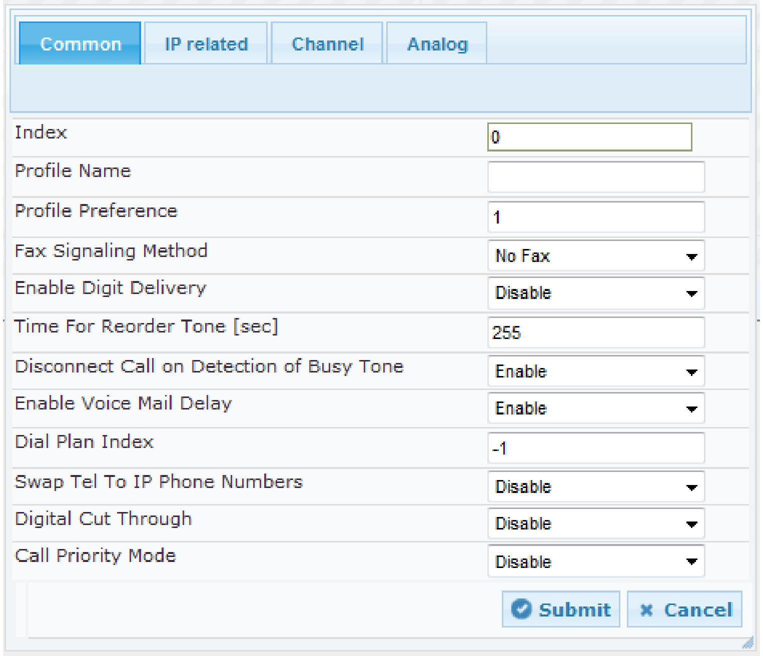
Tel Profiles provide high-level adaptation when the device interworks between different equipment and protocols (at both the Tel and IP sides), each of which may require different handling by the device. For example, if specific channels require the use of the G.711 coder, you can configure a Tel Profile with this coder and assign it to these channels.

To use your Tel Profile for specific calls, you need to assign it to specific channels (trunks) in the Trunk Group table (see Configuring Trunk Group Table on page 365)).

The following procedure describes how to configure Tel Profiles in the Web interface. You can also configure Tel Profiles using the table ini file parameter, TelProfile or CLI command, configure voip/coders-and-profiles tel-profile.

➢ **To configure a Tel Profile:**

1. Open the Tel Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **Tel Profile Settings**).
2. Click **Add**; the following dialog box appears:



Index	0
Profile Name	
Profile Preference	1
Fax Signaling Method	No Fax
Enable Digit Delivery	Disable
Time For Reorder Tone [sec]	255
Disconnect Call on Detection of Busy Tone	Enable
Enable Voice Mail Delay	Enable
Dial Plan Index	-1
Swap Tel To IP Phone Numbers	Disable
Digital Cut Through	Disable
Call Priority Mode	Disable

3. Configure a Tel Profile according to the parameters described in the table below. For a description of each parameter, refer to the corresponding "global" parameter.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 23-4: Tel Profile Table Parameters and Corresponding Global Parameters

Tel Profile Parameter	Global Parameter
Common	
Web: Index **[TelProfile_Index]**	Defines an index number for the new table record.
Web: Profile Name CLI: profile-name **[TelProfile_ProfileName]**	Defines an arbitrary name to easily identify the Tel Profile. The valid value is a string of up to 20 characters.
Web: Profile Preference CLI: tel-preference **[TelProfile_TelPreference]**	Defines the priority of the Tel Profile, where **1** is the lowest priority and **20** the highest priority. **Notes:** 1. If both the IP Profile and Tel Profile apply to the same call, the coders and common parameters of the Preferred profile are applied to the call. 2. If the Preference of the Tel Profile and IP Profile are identical, the Tel Profile parameters are applied. 3. If the coder lists of both the IP Profile and Tel Profile apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference.
Web: Fax Signaling Method CLI: fax-sig-method [TelProfile_IsFaxUsed]	IsFaxUsed
Web: Enable Digit Delivery CLI: digit-delivery [TelProfile_EnableDigitDelivery]	EnableDigitDelivery
Web: Time For Reorder Tone CLI: time-for-reorder-tone [TelProfile_TimeForReorderTone]	TimeForReorderTone
Web: Disconnect Call on Detection of Busy Tone CLI: disconnect-on-busy-tone [TelProfile_DisconnectOnBusyTone]	DisconnectOnBusyTone

Tel Profile Parameter	Global Parameter
Web: Enable Voice Mail Delay CLI: enable-voice-mail-delay [TelProfile_EnableVoiceMailDelay]	VoiceMailInterface This is useful for disabling voice mail services per Trunk Group to eliminate the phenomenon of call delay on Trunks that do not implement voice mail when voice mail is enabled using the global parameter.
Web: Dial Plan Index CLI: dial-plan-index [TelProfile_DialPlanIndex]	DialPlanIndex
Web: Swap Tel To IP Phone Numbers CLI: swap-teltoip-phone-numbers [TelProfile_SwapTelToIpPhoneNumbers]	SwapTEl2IPCalled&CallingNumbers
Web: Digital Cut Through CLI: digital-cut-through [TelProfile_DigitalCutThrough]	DigitalCutThrough
Web: Call Priority Mode CLI: call-priority-mode [TelProfile_CallPriorityMode]	CallPriorityMode
IP Related	
Web: Coders Group ID CLI: coders-group-id [TelProfile_CodersGroupID]	CodersGroup0
Web: RTP IP DiffServ CLI: rtp-ip-diffserv [TelProfile_IPDiffServ]	PremiumServiceClassMediaDiffServ
Web: Signaling DiffServ CLI: signaling-diffserv [TelProfile_SigIPDiffServ]	PremiumServiceClassControlDiffServ
Web: Enable Early Media CLI: early-media [TelProfile_EnableEarlyMedia]	EnableEarlyMedia
Web: Progress Indicator to IP CLI: prog-ind-to-ip [TelProfile_ProgressIndicator2IP]	ProgressIndicator2IP
Channel	
Web: Dynamic Jitter Buffer Minimum Delay CLI: jitter-buffer-minimum-delay [TelProfile_JitterBufMinDelay]	DJBufMinDelay
Web: Dynamic Jitter Buffer Optimization Factor CLI: jitter-buffer-optimization-factor [TelProfile_JitterBufOptFactor]	DJBufOptFactor
Web: DTMF Volume CLI: dtmf-volume [TelProfile_DtmfVolume]	DTMFVolume
Web: Input Gain CLI: input-gain [TelProfile_InputGain]	InputGain
Web: Voice Volume CLI: voice-volume [TelProfile_VoiceVolume]	VoiceVolume
Web: Echo Canceler CLI: echo-canceller [TelProfile_EnableEC]	EnableEchoCanceller
Web: Enable AGC CLI: enable-agc [TelProfile_EnableAGC]	EnableAGC

Tel Profile Parameter	Global Parameter
Web: EC NLP Mode CLI: echo-canceller-nlp-mode [TelProfile_ECNlpMode]	ECNLPMode
Analog	
Web: Enable Polarity Reversal CLI: polarity-rvrsl [TelProfile_EnableReversePolarity]	EnableReversalPolarity
Web: Enable Current Disconnect CLI: current-disconnect [TelProfile_EnableCurrentDisconnect]	EnableCurrentDisconnect
Web: MWI Analog Lamp CLI: mwi-analog-lamp [TelProfile_MWIAnalog]	MWIAnalogLamp
Web: MWI Display CLI: mwi-display [TelProfile_MWIDisplay]	MWIDisplay
Web: Flash Hook Period CLI: flash-hook-period [TelProfile_FlashHookPeriod]	FlashHookPeriod
Web: DID Wink CLI: enable-did-wink [TelProfile_EnableDIDWink]	EnableDIDWink
Web: Two Stage Dialing CLI: is-two-stage-dial [TelProfile_IsTwoStageDial]	IsTwoStageDial
Web: Enable 911 PSAP CLI: enable-911-psap [TelProfile_Enable911PSAP]	Enable911PSAP
Web: FXO Double Answer CLI: fxo-double-answer [TelProfile_EnableFXODoubleAnswer]	EnableFXODoubleAnswer
Web: FXO Ring Timeout CLI: fxo-ring-timeout [TelProfile_FXORingTimeout]	FXORingTimeout Note: If this parameter is configured for a specific FXO port, Caller ID detection does not occur, and the RingBeforeCallerID and FXONumberOfRings parameters do not affect the outgoing INVITE for that FXO port.

23.4 Configuring IP Profiles

The IP Profile Settings table lets you configure up to 20 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 270
- Gateway application: Outbound IP Routing rules - see Configuring Outbound IP Routing Table on page 397
- Gateway application: Inbound IP Routing rules - see Configuring Inbound IP Routing Table on page 406

For the Gateway application: The device selects the IP Profile as follows:

- If you assign different IP Profiles (not default) to the same specific calls in all of the above-mentioned tables, the device uses the IP Profile that has the highest preference level (as set in the 'Profile Preference' parameter). If these IP Profiles have the same preference level, the device uses the IP Profile that you assigned in the IP Group table.
- If you assign different IP Profiles to all of the above-mentioned tables and one table is set to the default IP Profile, the device uses the IP Profile that is not the default.

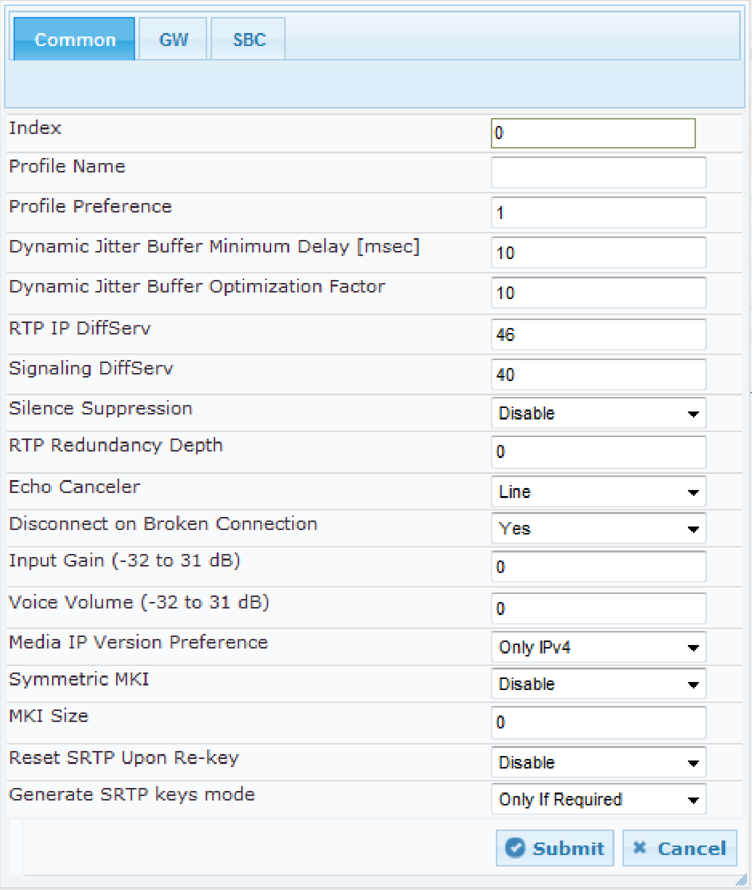
Many of the parameters in the IP Profile table have a corresponding "global" parameter. For calls that are not associated with any IP Profile, the settings of the "global" parameters are applied.

Note: IP Profiles can also be implemented when using a Proxy server (when the AlwaysUseRouteTable parameter is set to 1).

The following procedure describes how to configure IP Profiles in the Web interface. You can also configure IP Profiles using the table ini file parameter, IPProfile or the CLI command, configure voip > coders-and-profiles ip-profile.

➢ To configure an IP Profile:

1. Open the IP Profile Settings page (**Configuration** tab > **VoIP** menu > **Coders and Profiles** > **IP Profile Settings**).
2. Click **Add**; the following dialog box appears:



Index	0
Profile Name	
Profile Preference	1
Dynamic Jitter Buffer Minimum Delay [msec]	10
Dynamic Jitter Buffer Optimization Factor	10
RTP IP DiffServ	46
Signaling DiffServ	40
Silence Suppression	Disable
RTP Redundancy Depth	0
Echo Canceler	Line
Disconnect on Broken Connection	Yes
Input Gain (-32 to 31 dB)	0
Voice Volume (-32 to 31 dB)	0
Media IP Version Preference	Only IPv4
Symmetric MKI	Disable
MKI Size	0
Reset SRTP Upon Re-key	Disable
Generate SRTP keys mode	Only If Required

3. Configure an IP Profile according to the parameters described in the table below.

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

Table 23-5: IP Profile Settings Table Parameter Descriptions

Parameter	Description
Common	
Web: Index **[IpProfile_Index]**	Defines an index number for the new table record.
Web: Profile Name CLI: profile-name **[IpProfile_ProfileName]**	Defines an arbitrary name to easily identify the IP Profile. The valid value is a string of up to 20 characters.
Web: Profile Preference CLI: ip-preference [IpProfile_IpPreference]	Defines the priority of the IP Profile, where 20 is the highest priority and 1 the lowest priority. **Notes:** ▪ If an IP Profile and a Tel Profile apply to the same call, the coders and other common parameters of the profile with the highest preference are applied to the call. If the preference of the profiles is identical, the Tel Profile parameters are applied. ▪ If the coder lists of both an IP Profile and a Tel Profile apply to the same call, only the coders common to both are used. The order of the coders is determined by the preference. ▪ This parameter is applicable only to the Gateway application.
Web: Dynamic Jitter Buffer Minimum Delay CLI: 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. **Notes:** ▪ For more information on the Jitter Buffer, see Dynamic Jitter Buffer Operation on page 190. ▪ The corresponding global parameter is DJBufMinDelay.
Web: Dynamic Jitter Buffer Optimization Factor CLI: 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. **Notes:** ▪ For data (fax and modem) calls, set this parameter to 12. ▪ For more information on Jitter Buffer, see Dynamic Jitter Buffer Operation on page 190. ▪ The corresponding global parameter is DJBufOptFactor.
Web: RTP IP DiffServ CLI: rtp-ip-diffserv **[IpProfile_IPDiffServ]**	Defines the DiffServ value for Premium Media class of service (CoS) content. The valid range is 0 to 63. The default is 46. **Note:** The corresponding global parameter is PremiumServiceClassMediaDiffServ.
Web: Signaling DiffServ CLI: signaling-diffserv **[IpProfile_SigIPDiffServ]**	Defines the DiffServ value for Premium Control CoS content (Call Control applications).

Parameter	Description
	The valid range is 0 to 63. The default is 40. **Note:** The corresponding global parameter is PremiumServiceClassControlDiffServ.
Web: Silence Suppression CLI: sce [IpProfile_SCE]	Enables the Silence Suppression feature. When enabled, the device, upon detection of silence period during a call does not send packets, thereby conserving bandwidth during the VoIP call. ▪ [0] Disable (default) ▪ [1] Enable = Silence Suppression is enabled. ▪ [2] Enable Without Adaptation = A single silence packet is sent during a silence period (applicable only to G.729). **Notes:** ▪ If the coder is G.729, the value of the 'annexb' parameter of the 'fmtp' attribute in the SDP is determined by the following: ✓ This parameter is set to Disable [0]: 'annexb=no' ✓ This parameter is set to Enable [1]: 'annexb=yes' ✓ This parameter is set to Enable Without Adaptation [2] and IsCiscoSCEMode to [0]: 'annexb=yes' ✓ Enable Without Adaptation is set to Enable Without Adaptation [2] and IsCiscoSCEMode to [1]: 'annexb=no' ▪ The corresponding global parameter is EnableSilenceCompression.
Web: RTP Redundancy Depth CLI: rtp-redundancy-depth **[IpProfile_RTPRedundancyDepth]**	Enables the device to generate RFC 2198 redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver's end from the redundant data that arrives in subsequent packets. This is required, for example, in wireless networks where a high percentage (up to 50%) of packet loss can be experienced. ▪ **[0]** 0 = (Default) Disable. ▪ **[1]** 1 = Enable - previous voice payload packet is added to current packet. **Notes:** ▪ When enabled, you can configure the payload type, using the RFC2198PayloadType parameter. ▪ The RTP redundancy dynamic payload type can be included in the SDP, by using the EnableRTPRedundancyNegotiation parameter. ▪ The corresponding global parameter is RTPRedundancyDepth.
Web: Echo Canceler CLI: echo-canceller [IpProfile_EnableEchoCanceller]	Enables echo cancellation (i.e., echo from voice calls is removed). ▪ **[0]** Disable ▪ **[1]** Line (default) ▪ **[2]** Acoustic **Notes:** ▪ For more information on the Echo Cancellation feature,

Parameter	Description
	see Echo Cancellation on page 174. ▪ The corresponding global parameter is EnableEchoCanceller.
Web: Disconnect on Broken Connection CLI: disconnect-on-broken-connection **[IpProfile_DisconnectOnBrokenConnection]**	Enables the device to release the call if RTP packets are not received within a user-defined timeout, configured by the BrokenConnectionEventTimeout parameter. ▪ **[0]** No ▪ **[1]** Yes (default) **Notes:** ▪ This feature is applicable only if the RTP session is used without Silence Compression. If Silence Compression is enabled, the device doesn't detect a broken RTP connection. ▪ During a call, if the source IP address (from where the RTP packets are received) is changed without notifying the device, the device filters these RTP packets. To overcome this, set the DisconnectOnBrokenConnection parameter to 0; the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address. ▪ The corresponding global parameter is DisconnectOnBrokenConnection.
Web: Input Gain CLI: input-gain [IpProfile_InputGain]	Defines the pulse-code modulation (PCM) input gain control (in decibels). Gateway application: Defines the level of the received signal for Tel-to-IP calls. The valid range is -32 to 31 dB. The default is 0 dB. **Note:** The corresponding global parameter is InputGain.
Web: Voice Volume CLI: voice-volume [IpProfile_VoiceVolume]	Defines the voice gain control (in decibels). Gateway application: Defines the level of the transmitted signal for IP-to-Tel calls. The valid range is -32 to 31 dB. The default is 0 dB. **Note:** The corresponding global parameter is VoiceVolume.
Web: Media IP Version Preference CLI: media-ip-version-preference [IpProfile_MediaIPVersionPreference]	Defines the preferred RTP media IP addressing version for outgoing SIP calls. This is indicated in the "c=" field (Connection Information) of the SDP. ▪ [0] Only IPv4 = (Default) SDP offer includes only IPv4 media IP addresses. ▪ [1] Only IPv6 = SDP offer includes only IPv6 media IP addresses. ▪ [2] Prefer IPv4 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first media is IPv4. ▪ [3] Prefer IPv6 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first media is IPv6. **Notes:** ▪ This parameter is applicable only when the device offers an SDP. ▪ The IP addressing version is determined according to

Parameter	Description
	the first SDP "m=" field. ▪ The corresponding global parameter is MediaIPVersionPreference.

Web: Symmetric MKI
CLI: enable-symmetric-mki
[IpProfile_EnableSymmetricMKI]
	Enables symmetric MKI negotiation.
	▪ **[0]** Disable = (Default) The device includes the MKI in its SIP 200 OK response according to the SRTPTxPacketMKISize parameter (if set to 0, it is not included; if set to any other value, it is included with this value).
	▪ **[1]** Enable = The answer crypto line contains (or excludes) an MKI value according to the selected crypto line in the offer. For example, assume that the device receives an INVITE containing the following two crypto lines in SDP:
	`a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:TAaxNnQt8/qLQMnDuG4vxYfWl6K7eBK/ufk04pR4|2^31|1:1`
	`a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:bnuYZnMxSfUiGitviWJZmzr7OF3AiRO0l5Vnh0kH|2^31`
	The first crypto line includes the MKI parameter "1:1". In the 200 OK response, the device selects one of the crypto lines (i.e., '2' or '3'). Typically, it selects the first line that supports the crypto suite. However, for SRTP-to-SRTP in SBC sessions, it can be determined by the remote side on the outgoing leg. If the device selects crypto line '2', it includes the MKI parameter in its answer SDP, for example:
	`a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:R1VyA1xV/qwBjkEklu4kSJyl3wCtYeZLq1/QFuxw|2^31|1:1`
	If the device selects a crypto line that does not contain the MKI parameter, then the MKI parameter is not included in the crypto line in the SDP answer (even if the SRTPTxPacketMKISize parameter is set to any value other than 0).
	Note: The corresponding global parameter is EnableSymmetricMKI.

Web: MKI Size CLI: mki-size **[IpProfile_MKISize]**	Defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. The valid value is 0 to 4. The default is 0 (i.e., new keys are generated without MKI). **Notes:** ▪ Gateway application: The device only initiates the MKI size. ▪ SBC application: The device can forward MKI size as is for SRTP-to-SRTP flows or override the MKI size during negotiation. This can be done on the inbound or outbound leg. ▪ The corresponding global parameter is SRTPTxPacketMKISize.
Web: Reset SRTP Upon Re-key	Enables synchronization of the SRTP state between the

Parameter	Description
CLI: reset-srtp-upon-re-key **[IpProfile_ResetSRTPStateUponRekey]**	device and a server when a new SRTP key is generated upon a SIP session expire. This feature ensures that the roll-over counter (ROC), one of the parameters used in the SRTP encryption/decryption process of the SRTP packets is synchronized on both sides for transmit and receive packets. ▪ **[0]** Disable = (Default) ROC is not reset on the device side. ▪ **[1]** Enable = If the session expires causing a session refresh through a re-INVITE, the device or server generates a new key and the device resets the ROC index (and other SRTP fields) as done by the server, resulting in a synchronized SRTP. **Notes:** ▪ If this feature is disabled and the server resets the ROC upon a re-key generation, one-way voice may occur. ▪ The corresponding global parameter is ResetSRTPStateUponRekey.
Generate SRTP keys mode CLI: generate-srtp-keys **[IpProfile_GenerateSRTPKeys]**	Enables the device to generate a new SRTP key upon receipt of a re-INVITE with this SIP entity. ▪ **[0]** Only If Required= (Default) The device generates an SRTP key only if necessary. ▪ **[1]** Always = The device always generates a new SRTP key.
Jitter Buffer Max Delay CLI: 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.
GW	
Web: Coders Group ID CLI: coders-group-id [IpProfile_CodersGroupID]	Defines coders supported by this SIP entity, by assigning a Coders Group. The value, Default Coders Group represents the coders configured in the Coders table (see Configuring Coders on page 307). All other optional values (e.g., Coders Group 1), represent the coders defined for the specific Coder Group configured in the Coder Group Settings table (see Configuring Coder Groups on page 310).
Web: Fax Signaling Method CLI: fax-sig-method [IpProfile_IsFaxUsed]	Defines the SIP signaling method for establishing and transmitting a fax session when the device detects a fax. ▪ [0] No Fax = (Default) No fax negotiation using SIP signaling. The fax transport method is according to the FaxTransportMode parameter. ▪ [1] T.38 Relay = Initiates T.38 fax relay. ▪ [2] G.711 Transport = Initiates fax/modem using the coder G.711 A-law/Mu-law with adaptations (see Note below). ▪ [3] Fax Fallback = Initiates T.38 fax relay. If the T.38 negotiation fails, the device re-initiates a fax session using the coder G.711 A-law/□-law with adaptations (see the Note below).

Parameter	Description
	Notes: ▪ Fax adaptations (for options 2 and 3): ✓ 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 ▪ If the device initiates a fax session using G.711 (option 2 or 3), a 'gpmd' attribute is added to the SDP in the following format: ✓ For A-law: 'a=gpmd:8 vbd=yes;ecan=on' ✓ For Mu-law: 'a=gpmd:0 vbd=yes;ecan=on' ▪ When this parameter is set to 1, 2, or 3, the parameter FaxTransportMode is ignored. ▪ When this parameter is set to 0, T.38 might still be used without the control protocol's involvement. To completely disable T.38, set FaxTransportMode to a value other than 1. ▪ For more information on fax transport methods, see Fax/Modem Transport Modes on page 177. ▪ The corresponding global parameter is IsFaxUsed.
Web: Remote RTP Base UDP Port CLI: remote-base-udp-port [IpProfile_RemoteBaseUDPPort]	Defines the lower boundary of the UDP port range used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). Once this parameter is configured, the device calculates the UDP port range (lower to upper boundary). For more information on determining the port range, see Configuring RTP Base UDP Port on page 194. The range of possible UDP ports is 6,000 to 65,535. The default base UDP port is 6000. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If you set this parameter to a value that is not a factor of 10, the following message is generated: "invalid local RTP port". ▪ You can also configure a UDP port range per Media Realm (see Configuring Media Realms on page 259). ▪ The corresponding global parameter is BaseUDPPort.
Web: CNG Detector Mode CLI: cng-mode [IpProfile_CNGmode]	Enables the detection of the fax calling tone (CNG) and defines the detection method. ▪ [0] Disable = (Default) The originating fax does not detect CNG; the device passes the CNG signal transparently to the remote side. ▪ [1] Relay = The originating fax detects CNG. The device sends CNG packets to the remote side according to T.38 (if IsFaxUsed is set to 1) and the fax session is started. A SIP Re-INVITE message is not sent and the fax session starts by the terminating fax. This option is useful, for example, when the originating fax 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 (i.e., originating fax).

Parameter	Description
	To also send a Re-INVITE message upon detection of a fax CNG tone in this mode, set the parameter FaxCNGMode to 1 or 2. - [2] Event Only = The originating fax detects CNG and a fax session is started by the originating fax, using the Re-INVITE message. Typically, T.38 fax session starts when the preamble signal is detected by the answering fax. Some SIP devices do not support the detection of this fax signal on the answering fax and thus, in these cases, it is possible to configure the device to start the T.38 fax session when the CNG tone is detected by the originating fax. However, this mode is not recommended. **Note:** The corresponding global parameter is CNGDetectorMode.
Web: Vxx Modem Transport Type CLI: vxx-transport-type [IpProfile_VxxTransportType]	Defines the modem transport type. - [-1] Not Configured = The settings of the global parameters are used: ✓ V21ModemTransportType ✓ V22ModemTransportType ✓ V23ModemTransportType ✓ V32ModemTransportType ✓ V34ModemTransportType - [0] Disable = Transparent. - [2] Enable Bypass (Default) - [3] Events Only = Transparent with Events. For a detailed description of this parameter per modem type, see the relevant global parameter (listed above).
Web: NSE Mode CLI: nse-mode [IpProfile_NSEMode]	Enables Cisco's compatible fax and modem bypass mode, Named Signaling Event (NSE) packets. - [0] Disable (Default) - [1] Enable In NSE bypass mode, the device starts using G.711 A-Law (default) or G.711□Law, according to the FaxModemBypassCoderType parameter. The payload type for these G.711 coders is a standard one (8 for G.711 A-Law and 0 for G.711 □Law). The parameters defining payload type for the 'old' Bypass mode FaxBypassPayloadType and ModemBypassPayloadType are not used with NSE Bypass. The bypass packet interval is configured according to the FaxModemBypassBasicRtpPacketInterval parameter. The SDP contains the following line: 'a=rtpmap:100 X-NSE/8000'. **Notes:** - When enabled, the following conditions must also be met: ✓ The Cisco gateway must include the following definition: 'modem passthrough nse payload-type 100 codec g711alaw'. ✓ Set the Modem transport type to Bypass mode

Parameter	Description
	(VxxModemTransportType is set to 2) for all modems. ✓ Set the NSEPayloadType parameter to 100. ▪ The corresponding global parameter is NSEMode.
Web: Play RB Tone to IP CLI: play-rbt-to-ip [IpProfile_PlayRBTone2IP]	Enables the device to play a ringback tone to the IP side for IP-to-Tel calls. ▪ [0] Disable (Default) ▪ [1] Enable = Plays a ringback tone after a SIP 183 session progress response is sent. **Notes:** ▪ To enable the device to send a 183/180+SDP responses, set the EnableEarlyMedia parameter to 1. ▪ If the EnableDigitDelivery parameter is set to 1, the device doesn't play a ringback tone to IP and doesn't send 183 or 180+SDP responses. ▪ Digital interfaces: If this parameter is enabled and EnableEarlyMedia is set to 1, the device plays a ringback tone according to the following: ✓ CAS: The device opens a voice channel, sends a 183+SDP response, and then plays a ringback tone to IP. ✓ ISDN: If a Progress or an Alerting message with PI (1 or 8) is received from the ISDN, the device opens a voice channel, sends a 183+SDP or 180+SDP response, but doesn't play a ringback tone to IP. If PI (1 or 8) is received from the ISDN, the device assumes that ringback tone is played by the ISDN switch; otherwise, the device plays a ringback tone to IP after receiving an Alerting message from the ISDN. It sends a 180+SDP response, signaling to the calling party to open a voice channel to hear the played ringback tone. ▪ The corresponding global parameter is PlayRBTone2IP.
Web: Early Media CLI: early-media [IpProfile_EnableEarlyMedia]	Enables the Early Media feature for sending media (e.g., ringing) before the call is established. ▪ [0] Disable (default) ▪ [1] Enable ✓ Digital: The device sends a SIP 18x response with SDP, allowing the media stream to be established before the call is answered. ✓ Analog: The device sends a SIP 183 Session Progress response with SDP instead of a 180 Ringing, allowing the media stream to be established before the call is answered. **Notes:** ▪ Digital: The inclusion of the SDP in the 18x response depends on the ISDN Progress Indicator (PI). The SDP is sent only if PI is set to 1 or 8 in the received Proceeding, Alerting, or Progress PRI messages. See also the ProgressIndicator2IP parameter, which if set to 1 or 8, the device behaves as if it received the ISDN

Parameter	Description
	messages with the PI. ✓ CAS: See the ProgressIndicator2IP parameter. ✓ ISDN: Sending a 183 response depends on the ISDN PI. It is sent only if PI is set to 1 or 8 in the received Proceeding or Alerting PRI messages. Sending 183 response also depends on the ReleaseIP2ISDNCallOnProgressWithCause parameter, which must be set to any value other than 2. ▪ See also the IgnoreAlertAfterEarlyMedia parameter. This parameter allows, for example, to interwork Alert with PI to SIP 183 with SDP instead of 180 with SDP. ▪ You can also configure early SIP 183 response immediately upon the receipt of an INVITE, using the EnableEarly183 parameter. ▪ Analog: To send a 183 response, you must also set the ProgressIndicator2IP parameter to 1. If set to 0, a 180 Ringing response is sent. ▪ The corresponding global parameter is EnableEarlyMedia.
Web: Progress Indicator to IP CLI: prog-ind-to-ip [IpProfile_ProgressIndicator2IP]	Defines the progress indicator (PI) sent to the IP. ▪ [-1] Not Configured = (Default) ✓ Analog: Default values are used (1 for FXO interfaces and 0 for FXS interfaces). ✓ Digital ISDN: The PI received in ISDN Proceeding, Progress, and Alerting messages is used, as described in the options below. ▪ [0] No PI = ✓ Analog: For IP-to-Tel calls, the device sends a 180 Ringing response to IP after placing a call to a phone (FXS) or PBX (FXO). ✓ Digital: For IP-to-Tel calls, the device sends 180 Ringing response to the IP after receiving an ISDN Alerting or (for CAS) after placing a call to the PBX/PSTN. ▪ [1] PI = 1: ✓ Analog: For IP-to-Tel calls, if the EnableEarlyMedia parameter is set to 1, the device sends a 183 Session Progress message with SDP immediately after a call is placed to a phone/PBX. This is used to cut-through the voice path before the remote party answers the call. This allows the originating party to listen to network call progress tones such as ringback tone or other network announcements. ✓ Digital: For IP-to-Tel calls, if the parameter EnableEarlyMedia is set to 1, the device sends 180 Ringing with SDP in response to an ISDN Alerting or it sends a 183 Session Progress message with SDP in response to only the first received ISDN Proceeding or Progress message after a call is placed to PBX/PSTN over the trunk. ▪ [8] PI = 8: same as PI = 1.

Parameter	Description
	Note: The corresponding global parameter is ProgressIndicator2IP.
Web: Copy Destination Number to Redirect Number CLI: copy-dst-to-redirect-number [IpProfile_CopyDest2RedirectNumber]	Enables the device to copy the called number, received in the SIP INVITE message, to the redirect number in the outgoing Q.931 Setup message, for IP-to-Tel calls. Thus, even if there is no SIP Diversion or History header in the incoming INVITE message, the outgoing Q.931 Setup message will contain a redirect number. ▪ [0] Disable (default). ▪ [1] After Manipulation = Copies the called number after manipulation. The device first performs IP-to-Tel destination phone number manipulation, and only then copies the manipulated called number to the redirect number sent in the Q.931 Setup message to the Tel. Thus, the called and redirect numbers are the same. ▪ [2] Before Manipulation = Copies the called number before manipulation. The device first copies the original called number to the SIP Diversion header, and then performs IP-to-Tel destination phone number manipulation. Thus, the called (i.e., SIP To header) and redirect (i.e., SIP Diversion header) numbers are different. **Note:** The corresponding global parameter is CopyDest2RedirectNumber.
Web: Media Security Behavior CLI: media-security-behaviour [IpProfile_MediaSecurityBehaviour]	Defines the handling of SRTP for this SIP entity. ▪ [-1] Not Configured = Applies the settings of the corresponding global parameter, MediaSecurityBehaviour. ▪ [0] Preferable = (Default) The device initiates encrypted calls to this SIP entity. However, if negotiation of the cipher suite fails, an unencrypted call is established. The device accepts incoming calls received from the SIP entity that don't include encryption information. ▪ [1] Mandatory = The device initiates encrypted calls to this SIP entity, but if negotiation of the cipher suite fails, the call is terminated. The device rejects incoming calls received from the SIP entity that don't include encryption information. ▪ [2] Disable = This SIP entity does not support encrypted calls (i.e., SRTP). ▪ [3] Preferable - Single Media = The device sends SDP with a single media ('m=') line only (e.g., m=audio 6000 RTP/AVP 4 0 70 96) with RTP/AVP and crypto keys. The SIP entity can respond with SRTP or RTP parameters: ✓ If the SIP entity does not support SRTP, it uses RTP and ignores the crypto lines. ✓ If the device receives an SDP offer with a single media (as shown above) from the SIP entity, it responds with SRTP (RTP/SAVP) if the EnableMediaSecurity parameter is set to 1. If SRTP is not supported (i.e., EnableMediaSecurity is set to 0), it responds with RTP.

Parameter	Description
	✓ If two 'm=' lines are received in the SDP offer, the device prefers the SAVP (secure audio video profile), regardless of the order in the SDP. **Notes:** ▪ This parameter is applicable only when the EnableMediaSecurity parameter is set to 1. ▪ The corresponding global parameter is MediaSecurityBehaviour.
Web: Number of Calls Limit CLI: call-limit [IpProfile_CallLimit]	Defines the maximum number of concurrent calls (incoming and outgoing). If the number of concurrent calls reaches this limit, the device rejects any new incoming and outgoing calls belonging to this IP Profile. This parameter can also be set to the following: ▪ [-1] = (Default) No limitation on calls. ▪ [0] = All calls are rejected. **Note:** For Gateway IP-to-IP calls, you can configure the device to route calls to an alternative IP Group when this maximum number of concurrent calls is reached. To do so, you need to add an alternative routing rule in the Outbound IP Routing table that reroutes the call to an alternative IP Group. You also need to add a rule to the Reason for Alternative Routing table to initiate an alternative rule for Tel-to-IP calls using cause 805.
Web: First Tx DTMF Option CLI: first-tx-dtmf-option [IpProfile_FirstTxDtmfOption]	Defines the first preferred transmit DTMF negotiation method. ▪ [0] Not Supported = No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType (for transmit and receive). ▪ [1] INFO (Nortel) = Sends DTMF digits according to IETF Internet-Draft draft-choudhuri-sip-info-digit-00. ▪ [2] NOTIFY = Sends DTMF digits according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01. ▪ [3] INFO (Cisco) = Sends DTMF digits according to the Cisco format. ▪ [4] RFC 2833 (Default) = The device: ✓ negotiates RFC 2833 payload type using local and remote SDPs. ✓ sends DTMF packets using RFC 2833 payload type according to the payload type in the received SDP. ✓ expects to receive RFC 2833 packets with the same payload type as configured by the parameter RFC2833PayloadType. ✓ removes DTMF digits in transparent mode (as part of the voice stream). ▪ [5] INFO (Korea) = Sends DTMF digits according to the Korea Telecom format. **Notes:** ▪ When out-of-band DTMF transfer is used ([1], [2], [3], or [5]), the DTMFTransportType parameter is automatically set to 0 (DTMF digits are removed from the RTP stream).

Parameter	Description
	▪ If an ISDN phone user presses digits (e.g., for interactive voice response / IVR applications such as retrieving voice mail messages), ISDN Information messages received by the device for each digit are sent in the voice channel to the IP network as DTMF signals, according to the settings of this parameter. ▪ The corresponding global parameter is TxDTMFOption.
Web: Second Tx DTMF Option CLI: second-tx-dtmf-option [IpProfile_SecondTxDtmfOption]	Defines the second preferred transmit DTMF negotiation method. For a description, see IpProfile_FirstTxDtmfOption (above). **Note:** The corresponding global parameter is TxDTMFOption
Web: Rx DTMF Option CLI: rx-dtmf-option [IpProfile_RxDTMFOption]	Enables the device to declare the RFC 2833 'telephony-event' parameter in the SDP. ▪ [0] Not Supported ▪ [3] Supported (default) The device is always receptive to RFC 2833 DTMF relay packets. Thus, it is always correct to include the 'telephony-event' parameter by default in the SDP. However, some devices use the absence of the 'telephony-event' in the SDP to decide to send DTMF digits in-band using G.711 coder. If this is the case, set this parameter to 0. **Note:** The corresponding global parameter is RxDTMFOption.
Web: Hold CLI: enable-hold [IpProfile_EnableHold]	Enables the Call Hold feature (analog interfaces) and interworking of the Hold/Retrieve supplementary service from ISDN PRI to SIP (digital interfaces). For analog: The Call Hold feature allows users, connected to the device, to place a call on hold (or remove from hold), using the phone's Hook Flash button. ▪ [0] Disable ▪ [1] Enable (default) **Notes:** ▪ Digital interfaces: To interwork the Hold/Retrieve supplementary service from SIP to ISDN (QSIG and Euro ISDN), set the EnableHold2ISDN parameter to 1. ▪ Analog interfaces: To use the call hold service, the devices at both ends must support this option. ▪ The corresponding global parameter is EnableHold.
Web: Add IE In Setup CLI: add-ie-in-setup [IpProfile_AddIEInSetup]	Defines an optional Information Element (IE) data (in hex format) which is added to ISDN Setup messages. For example, to add IE '0x20,0x02,0x00,0xe1', enter the value "200200e1". **Notes:** ▪ This IE is sent from the Trunk Group IDs that are defined by the SendIEonTG parameter . ▪ You can configure different IE data for Trunk Groups by configuring this parameter for different IP Profiles and then assigning the required IP Profile ID in the Inbound IP Routing table (PSTNPrefix).

Parameter	Description
	▪ This feature is similar to that of the EnableISDNTunnelingIP2Tel parameter. If both parameters are configured, the EnableISDNTunnelingIP2Tel parameter takes precedence. ▪ The corresponding global parameter is AddIEinSetup.
Web: QSIG Tunneling CLI: enable-qsig-tunneling [IpProfile_EnableQSIGTunneling]	Enables QSIG tunneling-over-SIP for this SIP entity. This is according to IETF Internet-Draft draft-elwell-sipping-qsig-tunnel-03 and ECMA-355 and ETSI TS 102 345. ▪ [0] Disable (default). ▪ [1] Enable = Enables QSIG tunneling from QSIG to SIP, and vice versa. All QSIG messages are sent as raw data in corresponding SIP messages using a dedicated message body. **Notes:** ▪ QSIG tunneling must be enabled on originating and terminating devices. ▪ To enable this function, set the ISDNDuplicateQ931BuffMode parameter to 128 (i.e., duplicate all messages). ▪ To define the format of encapsulated QSIG messages, use the QSIGTunnelingMode parameter. ▪ Tunneling according to ECMA-355 is applicable to all ISDN variants (in addition to the QSIG protocol). ▪ For more information on QSIG tunneling, see QSIG Tunneling on page 358. ▪ The corresponding global parameter is EnableQSIGTunneling.
Web: Early 183 CLI: enable-early-183 [IpProfile_EnableEarly183]	Enables the device to send SIP 183 responses with SDP to the IP upon receipt of INVITE messages. This parameter is applicable to IP-to-Tel (ISDN) and IP-to-IP calls, and applies to all calls. ▪ [0] Disable (default) ▪ [1] Enable = ✓ IP-to-Tel calls: By sending the 183 response, the device opens an RTP channel before receiving the "progress" tone from the ISDN side. The device sends RTP packets immediately upon receipt of an ISDN Progress, Alerting with Progress indicator, or Connect message according to the initial negotiation without sending the 183 response again, thereby saving response time and avoiding early media clipping. ✓ IP-to-IP calls: Sending the 183 response enables SIP servers that require a stream of early media, to keep sessions open. **Notes:** ▪ To enable this feature, set the EnableEarlyMedia parameter to 1. ▪ When the BChannelNegotiation parameter is set to Preferred or Any, the EnableEarly183 parameter is ignored and a SIP 183 is not sent upon receipt of an

Parameter	Description
	INVITE. In such a case, you can set the ProgressIndicator2IP parameter to 1 (PI = 1) for the device to send a SIP 183 upon receipt of an ISDN Call Proceeding message. - The corresponding global parameter is EnableEarly183.
Web: Early Answer Timeout CLI: early-answer-timeout [IpProfile_EarlyAnswerTimeout]	Defines the duration (in seconds) that the device waits for an ISDN Connect message from the called party (Tel side), started from when it sends a Setup message. If this timer expires, the call is answered by sending a SIP 200 OK message (to the IP side). The valid range is 0 to 2400. The default is 0 (i.e., disabled). **Note:** The corresponding global parameter is EarlyAnswerTimeout.
SBC	
Web: Extension Coders Group ID CLI: sbc-ext-coders-group-id [IpProfile_SBCExtensionCodersGroupID]	Assigns a Coder Group ID used for Extended (additional) coders, added to the outgoing leg for this SIP entity. 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). Therefore, to allow IP entities to communicate with each other regardless of their capabilities, an Extended coders table with at least one coder that is supported by each IP entity needs to be assigned to each IP Group. Therefore, each offer destined to specific IP Groups includes this coder. **Note:** To configure Coders Groups, see Configuring Coder Groups on page 310.
Web: Transcoding Mode CLI: transcoding-mode [IpProfile_TranscodingMode]	Defines the voice transcoding mode (media negotiation) for this SIP entity. - [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 between for this 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). **Notes:** - To implement transcoding, you must configure the number of required DSP channels for transcoding (using the MediaChannels parameter). Each transcoding session uses two DSP resources. - The corresponding global parameter is TranscodingMode.

Parameter	Description
Allowed Media Types CLI: sbc-allowed-media-types [IPProfile_SBCAllowedMediaTypes]	Defines media types permitted for this SIP entity. The media type appears in the SDP 'm=' line (e.g., 'm=audio'). The device permits only media types that appear in both the SDP offer and this configured list. If no common media types exist between the SDP offer and this list, the device drops the call. The valid value is a string of up to 64 characters. To configure multiple media types, separate the strings with a comma, e.g., "media, audio" (without quotes). By default, no media types are configured (i.e., all media types are permitted).
Web: Allowed Coders Group ID CLI: sbc-allowed-coders-group-id [IpProfile_SBCAllowedCodersGroupID]	Assigns an Allowed Coders Group to this SIP entity. This defines audio (voice) coders that can be used for this SIP entity. To configure Allowed Coders Groups, see Configuring Allowed Audio Coder Groups on page 543. For a description of the Allowed Coders feature, see "Restricting Coders" on page 513.
Web: Allowed Video Coders Group ID CLI: sbc-allowed-video-coders-group-id [IPProfile_SBCAllowedVideoCodersGroupID]	Assigns an Allowed Video Coders Group to this SIP entity. This defines permitted video coders when forwarding video streams to the SIP entity. The video coders are listed in the "video" media type in the SDP (i.e., 'm=video' line). For this SIP entity, the device uses only video coders that appear in both the SDP offer and the Allowed Video Coders Group ID. By default, no Allowed Video Coders Group is assigned (i.e., all video coders are allowed). To configure Allowed Video Coders Groups, see Configuring Allowed Video Coder Groups on page 544.
Web: Allowed Coders Mode CLI: sbc-allowed-coders-mode [IpProfile_SBCAllowedCodersMode]	Defines the mode of the Allowed Coders feature for this SIP entity. ▪ [0] Restriction = In the incoming SDP offer, the device uses only Allowed coders; the rest are removed from the SDP offer (i.e., only coders common between those in the received SDP offer and the Allowed coders are used). If an Extension Coders Group is also assigned (using the 'Extension Coders Group ID' parameter, above), these coders are added to the SDP offer. ▪ [1] Preference = The device re-arranges the priority (order) of the coders in the incoming SDP offer according to their order of appearance in the Allowed Coders Group or Allowed Video Coders tables. The coders received in the SDP offer are listed after the Allowed coders. ▪ [2] Restriction and Preference = Performs both Restriction and Preference. **Notes:** ▪ This parameter is applicable only if Allowed coders are assigned to the IP Profile (using the 'Allowed Coders Group ID' or 'Allowed Video Coders Group ID' parameters). ▪ For more information on the Allowed Coders feature,

Parameter	Description
	see Restricting Coders on page 513.
Web: SBC Media Security Behavior CLI: sbc-media-security-behaviour [IpProfile_SBCMediaSecurityBehaviour]	Defines the handling of RTP and SRTP for this SIP entity. ▪ [0] As is = (Default) No special handling for RTP\SRTP is done. ▪ [1] SRTP = SBC legs negotiate only SRTP media lines, and RTP media lines are removed from the incoming SDP offer\answer. ▪ [2] RTP = SBC legs negotiate only RTP media lines, and SRTP media lines are removed from the incoming offer\answer. ▪ [3] Both = Each offer\answer is extended (if not already) to two media lines - one RTP and the other SRTP. If two SBC legs (after offer\answer negotiation) use different security types (i.e., one RTP and the other SRTP), the device performs RTP-SRTP transcoding. To transcode between RTP and SRTP, the following prerequisites must be met: ▪ At least one supported SDP "crypto" attribute and parameters. ▪ EnableMediaSecurity must be set to 1. If one of the above transcoding prerequisites is not met, then: ▪ any value other than “As is” is discarded. ▪ if the incoming offer is SRTP, force transcoding, coder transcoding, and DTMF extensions are not applied.
Web: RFC 2833 Behavior CLI: sbc-rfc2833-behavior [IpProfile_SBCRFC2833Behavior]	Defines the handling of RFC 2833 SDP offer-answer negotiation for this SIP entity. ▪ [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.
Web: Alternative DTMF Method CLI: sbc-alternative-dtmf-method [IpProfile_SBCAlternativeDTMFMethod]	The device's first priority for DTMF method at each leg is RFC 2833. Thus, if the device successfully negotiates RFC 2833 for this SIP entity, the chosen DTMF method for this leg is RFC 2833. When RFC 2833 negotiation fails, the device uses this 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

Parameter	Description
	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.
Web: P-Asserted-Identity CLI: sbc-assert-identity [IpProfile_SBCAssertIdentity]	Defines the device's handling of the SIP P-Asserted-Identity header for this SIP entity. This header indicates how the outgoing SIP message asserts identity. ▪ [0] As Is = (Default) P-Asserted Identity header is not affected and the device uses the same P-Asserted-Identity header (if present) in the incoming message for the outgoing message. ▪ [1] Add = Adds a P-Asserted-Identity header. The header's values are taken from the source URL. ▪ [2] Remove = Removes the P-Asserted-Identity header. **Notes:** ▪ This parameter affects only the initial INVITE request. ▪ The corresponding global parameter is SBCAssertIdentity.
Web: Diversion Mode CLI: sbc-diversion-mode [IpProfile_SBCDiversionMode]	Defines the device’s handling of the SIP Diversion header for this SIP entity. ▪ [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 521. **Note:** If the Diversion header is used, you can specify the URI type (e.g., "tel:") to use in the header, using the SBCDiversionUriType parameter.
Web: History-Info Mode CLI: sbc-history-info-mode [IpProfile_SBCHistoryInfoMode]	Defines the device’s handling of the SIP History-Info header for this SIP entity. ▪ [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 521.
Web: Fax Coders Group ID CLI: sbc-fax-coders-group-id [IpProfile_SBCFaxCodersGroupID]	Assigns a Coders Group ID to define the supported fax coders for fax negotiation for this SIP entity. For configuring Coder Groups, see Configuring Coder Groups on page 310. **Note:** The parameter is applicable only if you set the IpProfile_SBCFaxBehavior parameter to a value other

Parameter	Description
	than [0].
Web: Fax Behavior CLI: sbc-fax-behavior [IpProfile_SBCFaxBehavior]	Defines the fax-offer negotiation method for this SIP entity. ▪ [0] As Is = (Default) Pass 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.
Web: Fax Offer Mode CLI: sbc-fax-offer-mode [IpProfile_SBCFaxOfferMode]	Defines the coders included in the outgoing SDP offer (sent to the called "fax") for this SIP entity. ▪ [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, then the device uses this coder. If no match exists, then 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].
Web: Fax Answer Mode CLI: sbc-fax-answer-mode [IpProfile_SBCFaxAnswerMode]	Defines the coders included in the outgoing SDP answer (sent to the calling "fax") for this SIP entity. ▪ [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].
Web: PRACK Mode CLI: sbc-prack-mode [IpProfile_SbcPrackMode]	Defines the device's handling of SIP PRACK messages for this SIP entity. ▪ [1] Optional = PRACK is optional. If required, the device performs the PRACK process on behalf of the

Parameter	Description
	SIP entity. ▪ [2] Mandatory = PRACK is required for this SIP entity. Calls from endpoints that do not support PRACK are rejected. Calls destined to these endpoints are also required to support PRACK. ▪ [3] Transparent (default) = The device does not intervene with the PRACK process and forwards the request as is.
Web: Session Expires Mode CLI: sbc-session-expires-mode [IpProfile_SBCSessionExpiresMode]	Defines the required session expires mode for this SIP entity. ▪ [0] Transparent = (Default) The device does not interfere with the session expires negotiation. ▪ [1] Observer = If the SIP Session-Expires header is present, the device does not interfere, but maintains an independent timer for each leg to monitor the session. If the session is not refreshed on time, the device disconnects the call. ▪ [2] Not Supported = The device does not allow a session timer with this SIP entity. ▪ [3] Supported = The device enables the session timer with this SIP entity. If the incoming SIP message does not include any session timers, the device adds the session timer information to the sent message. You can configure the value of the Session-Expires and Min-SE headers, using the SBCSessionExpires and SBCMinSE parameters, respectively.
Web: Remote Update Support CLI: sbc-rmt-update-supp [IpProfile_SBCRemoteUpdateSupport]	Defines whether this SIP entity supports the SIP UPDATE message. ▪ [0] Not Supported = UPDATE message is not supported. ▪ [1] Supported Only After Connect = UPDATE message is supported only after the call is connected. ▪ [2] Supported = (Default) UPDATE message is supported during call setup and after call establishment.
Web: Remote re-INVITE CLI: sbc-rmt-re-invite-supp [IpProfile_SBCRemoteReinviteSupport]	Defines whether the destination UA of the re-INVITE request supports re-INVITE messages and if so, whether it supports re-INVITE with or without SDP. ▪ [0] Not Supported = re-INVITE is not supported and the device does not forward re-INVITE requests. The device sends a SIP response to the re-INVITE request, which can either be a success or a failure, depending on whether the device can bridge the media between the endpoints. ▪ [1] Supported only with SDP = re-INVITE is supported, but only with SDP. If the incoming re-INVITE arrives without SDP, the device creates an SDP and adds it to the outgoing re-INVITE. ▪ [2] Supported = (Default) re-INVITE is supported with or without SDP.
Web: Remote Delayed Offer Support CLI: sbc-rmt-delayed-offer	Defines whether the remote endpoint supports delayed offer (i.e., initial INVITEs without an SDP offer).

Parameter	Description
[IpProfile_SBCRemoteDelayedOfferSupport]	▪ [0] Not Supported = Initial INVITE requests without SDP are not supported. ▪ [1] Supported = (Default) Initial INVITE requests without SDP are supported. **Note:** For this parameter to function, you need to configure a valid Extension Coders Group ID for IP Profiles that do not support delayed offer.
Web: Remote REFER Behavior CLI: sbc-rmt-refer-behavior [IpProfile_SBCRemoteReferBehavior]	Defines the device's handling of REFER requests for this SIP entity. ▪ [0] Regular = (Default) Refer-To header is unchanged and the device forwards the REFER as is. ▪ [1] Database URL = Changes the Refer-To header so that the re-routed INVITE is sent through the SBC: a. Before forwarding the REFER request, the device changes the host part to the device's IP address and adds a special prefix ("T~&R_") to the Contact user part. b. The incoming INVITE is identified as a REFER-resultant INVITE according to this special prefix. c. The device replaces the host part in the Request-URI with the host from the REFER contact. The special prefix remains in the user part for regular classification, manipulation, and routing. The special prefix can also be used for specific routing rules for REFER-resultant INVITEs. d. The special prefix is removed before the resultant INVITE is sent to the destination. ▪ [2] IP Group Name = Sets the host part in the REFER message to the name defined for the IP Group (in the IP Group table). ▪ [3] Handle Locally = Handles the incoming REFER request itself without forwarding the REFER. The device generates a new INVITE to the alternative destination according to the rules in the IP-to-IP Routing table (the 'Call Trigger' field must be set to REFER). **Note:** The corresponding global parameter is SBCReferBehavior.
Web: Remote 3xx Behavior CLI: sbc-rmt-3xx-behavior [IpProfile_SBCRemote3xxBehavior]	Defines the device's handling of SIP 3xx redirect responses for this SIP entity. By default, the device's handling of SIP 3xx responses is to send the Contact header unchanged. However, some SIP entities may support different versions of the SIP 3xx standard while others may not even support SIP 3xx. When enabled, the device handles SIP redirections between different subnets (e.g., between LAN and WAN sides). This is required when the new address provided by the redirector (Redirect sever) may not be reachable by the far-end user (FEU) located in another subnet. For example, a far-end user (FEU) in the WAN sends a SIP request via the device to a Redirect 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

Parameter	Description
	this response as is (i.e., with the original Contact header), the FEU is unable to reach the new destination. ▪ [0] Transparent = (Default) The device forwards the received SIP 3xx response as is, without changing the Contact header (i.e.,transparent handling). ▪ [1] Database URL = The device changes the Contact header so that the re-route request is sent through the device. The device changes the URI in the Contact header of the received SIP 3xx response to its own URI and adds a special user prefix ("T~&R_"), which is then sent to the FEU. The FEU then sends a new INVITE to the device, which the device then sends to the correct destination. ▪ [2] Handle Locally = The device handles SIP 3xx responses on behalf of the dialog-initiating UA and retries the request (e.g., INVITE) using one or more alternative URIs included in the 3xx response. The device sends the new request to the alternative destination according to the IP-to-IP Routing table (the 'Call Trigger' field must be set to 3xx). **Notes:** ▪ When this parameter is changed from 1 to 0, new 3xx Contact headers remain unchanged. However, requests with the special prefix continue using the device's database to locate the new destination. ▪ Only one database entry is supported for the same host, port, and transport combination. For example, the following URLs cannot be distinguished by the device: ✓ sip:10.10.10.10:5060;transport=tcp;param=a ✓ sip:10.10.10.10:5060;transport=tcp;param=b ▪ The database entry expires two hours after the last use. ▪ The maximum number of destinations (i.e., database entries) is 50. ▪ The corresponding global parameter is SBC3xxBehavior.
Web: Remote Multiple 18x CLI: sbc-rmt-mltple-18x-supp [IpProfile_SBCRemoteMultiple18xSupport]	Defines whether multiple 18x responses including 180 Ringing, 181 Call is Being Forwarded, 182 Call Queued, and 183 Session Progress are forwarded to the caller, for this SIP entity. ▪ [0] Not Supported = Only the first 18x response is forwarded to the caller. ▪ [1] Supported = (Default) Multiple 18x responses are forwarded to the caller.
Web: Remote Early Media Response Type CLI: sbc-rmt-early-media-resp [IpProfile_SBCRemoteEarlyMediaResponseType]	Defines the SIP provisional response type - 180 or 183 - for forwarding early media to the caller, for this SIP entity. ▪ [0] Transparent = (Default) All early media response types are supported; the device forwards all responses as is (unchanged). ▪ [1] 180 = Early media is sent as 180 response only. ▪ [2] 183 = Early media is sent as 183 response only.
Web: Remote Early Media	Defines whether the remote side can accept early media

Parameter	Description
CLI: sbc-rmt-early-media-supp [IpProfile_SBCRemoteEarlyMediaSupport]	or not. ▪ [0] Not Supported = Early media is not supported. ▪ [1] Supported = (Default) Early media is supported.
Web: Enforce MKI Size CLI: sbc-enforce-mki-size [IpProfile_SBCEnforceMKISize]	Enables MKI length negotiation for SRTP-to-SRTP flows between SIP networks (i.e., IP Groups). This includes the capability of modifying the MKI length on the inbound or outbound SBC call leg for this SIP entity. ▪ [0] Don't enforce = (Default) Device forwards the MKI size as is. ▪ [1] Enforce = Device changes the MKI length according to the settings of the IP Profile parameter, MKISize.
Web: Remote Early Media RTP Behavior CLI: sbc-rmt-early-media-rtp [IpProfile_SBCRemoteEarlyMediaRTP]	Defines whether the destination UA sends RTP immediately after it sends a 18x response. ▪ [0] Immediate = (Default) Remote client sends RTP immediately after it sends 18x response with early media. Device forwards 18x and RTP as is. ▪ [1] Delayed = After sending 18x response, the remote client waits before sending RTP (e.g., Microsoft Lync environment). For the device's handling of this remote UA support, see Interworking SIP Early Media on page 522.
Web: Remote RFC 3960 Gateway Model Support CLI: sbc-rmt-rfc3960-supp [IpProfile_SBCRemoteSupportsRFC3960]	Defines whether the destination UA is capable of receiving 18x messages with delayed RTP. ▪ [0] Not Supported = (Default) UA does not support receipt of 18x messages with delayed RTP. For the device's handling of this remote UA support, see Interworking SIP Early Media on page 522. ▪ [1] Supported = UA is capable of receiving 18x messages with delayed RTP.
Web: Remote Can Play Ringback CLI: sbc-rmt-can-play-ringback [IpProfile_SBCRemoteCanPlayRingback]	Defines whether the destination UA can play a local ringback tone. ▪ [0] No = UA does not support local ringback tone. The device sends 18x with delayed SDP to the UA. ▪ [1] Yes = (Default) UA supports local ringback tone. For the device's handling of this remote UA support, see Interworking SIP Early Media on page 522.
Web: RFC 2833 DTMF Payload Type CLI: sbc-2833dtmf-payload [IpProfile_SBC2833DTMFPayloadType]	Defines the payload type of DTMF digits for this SIP entity. This enables the interworking of the DTMF payload type for RFC 2833 between different SBC call legs. For example, if two entities require different DTMF payload types, the SDP offer received by the device from one entity is forwarded to the destination entity with its payload type replaced with the configured payload type, and vice versa. The value range is 0 to 200. The default is 0 (i.e., the device forwards the received payload type as is).
Web: User Registration Time CLI: sbc-usr-reg-time [IpProfile_SBCUserRegistrationTime]	Defines the duration (in seconds) of the periodic registrations that occur between the users of this SIP entity and the device (the device responds with this value to the user in the Expires header).

Parameter	Description
	The valid range is 0 to 2,000,000 seconds. The default is 0. When set to 0, the device does not change the Expires header's value received in the user's REGISTER request. If no Expires header is received in the REGISTER message and this parameter is set to 0, the Expires header's value is set to 180 seconds, by default. **Note:** The corresponding global parameter is SBCUserRegistrationTime.
Web: Reliable Held Tone Source CLI: reliable-heldtone-source [IPProfile_ReliableHoldToneSource]	Enables the device to consider the received call-hold request (re-INVITE/UPDATE) with SDP containing 'a=sendonly', as genuine. ▪ [0] No (default) = Even if the received SDP contains 'a=sendonly', the device plays a held tone to the held party. This is useful in cases where the initiator of the call hold does not support the generation of held tones. ▪ [1] Yes = If the received SDP contains 'a=sendonly', the device does not play a held tone to the held party (and assumes that the initiator of the call hold plays the held tone). **Note:** The device plays a held tone only if the 'SBC Play Held Tone' parameter is set to Yes.
Web: Play Held Tone CLI: play-held-tone [IpProfile_SBCPlayHeldTone]	Enables the device to play a held tone to the held party. This is useful if the held party does not support playing a local held tone, or for IP entities initiating call hold that do not support the generation of held tones. ▪ [0] No (default) ▪ [1] Yes **Note:** If this parameter is set to Yes, the device plays the tone only if the 'SBC Remote Hold Format' parameter is set to transparent, send-only, send only 0.0.0.0, or not supported.
Web: Remote Hold Format CLI: remote-hold-Format [IPProfile_SBCRemoteHoldFormat]	Defines the format of the SDP in the re-INVITE for call hold that the device sends to the held party. ▪ [0] Transparent = Device forwards SDP as is. ▪ [1] Send Only = Device sends SDP with 'a=sendonly'. ▪ [2] Send Only Zero ip = Device sends SDP with 'a=sendonly' and 'c=0.0.0.0'. ▪ [3] Inactive = Device sends SDP with 'a=inactive'. ▪ [4] Inactive Zero ip = Device sends SDP with 'a=inactive' and 'c=0.0.0.0'. ▪ [5] Not Supported = Used when remote side cannot identify a call-hold message. The device terminates the received call-hold message (re-INVITE / UPDATE) and sends a 200 OK to the initiator of the call hold. The device plays a held tone to the held party if the 'SBC Play Held Tone' parameter is set to Yes.
Web: Remote Replaces Behavior CLI: sbc-rmt-replaces-behavior [IpProfile_SBCRemoteReplacesBehavior]	Enables the device to handle an incoming INVITE with the Replaces header for this SIP entity (which does not support this header). The Replaces header is used to replace an existing SIP dialog with a new dialog such as in call transfer or call pickup

Parameter	Description
	▪ [0] Transparent = (Default) SIP entity supports the Replaces header. The device forwards the received INVITE with the Replaces header as is to the SIP entity. ▪ [1] Handle Locally = The SIP entity does not support INVITE with the Replaces header. The device terminates the received INVITE with the Replaces header and establishes a new call between the SIP entity and the new call party. It then disconnects the call with the initial call party, by sending it a SIP BYE request. For example, assume that the device establishes a call between A and B. If B initiates a call transfer to C, the device receives an INVITE with the Replaces header from C. If A supports the Replaces header, the device simply forwards the INVITE as is to A; a new call is established between A and C and the call between A and B is disconnected. However, if A does not support the Replaces header, the device uses this feature to terminate the INVITE with Replaces header and handles the transfer for A. The device does this by connecting A to C, and disconnecting the call between A and B, by sending a SIP BYE request to B. Note that if media transcoding is required, the device sends an INVITE to C on behalf of A with a new SDP offer.
Web: SDP Ptime Answer CLI: sbc-sdp-ptime-ans [IpProfile_SBCSDPPtimeAnswer]	Defines the packetization time (ptime) of the coder in RTP packets for this SIP entity. This is useful when implementing transrating. ▪ [0] Remote Answer (Default) = Use ptime according to SDP answer. ▪ [1] Original Offer = Use ptime according to SDP offer. ▪ [2] Preferred Value= Use preferred ptime for negotiation, if configured by the 'Preferred Ptime' parameter.
Web: Preferred Ptime CLI: sbc-preferred-ptime [IpProfile_SBCPreferredPTime]	Defines the packetization time (in msec) for this SIP entity if the 'SBC SDP Ptime Answer' parameter is set to Preferred Value. The valid range is 0 to 200. The default is 0 (i.e., preferred ptime is not used).
Web: Use Silence Suppression CLI: sbc-use-silence-supp [IpProfile_SBCUseSilenceSupp]	Defines silence suppression support for this SIP entity. ▪ [0] Transparent (default) = Forward as is. ▪ [1] Add = Enable silence suppression for each relevant coder listed in the SDP. ▪ [2] Remove = Disable silence suppression for each relevant coder listed in the SDP.

Parameter	Description
Web: Play RBT To Transferee CLI: sbc-play-rbt-to-xferee [IpProfile_SBCPlayRBTToTransferee]	Enables the device to play a ringback tone to the transferred party (transferee) during a blind call transfer, for this SIP entity (which does not support such a tone generation during call transfer). The ringback tone indicates to the transferee of the ringing of the transfer target (to where the transferee is being transferred). ▪ [0] No (Default) ▪ [1] Yes Typically, the transferee hears a ringback tone only if the transfer target sends it early media. However, if the transferee is put on-hold before being transferred, no ringback tone is heard. When this feature is enabled, the device generates a ringback tone to the transferee during call transfer in the following scenarios: ▪ Transfer target sends a SIP 180 (Ringing) to the device. ▪ For non-blind transfer, if the call is transferred while the transfer target is ringing and no early media occurs. ▪ The 'Remote Early Media RTP Behavior parameter is set to Delayed (used in the Lync environment), and transfer target sends a 183 Session progress with SDP offer. If early media from the transfer target has already been detected, the transferee receives RTP stream from the transfer target. If it has not been detected, the device generates a ringback tone to the transferee and stops the tone generation once RTP has been detected from the transfer target. For any of these scenarios, if the transferee is put on-hold by the transferor, the device retrieves the transferee from hold, sends a re-INVITE if necessary, and then plays the ringback tone. **Note:** For the device to play the ringback tone, it must be loaded with a Prerecorded Tones (PRT) file. For more information, see Prerecorded Tones File on page 604.
Web: RTCP Mode CLI: sbc-rtcp-mode [IPProfile_SBCRTCPMode]	Defines how the device handles RTCP packets during call sessions for this SIP entity. This is useful for interworking RTCP between SIP entities. For example, this may be necessary when incoming RTCP is not compatible with the destination SIP entity's (this IP Profile) RTCP support. In such a scenario, the device can generate the RTCP and send it to the SIP entity. ▪ [0] Transparent (default) = RTCP is forwarded as is (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).

Parameter	Description
	Note: The corresponding global parameter is SBCRTCPMode.
Web: Jitter Compensation CLI: sbc-jitter-compensation [IpProfile_SBCJitterCompensation]	Enables the on-demand jitter buffer for SBC calls. This jitter buffer can be used when other functionality such as voice transcoding are not done on the call. This jitter buffer is useful when incoming packets are received at inconsistent intervals (i.e., packet delay variation). The jitter buffer stores the packets and sends them out at a constant rate (according to the coder's settings). ▪ [0] Disable (default) ▪ [1] Enable **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
Web: Remote Renegotiate on Fax Detection CLI: 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 (to which the IP Profile is assigned). 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 Group ID'. ▪ **[0]** Don't Care = (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 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 this parameter is set to [1] or [2], while the peer SIP entity must be assigned an IP Profile where this parameter is set to [2]. ▪ This feature is supported only if at least one of the SIP entities uses the G.711 coder. ▪ This feature utilizes DSP resources. If there are insufficient resources, the fax transaction fails.

Part V

Gateway Application

24 Introduction

This section describes configuration of the Gateway application. The Gateway application refers to IP-to-Tel (PSTN for digital interfaces) call routing and vice versa.

Notes:

- The IP-to-IP application has been superseded by the SBC application.
- In some areas of the Web interface, the term "GW" refers to the Gateway application.
- The terms *IP-to-Tel* and *Tel-to-IP* refer to the direction of the call relative to the device. IP-to-Tel refers to calls received from the IP network and destined to the PSTN/PBX (i.e., telephone connected directly or indirectly to the device); Tel-to-IP refers to calls received from telephones connected directly to the device's FXS ports or from the PSTN/PBX, and destined for the IP network.
- FXO (Foreign Exchange Office) is the interface replacing the analog telephone and connects to a Public Switched Telephone Network (PSTN) line from the Central Office (CO) or to a Private Branch Exchange (PBX). The FXO is designed to receive line voltage and ringing current, supplied from the CO or the PBX (just like an analog telephone). An FXO VoIP device interfaces between the CO/PBX line and the Internet.
- FXS (Foreign Exchange Station) is the interface replacing the Exchange (i.e., the CO or the PBX) and connects to analog telephones, dial-up modems, and fax machines. The FXS is designed to supply line voltage and ringing current to these telephone devices. An FXS VoIP device interfaces between the analog telephone devices and the Internet.

This page is intentionally left blank.

25 Digital PSTN

This section describes the configuration of the public switched telephone network (PSTN) related parameters.

25.1 Configuring Trunk Settings

The Trunk Settings page allows you to configure the device's trunks. This includes selecting the PSTN protocol and configuring related parameters.

This page also enables the following maintenance procedures:

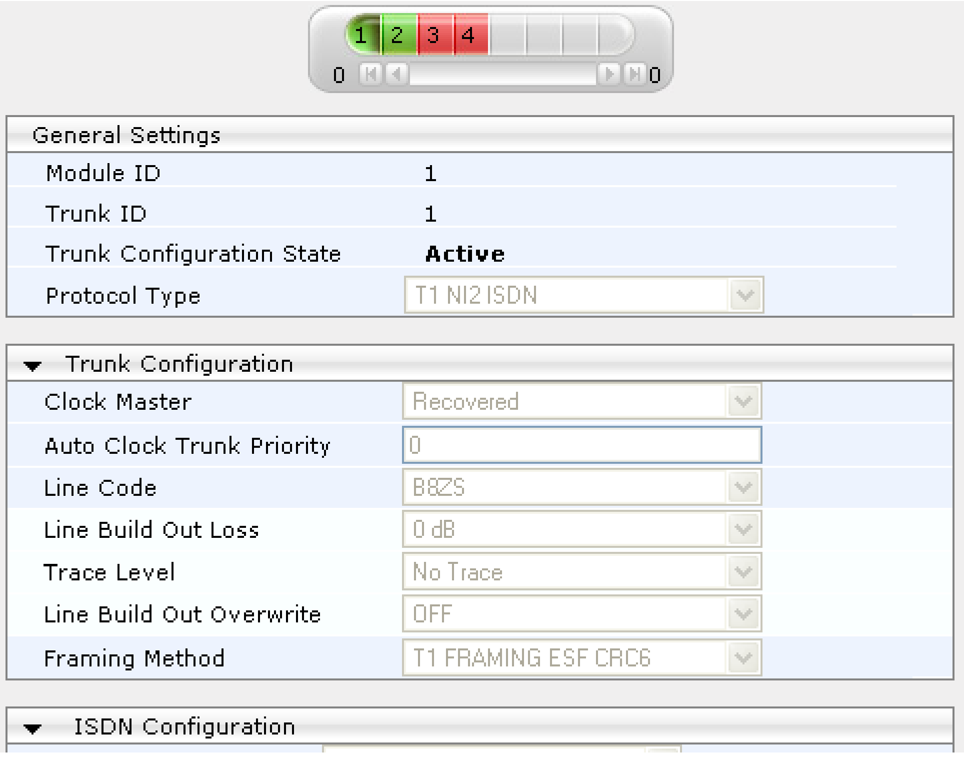
- **Taking a Trunk Out of Service:** Some parameters can be configured when the trunk is in service, while others require you to take the trunk out of service. This is done by clicking the **Stop** ■ button. Once you have "stopped" a trunk, all current calls are dropped and no new calls can be made on the trunk.
- **Deactivating a Trunk:** You can deactivate a trunk for maintenance. This is done by clicking the **Deactivate** Deactivate button. Deactivation temporarily disconnects (logically) the trunk from the PSTN network. Upon trunk deactivation, the device generates an AIS alarm on the trunk to the far-end. As a result, an RAI alarm signal may be received by the device. A subsequent trunk activation, done by clicking the **Activate** Activate button, reconnects the trunk to the PSTN network and clears the AIS alarm. Trunk deactivation is typically used for maintenance such as checking the trunk's physical integrity.
- **Creating a Loopback Line:** You can create (and remove) remote loopback for DS1 lines. This is done by clicking the **Create Loopback** Create Loopback button. To remove the loopback, click the **Remove Loopback** Remove Loopback button.

Notes:

- To delete a configured trunk, set the 'Protocol Type' parameter to **NONE**.
- For a description of the trunk parameters, see "PSTN Parameters" on page 907.
- During trunk deactivation, you cannot configure trunks.
- You cannot activate or deactivate a stopped trunk.
- If the trunk can't be stopped because it provides the device's clock (assuming the device is synchronized with the E1/T1 or BRI clock), assign a different trunk to provide the device's clock or enable 'TDM Bus PSTN Auto Clock' in the TDM Bus Settings page (see "TDM and Timing" on page 350).
- If the 'Protocol Type' parameter is set to NONE (i.e., no protocol type is selected) and no other trunks have been configured, after selecting a PRI protocol type you must reset the device.
- The displayed parameters depend on the protocol selected.
- BRI trunks can operate with E1 or T1 trunks.
- All PRI trunks of the device must be of the same line type (either E1 or T1). However, different variants of the same line type can be configured on different trunks. For example, E1 Euro ISDN and E1 CAS (subject to the constraints in the device's Release Notes).
- If the protocol type is CAS, you can assign or modify a dial plan (in the 'Dial Plan' field) and perform this without stopping the trunk.

➢ To configure trunks:

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



General Settings

Module ID	1
Trunk ID	1
Trunk Configuration State	**Active**
Protocol Type	T1 NI2 ISDN

Trunk Configuration

Clock Master	Recovered
Auto Clock Trunk Priority	0
Line Code	B8ZS
Line Build Out Loss	0 dB
Trace Level	No Trace
Line Build Out Overwrite	OFF
Framing Method	T1 FRAMING ESF CRC6

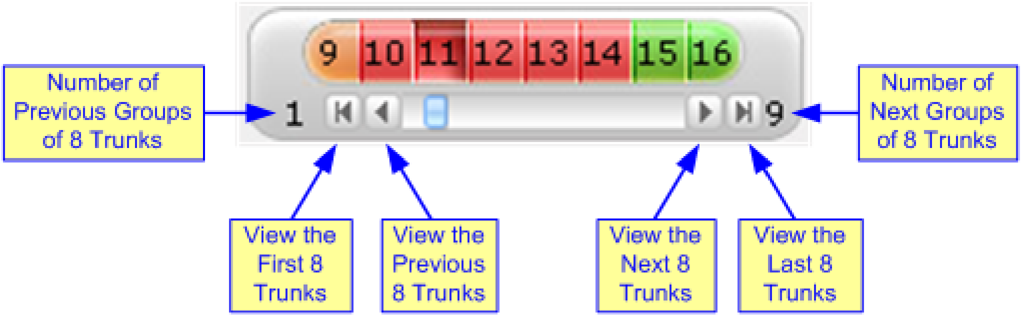
ISDN Configuration

On the top of the page, a bar with Trunk number icons displays the status of each trunk, according to the following color codes:

- **Grey:** Disabled
- **Green:** Active
- **Yellow:** RAI alarm (also appears when you deactivate a Trunk by clicking the **Deactivate** button)
- **Red:** LOS/LOF alarm
- **Blue:** AIS alarm
- **Orange:** D-channel alarm (ISDN only)

2. Select the trunk that you want to configure by clicking the desired Trunk number icon. The bar initially displays the first eight trunk number icons (i.e., trunks 1 through 8). To scroll through the trunk number icons (i.e., view the next/last or previous/first group of eight trunks), see the figure below:

Figure 25-1: Trunk Scroll Bar (Used Only as an Example)



9
10
11
12
13
14
15
16
1
9
Number of Previous Groups of 8 Trunks
Number of Next Groups of 8 Trunks
View the First 8 Trunks
View the Previous 8 Trunks
View the Next 8 Trunks
View the Last 8 Trunks

> **Note:** If the Trunk scroll bar displays all available trunks, the scroll bar buttons are unavailable.

After you have selected a trunk, the following is displayed:

- The read-only 'Module ID' field displays the module number to which the trunk belongs.
- The read-only 'Trunk ID' field displays the selected trunk number.
- The read-only ‘Trunk Configuration State’ displays the state of the trunk ('Active' or 'Inactive').
- The displayed parameters pertain to the selected trunk only.

3. Click the **Stop Trunk** ■ button (located at the bottom of the page) to take the trunk out of service so that you can configure the currently grayed out (unavailable) parameters. (Skip this step if you want to configure parameters that are available when the trunk is active). The stopped trunk is indicated by the following:
   - The ‘Trunk Configuration State’ field displays ‘Inactive’.
   - The **Stop Trunk** button is replaced by the **Apply Trunk Settings** ✓ button.

     When all trunks are stopped, the **Apply to All Trunks** ✓ button also appears.
   - All the parameters are available and can be modified.
4. Configure the trunk parameters as required.
5. Click the **Apply Trunk Settings** button to apply the changes to the selected trunk (or click **Apply to All Trunks** to apply the changes to all trunks); the **Stop Trunk** button replaces **Apply Trunk Settings** and the ‘Trunk Configuration State’ displays 'Active'.
6. To save the changes to flash memory, see "Saving Configuration" on page 592.
7. To reset the device, see "Resetting the Device" on page 589.

25.2 TDM and Timing

This section describes the configuration of the TDM and clock timing parameters.

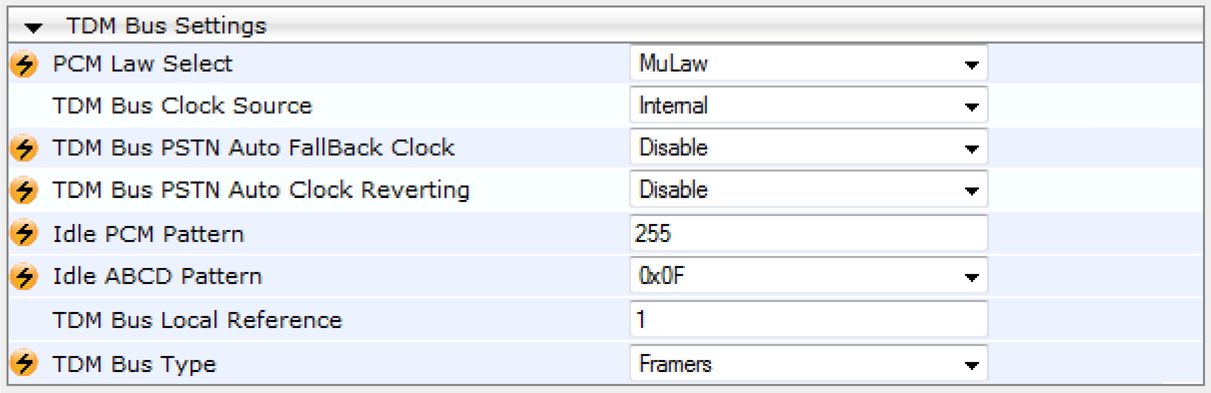
25.2.1 Configuring TDM Bus Settings

The TDM page allows you to configure the device's Time-Division Multiplexing (TDM) bus settings. For a description of these parameters, see "PSTN Parameters" on page 907.

➢ **To configure the TDM Bus settings:**

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

Figure 25-2: TDM Bus Settings Page



▼ TDM Bus Settings

⚡ PCM Law Select	MuLaw
TDM Bus Clock Source	Internal
⚡ TDM Bus PSTN Auto FallBack Clock	Disable
⚡ TDM Bus PSTN Auto Clock Reverting	Disable
⚡ Idle PCM Pattern	255
⚡ Idle ABCD Pattern	0x0F
TDM Bus Local Reference	1
⚡ TDM Bus Type	Framers

2. Configure the parameters as required.
3. Click **Submit**.
4. Save the changes to flash memory, see "Saving Configuration" on page 592.

25.2.2 Clock Settings

In a traditional TDM service network such as PSTN, both ends of the TDM connection must be synchronized. If synchronization is not achieved, voice frames are either dropped (to prevent a buffer overflow condition) or inserted (to prevent an underflow condition). In both cases, connection quality and reliability is affected.

- PSTN line clock (see "Recovering Clock from PSTN Line" on page 351)
- Internal clock (see "Configuring Internal Clock as Clock Source" on page 351)

> **Note:** When the device is used in a ‘non-span’ configuration, the internal device clock must be used (as explained above).

25.2.2.1 Recovering Clock from PSTN Line Interface

This section provides a brief description for configuring synchronization based on recovering clock from the PSTN line interface. For a full description of the clock parameters, see "PSTN Parameters" on page 907.

➢ **To configure synchronization based on clock from PSTN line:**

1. In the TDM Bus Settings page, do the following:
   a. Set the 'TDM Bus Clock Source' parameter (TDMBusClockSource) to **Network** to recover the clock from the line interface.
   b. Select the trunk from which the clock is derived, using the 'TDM Bus Local Reference' parameter (TDMBusLocalReference).

⚠ **Note:** The E1/T1 trunk should recover the clock from the remote side (see below description of the 'Clock Master' parameter). The BRI trunk should be configured as an ISDN user-side.

   c. Enable automatic switchover to the next available "slave" trunk if the device detects that the local-reference trunk is no longer capable of supplying the clock to the system:
      a. Set the 'TDM Bus PSTN Auto FallBack Clock' parameter (TDMBusPSTNAutoClockEnable) to **Enable**.
      b. Enable the device to switch back to a previous trunk that returns to service if it has higher switchover priority, using the 'TDM Bus PSTN Auto Clock Reverting' parameter (TDMBusPSTNAutoClockRevertingEnable).
      c. In the Trunk Settings page, configure the priority level of the trunk for taking over as a local-reference trunk, using the 'Auto Clock Trunk Priority' parameter (AutoClockTrunkPriority). A value of 100 means that it never uses the trunk as local reference.
2. For E1/T1 trunks only: Set the PSTN trunk to recover/derive clock from/to the remote side of the PSTN trunk (i.e. clock slave or clock master): In the Trunk Settings page, set the 'Clock Master' parameter (ClockMaster) to one of the following:
   - Recovered - to recover clock (i.e. slave)
   - Generated - to transmit clock (i.e. master)

25.2.2.2 Configuring Internal Clock as Clock Source

This section describes how to configure the device to use its internal clock source. The internal clock source is a stratum 4E-compliant clock source. When the device has no line interfaces, the device should be configured in this mode.

➢ **To configure internal clock as clock source:**

1. Set the clock source to be from the device's internal oscillator. In the TDM Bus Settings page, set the 'TDM Bus Clock Source' parameter (TDMBusClockSource) to **Internal**.
2. For E1/T1 trunks only: Set the line to drive the clock on all trunks: In the Trunk Settings page, set the 'Clock Master' parameter (ClockMaster) to Generated (for all trunks).

25.3 Configuring CAS State Machines

The CAS State Machine page allows you to modify various timers and other basic parameters to define the initialization of the CAS state machine without changing the state machine itself (no compilation is required). The change doesn't affect the state machine itself, but rather the configuration.

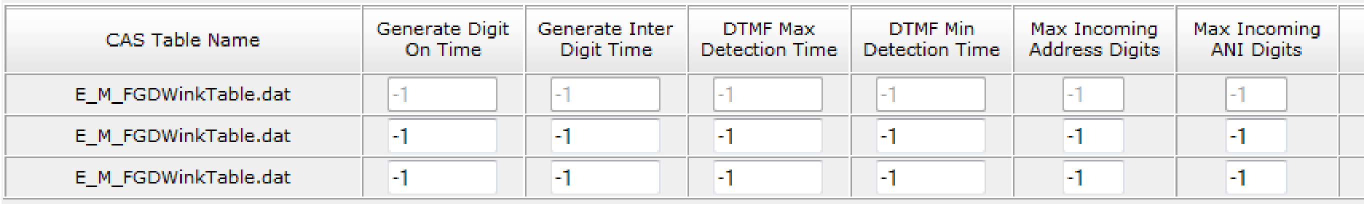
The CAS table used can be chosen in two ways (using the parameter CasChannelIndex):

- Single CAS table per trunk
- Different CAS table per group of B-channels in a trunk

➢ **To modify the CAS state machine parameters:**

1. Open the CAS State Machine page (**Configuration** tab > **VoIP** menu > **PSTN** > **CAS State Machines**).

Figure 25-3: CAS State Machine Page



CAS Table Name	Generate Digit On Time	Generate Inter Digit Time	DTMF Max Detection Time	DTMF Min Detection Time	Max Incoming Address Digits	Max Incoming ANI Digits
E_M_FGDWinkTable.dat	-1	-1	-1	-1	-1	-1
E_M_FGDWinkTable.dat	-1	-1	-1	-1	-1	-1
E_M_FGDWinkTable.dat	-1	-1	-1	-1	-1	-1

2. Ensure that the trunk is inactive. The trunk number displayed in the 'Related Trunks' field must be green. If it is red, indicating that the trunk is active, click the trunk number to open the Trunk Settings page (see "Configuring Trunk Settings" on page 347), select the required Trunk number icon, and then click **Stop Trunk**.
3. In the CAS State Machine page, modify the required parameters according to the table below.
4. Once you have completed the configuration, activate the trunk if required in the Trunk Settings page, by clicking the trunk number in the 'Related Trunks' field, and in the Trunk Settings page, select the required Trunk number icon, and then click **Apply Trunk Settings**.
5. Click **Submit**, and then reset the device (see "Resetting the Device" on page 589).

Notes:

- The CAS state machine can only be configured using the Web-based management tool.
- Don't modify the default values unless you fully understand the implications of the changes and know the default values. Every change affects the configuration of the state machine parameters and the call process related to the trunk you are using with this state machine.
- You can modify CAS state machine parameters only if the following conditions are met:
  1) Trunks are inactive (stopped), i.e., the 'Related Trunks' field displays the trunk number in green.
  2) State machine is not in use or is in reset, or when it is not related to any trunk. If it is related to a trunk, you must delete the trunk or de-activate (*Stop*) the trunk.
- Field values displaying '-1' indicate CAS default values. In other words, CAS state machine values are used.
- The modification of the CAS state machine occurs at the CAS application initialization only for non-default values (-1).
- For more information on the CAS Protocol table, refer to the *CAS Protocol Table Configuration Note*.

Table 25-1: CAS State Machine Parameters Description

Parameter	Description
Generate Digit On Time **[CasStateMachineGenerateDigitOnTime]**	Generates digit on-time (in msec). The value must be a positive value. The default is -1 (use value from CAS state machine).
Generate Inter Digit Time **[CasStateMachineGenerateInterDigitTime]**	Generates digit off-time (in msec). The value must be a positive value. The default is -1 (use value from CAS state machine).
DTMF Max Detection Time **[CasStateMachineDTMFMaxOnDetectionTime]**	Detects digit maximum on time (according to DSP detection information event) in msec units. The value must be a positive value. The default is -1 (use value from CAS state machine).
DTMF Min Detection Time **[CasStateMachineDTMFMinOnDetectionTime]**	Detects digit minimum on time (according to DSP detection information event) in msec units. The digit time length must be longer than this value to receive a detection. Any number may be used, but the value must be less than CasStateMachineDTMFMaxOnDetectionTime. The value must be a positive value. The default is -1 (use value from CAS state machine).
MAX Incoming Address Digits **[CasStateMachineMaxNumOfIncomingAddressDigits]**	Defines the limitation for the maximum address digits that need to be collected. After reaching this number of digits, the collection of address digits is stopped. The value must be an integer. The default is -1 (use value from CAS state machine).
MAX Incoming ANI Digits **[CasStateMachineMaxNumOfIncomingANIDigits]**	Defines the limitation for the maximum ANI digits that need to be collected. After reaching this number of digits, the collection of ANI digits is stopped. The value must be an integer. The default is -1 (use value from CAS state machine).
Collet ANI **[CasStateMachineCollectANI]**	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can control the state machine to collect ANI or discard ANI. ▪ **[0]** No = Don't collect ANI. ▪ **[1]** Yes = Collect ANI. ▪ **[-1]** Default = Default value - use value from CAS state machine.
Digit Signaling System **[CasStateMachineDigitSignalingSystem]**	Defines which Signaling System to use in both directions (detection\generation). ▪ **[0]** DTMF = Uses DTMF signaling. ▪ **[1]** MF = Uses MF signaling (default). ▪ **[-1]** Default = Default value - use value from CAS state machine.

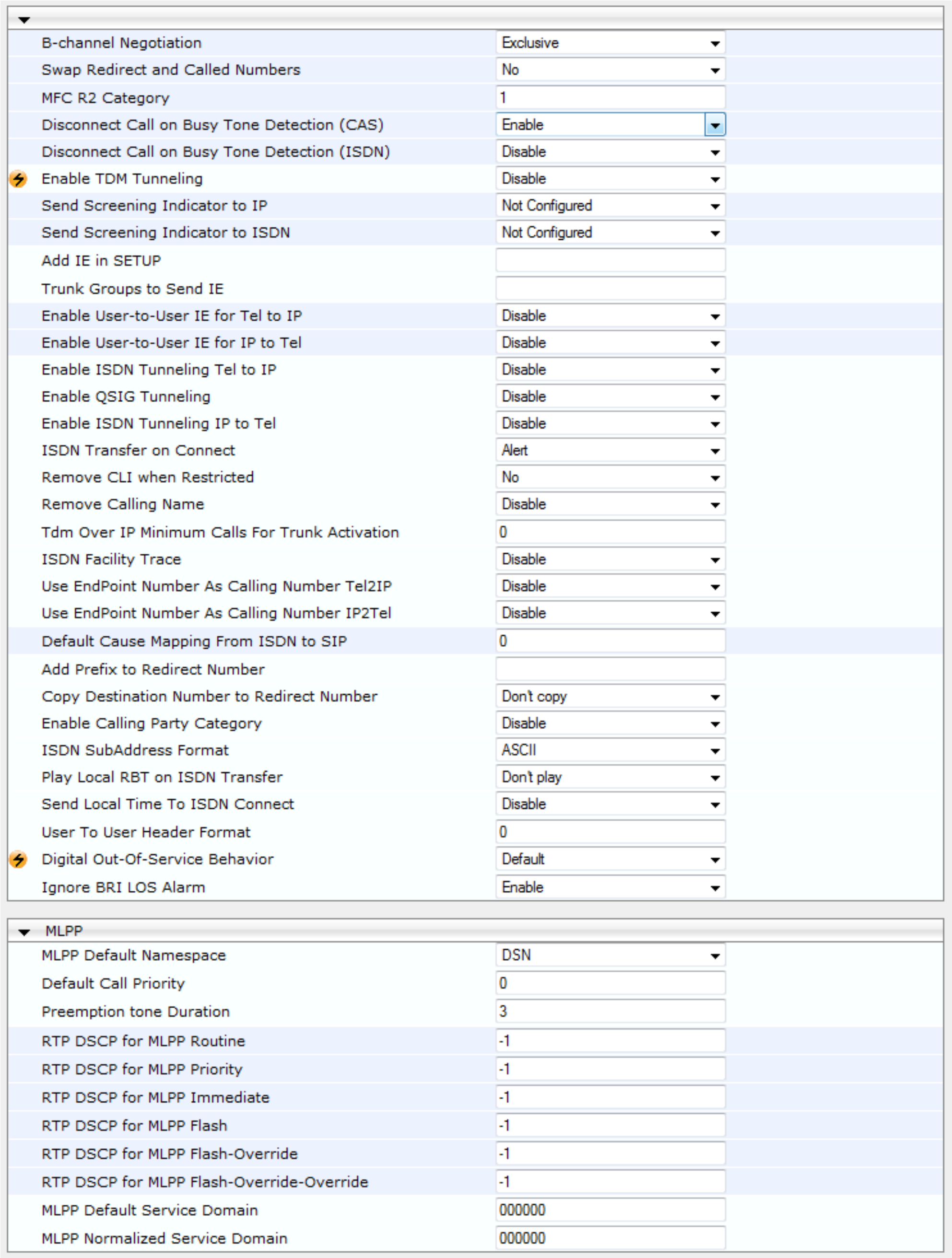
25.4 Configuring Digital Gateway Parameters

The Digital Gateway Parameters page allows you to configure miscellaneous digital parameters. For a description of these parameters, see "Configuration Parameters Reference" on page 763.

➢ To configure the digital gateway parameters:

1. Open the Digital Gateway Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Digital Gateway** > **Digital Gateway Parameters**).

Figure 25-4: Digital Gateway Parameters Page



B-channel Negotiation	Exclusive
Swap Redirect and Called Numbers	No
MFC R2 Category	1
Disconnect Call on Busy Tone Detection (CAS)	Enable
Disconnect Call on Busy Tone Detection (ISDN)	Disable
⚡ Enable TDM Tunneling	Disable
Send Screening Indicator to IP	Not Configured
Send Screening Indicator to ISDN	Not Configured
Add IE in SETUP	
Trunk Groups to Send IE	
Enable User-to-User IE for Tel to IP	Disable
Enable User-to-User IE for IP to Tel	Disable
Enable ISDN Tunneling Tel to IP	Disable
Enable QSIG Tunneling	Disable
Enable ISDN Tunneling IP to Tel	Disable
ISDN Transfer on Connect	Alert
Remove CLI when Restricted	No
Remove Calling Name	Disable
Tdm Over IP Minimum Calls For Trunk Activation	0
ISDN Facility Trace	Disable
Use EndPoint Number As Calling Number Tel2IP	Disable
Use EndPoint Number As Calling Number IP2Tel	Disable
Default Cause Mapping From ISDN to SIP	0
Add Prefix to Redirect Number	
Copy Destination Number to Redirect Number	Don't copy
Enable Calling Party Category	Disable
ISDN SubAddress Format	ASCII
Play Local RBT on ISDN Transfer	Don't play
Send Local Time To ISDN Connect	Disable
User To User Header Format	0
⚡ Digital Out-Of-Service Behavior	Default
Ignore BRI LOS Alarm	Enable

MLPP

MLPP Default Namespace	DSN
Default Call Priority	0
Preemption tone Duration	3
RTP DSCP for MLPP Routine	-1
RTP DSCP for MLPP Priority	-1
RTP DSCP for MLPP Immediate	-1
RTP DSCP for MLPP Flash	-1
RTP DSCP for MLPP Flash-Override	-1
RTP DSCP for MLPP Flash-Override-Override	-1
MLPP Default Service Domain	000000
MLPP Normalized Service Domain	000000

2. Configure the parameters as required.
3. Click Submi**t**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

25.5 Tunneling Applications

This section discusses the device's support for VoIP tunneling applications.

25.5.1 TDM Tunneling

The device's TDM Tunneling feature allows you to tunnel groups of digital trunk spans or timeslots (B-channels) over the IP network. TDM Tunneling utilizes the device's internal routing (without Proxy control) capabilities to receive voice and data streams from TDM spans or individual timeslots, convert them into packets, and then transmit them over the IP network (using point-to-point or point-to-multipoint device distributions). A device opposite it (or several devices when point-to-multipoint distribution is used) converts the IP packets back into TDM traffic. Each timeslot can be targeted to any other timeslot within a trunk in the opposite device.

Note: TDM tunneling is applicable only to PRI (not BRI).

When TDM Tunneling is enabled (the parameter EnableTDMoverIP is set to '1') on the originating device, the originating device automatically initiates SIP calls from all enabled B-channels belonging to the spans that are configured with the protocol type ‘Transparent’ (for ISDN trunks) or ‘Raw CAS’ (for CAS trunks). The called number of each call is the internal phone number of the B-channel from where the call originates. The Inbound IP Routing table is used to define the destination IP address of the terminating device. The terminating device automatically answers these calls if the protocol type is set to ‘Transparent’ (ProtocolType = 5) or ‘Raw CAS’ (ProtocolType = 3 for T1 and 9 for E1) and the parameter ChannelSelectMode is set to 0 (By Dest Phone Number).

Note: It's possible to configure both devices to also operate in symmetric mode. To do so, set EnableTDMOverIP to 1 and configure the Outbound IP Routing table in both devices. In this mode, each device (after it's reset) initiates calls to the second device. The first call for each B-channel is answered by the second device.

The device continuously monitors the established connections. If for some reason, one or more calls are released, the device automatically re-establishes these ‘broken’ connections. When a failure in a physical trunk or in the IP network occurs, the device re-establishes the tunneling connections when the network is restored.

Note: It's recommended to use the keep-alive mechanism for each connection, by activating the ‘session expires’ timeout and using Re-INVITE messages.

The device supports the configuration (TDMoIPInitiateInviteTime and TDMoIPInviteRetryTime parameters) of the following timers for the TDM-over-IP tunneling application:

- Time between successive INVITEs sent from the same trunk.
- Time between call release and the new INVITE that is sent on the same channel. The call can be released if the device receives a 4xx or 5xx response.

By utilizing the ‘Profiles’ mechanism (see "Coders and Profiles" on page 307), you can configure the TDM Tunneling feature to choose different settings based on a timeslot or groups of timeslots. For example, you can use low-bit-rate vocoders to transport voice and ‘Transparent’ coder to transport data (e.g., for D-channel). You can also use Profiles to assign ToS (for DiffServ) per source - a timeslot carrying data or signaling is assigned a higher priority value than a timeslot carrying voice.

For tunneling CAS trunks, set the protocol type to 'Raw CAS' (ProtocolType = 3 / 9) and enable RFC 2833 CAS relay mode ('CAS Transport Type' parameter is set to 'CAS RFC2833 Relay').

Note: For TDM over IP, the parameter CallerIDTransportType must be set to '0' (disabled), i.e., transparent.

Below is an example of *ini* files for two devices implementing TDM Tunneling for four E1 spans. Note that in this example both devices are dedicated to TDM tunneling.

Terminating Side:

```
EnableTDMOverIP = 1
;E1_TRANSPARENT_31
ProtocolType_0 = 5
ProtocolType_1 = 5
ProtocolType_2 = 5
ProtocolType_3 = 5
[PREFIX]
PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix,
PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_DestPort,
PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID,
PREFIX_SrcHostPrefix, PREFIX_TransportType,
PREFIX_SrcTrunkGroupID, PREFIX_DestSRD, PREFIX_CostGroup,
PREFIX_ForkingGroup;
Prefix 1 = *,10.8.24.12;
[\PREFIX]

;IP address of the device in the opposite
;location
;Channel selection by Phone number.
ChannelSelectMode = 0
;Profiles can be used do define different coders per B-channels
;such as Transparent
;coder for B-channels (timeslot 16) that carries PRI ;signaling.
[TrunkGroup]
FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum,
TrunkGroup_FirstTrunkId, TrunkGroup_LastTrunkId,
TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel,
TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId,
TrunkGroup_Module;
TrunkGroup 1 = 0,0,0,1,31,1000,1;
TrunkGroup 1 = 0,1,1,1,31,2000,1;
TrunkGroup 1 = 0,2,2,1,31,3000,1;
TrunkGroup 1 = 0,3,3,1,31,4000,1;
TrunkGroup 1 = 0,0,0,16,16,7000,2;
TrunkGroup 1 = 0,1,1,16,16,7001,2;
TrunkGroup 1 = 0,2,2,16,16,7002,2;
TrunkGroup 1 = 0,3,3,16,16,7003,2;
[/TrunkGroup]
```

```
[ CodersGroup0 ]
FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0 0 = g7231;
CodersGroup0 1 = Transparent;
[ \CodersGroup0 ]
[TelProfile]
FORMAT TelProfile_Index = TelProfile_ProfileName,
TelProfile_TelPreference, TelProfile_CodersGroupID,
TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay,
TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ,
TelProfile_SigIPDiffServ, TelProfile_DtmfVolume,
TelProfile_InputGain, TelProfile_VoiceVolume,
TelProfile_EnableReversePolarity,
TelProfile_EnableCurrentDisconnect,
TelProfile_EnableDigitDelivery, TelProfile_EnableEC,
TelProfile_MWIAnalog, TelProfile_MWIDisplay,
TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia,
TelProfile_ProgressIndicator2IP;
TelProfile 1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$;
TelProfile 2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$;
[\TelProfile]
```

Originating Side:

```
;E1_TRANSPARENT_31
ProtocolType_0 = 5
ProtocolType_1 = 5
ProtocolType_2 = 5
ProtocolType_3 = 5
;Channel selection by Phone number.
ChannelSelectMode = 0
[TrunkGroup]
FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum,
TrunkGroup_FirstTrunkId, TrunkGroup_LastTrunkId,
TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel,
TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId,
TrunkGroup_Module;
TrunkGroup 0 = 0,0,0,1,31,1000,1;
TrunkGroup 0 = 0,1,1,1,31,2000,1;
TrunkGroup 0 = 0,2,2,1,31,3000,1;
TrunkGroup 0 = 0,3,1,31,4000,1;
TrunkGroup 0 = 0,0,0,16,16,7000,2;
TrunkGroup 0 = 0,1,1,16,16,7001,2;
TrunkGroup 0 = 0,2,2,16,16,7002,2;
TrunkGroup 0 = 0,3,3,16,16,7003,2;
[\TrunkGroup]
[ CodersGroup0 ]
FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce;
CodersGroup0 0 = g7231;
CodersGroup0 1 = Transparent;
[ \CodersGroup0 ]
[TelProfile]
FORMAT TelProfile_Index = TelProfile_ProfileName,
TelProfile_TelPreference, TelProfile_CodersGroupID,
TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay,
TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ,
TelProfile_SigIPDiffServ, TelProfile_DtmfVolume,
TelProfile_InputGain, TelProfile_VoiceVolume,
TelProfile_EnableReversePolarity,
TelProfile_EnableCurrentDisconnect,
TelProfile_EnableDigitDelivery, TelProfile_EnableEC,
TelProfile_MWIAnalog, TelProfile_MWIDisplay,
TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia,
TelProfile_ProgressIndicator2IP;
TelProfile_1 = voice,$$,1,$$,$$,$$,$$,$$,$$,$$
TelProfile_2 = data,$$,2,$$,$$,$$,$$,$$,$$,$$
[\TelProfile]
```

25.5.1.1 DSP Pattern Detector

For TDM tunneling applications, you can use the DSP pattern detector feature to initiate the echo canceller at call start. The device can be configured to support detection of a specific one-byte idle data pattern transmitted over digital E1/T1 timeslots. The device can be configured to detect up to four different one-byte data patterns. When the defined idle data pattern is detected, the channel resets its echo canceller.

➢ **To configure DSP pattern detector:**

1. In the IPMedia Settings page (**Configuration** tab > **VoIP** menu > **Media** > **IPMedia Settings**), do the following:
   a. Set the 'IPMedia Detectors' parameter (EnableDSPIPMDetectors) to **Enable**.
   b. Set the 'Enable Pattern Detector' parameter (EnablePatternDetector) to **Enable**.
2. Configure the number (e.g., 5) of consecutive patterns to trigger the pattern detection event, using the ini file parameter, PDThreshold.
3. Configure the patterns that can be detected by the Pattern Detector, using the ini file parameter, PDPattern. For example:

```
PDPattern = 84, 85, 212, 213 ; for idle patterns 54, 55, D4
and D5
```

25.5.2 QSIG Tunneling

The device supports QSIG tunneling over SIP, according to IETF Internet-Draft draft-elwell-sipping-qsig-tunnel-03 ("Tunnelling of QSIG over SIP") and ECMA-355/ISO/IEC 22535. This is applicable to all ISDN variants. QSIG tunneling can be applied to all calls or to specific calls using IP Profiles.

Note: TDM tunneling is applicable to PRI and BRI.

QSIG tunneling sends all QSIG messages as raw data in corresponding SIP messages using a dedicated message body. This is used, for example, to enable two QSIG subscribers connected to the same or different QSIG PBX to communicate with each other over an IP network. Tunneling is supported in both directions (Tel-to-IP and IP-to-Tel).

The term tunneling means that messages are transferred ‘as is’ to the remote side without being converted (QSIG > SIP > QSIG). The advantage of tunneling over QSIG-to-SIP interworking is that by using interworking, QSIG functionality can only be partially achieved. When tunneling is used, all QSIG capabilities are supported and the tunneling medium (the SIP network) does not need to process these messages.

QSIG messages are transferred in SIP messages in a separate Multipurpose Internet Mail Extensions (MIME) body. Therefore, if a message contains more than one body (e.g., SDP and QSIG), multipart MIME must be used. The Content-Type of the QSIG tunneled message is ‘application/QSIG’. The device also adds a Content-Disposition header in the following format:

```
Content-Disposition: signal; handling=required.
```

QSIG tunneling is done as follows:

- **Call setup (originating device):** The QSIG Setup request is encapsulated in the SIP INVITE message without being altered. After the SIP INVITE request is sent, the device does not encapsulate the subsequent QSIG message until a SIP 200 OK response is received. If the originating device receives a 4xx, 5xx, or 6xx response, it disconnects the QSIG call with a ‘no route to destination’ cause.
- **Call setup (terminating device):** After the terminating device receives a SIP INVITE request with a 'Content-Type: application/QSIG', it sends the encapsulated QSIG Setup message to the Tel side and sends a 200 OK response (no 1xx response is sent) to IP. The 200 OK response includes an encapsulated QSIG Call Proceeding message (without waiting for a Call Proceeding message from the Tel side). If tunneling is disabled and the incoming INVITE includes a QSIG body, a 415 response is sent.
- **Mid-call communication:** After the SIP connection is established, all QSIG messages are encapsulated in SIP INFO messages.
- **Call tear-down:** The SIP connection is terminated once the QSIG call is complete. The Release Complete message is encapsulated in the SIP BYE message that terminates the session.

➢ To enable QSIG tunneling:

1. In the Digital Gateway Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Digital Gateway** > **Digital Gateway Parameters**), set the 'Enable QSIG Tunneling' parameter (EnableQSIGTunneling) to **Enable** on the originating and terminating devices.
2. Configure the QSIGTunnelingMode parameter for defining the format of encapsulated QSIG message data in the SIP message MIME body (0 for ASCII presentation; 1 for binary encoding).
3. Set the ISDNDuplicateQ931BuffMode parameter to 128 to duplicate all messages.
4. Set the ISDNInCallsBehavior parameter to 4096.
5. Set the ISDNRxOverlap parameter to 0 for tunneling of QSIG overlap-dialed digits (see below for description).

The configuration of the ISDNInCallsBehavior and ISDNRxOverlap parameters allows tunneling of QSIG overlap-dialed digits (Tel to IP). In this configuration, the device **delays** the sending of the QSIG Setup Ack message upon receipt of the QSIG Setup message. Instead, the device sends the Setup Ack message to QSIG only when it receives the SIP INFO message with Setup Ack encapsulated in its MIME body. The PBX sends QSIG Information messages (to complete the Called Party Number) only after it receives the Setup Ack. The device relays these Information messages encapsulated in SIP INFO messages to the remote party.

25.6 ISDN Non-Facility Associated Signaling (NFAS)

In regular T1 ISDN trunks, a single 64 kbps channel carries signaling for the other 23 B-channels of that particular T1 trunk. This channel is called the D-channel and usually resides on timeslot # 24. ISDN Non-Facility Associated Signaling (NFAS) enables the use of a single D-channel to control multiple PRI interfaces.

⚠ **Note:** NFAS is applicable only to T1 trunks.

With NFAS it is possible to define a group of T1 trunks, called an *NFAS group*, in which a single D-channel carries ISDN signaling messages for the entire group. The NFAS group's B-channels are used to carry traffic such as voice or data. The NFAS mechanism also enables definition of a backup D-channel on a different T1 trunk, to be used if the primary D-channel fails.

The device supports up to 12 NFAS groups. Each group can comprise up to 10 T1 trunks and each group must contain different T1 trunks. Each T1 trunk is called an "NFAS member". The T1 trunk whose D-channel is used for signaling is called the "Primary NFAS Trunk". The T1 trunk whose D-channel is used for backup signaling is called the "Backup NFAS Trunk". The primary and backup trunks each carry 23 B-channels while all other NFAS trunks each carry 24 B-channels.

The NFAS group is identified by an NFAS GroupID number (possible values are 1 to 12). To assign a number of T1 trunks to the same NFAS group, use the NFASGroupNumber_x = groupID (where x is the physical trunk ID (0 to the maximum number of trunks) or the Web interface (see "Configuring Trunk Settings" on page 347).

The parameter DchConfig_x = Trunk_type defines the type of NFAS trunk. Trunk_type is set to 0 for the primary trunk, to 1 for the backup trunk, and to 2 for an ordinary NFAS trunk. 'x' denotes the physical trunk ID (0 to the maximum number of trunks). You can also use the Web interface (see "Configuring Trunk Settings" on page 347).

For example, to assign the first four T1 trunks to NFAS group #1, in which trunk #0 is the primary trunk and trunk #1 is the backup trunk, use the following configuration:

```
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0             ;Primary T1 trunk
DchConfig_1 = 1             ;Backup T1 trunk
DchConfig_2 = 2             ;24 B-channel NFAS trunk
DchConfig_3 = 2             ;24 B-channel NFAS trunk
```

The NFAS parameters are described in "PSTN Parameters" on page 907.

25.6.1 NFAS Interface ID

Several ISDN switches require an additional configuration parameter per T1 trunk that is called 'Interface Identifier'. In NFAS T1 trunks, the Interface Identifier is sent explicitly in Q.931 Setup / Channel Identification IE for all NFAS trunks, except for the B-channels of the Primary trunk (see note below).

The Interface ID can be defined per member (T1 trunk) of the NFAS group, and must be coordinated with the configuration of the Switch. The default value of the Interface ID is identical to the number of the physical T1 trunk (0 for the first trunk, 1 for the second T1 trunk, and so on, up to the maximum number of trunks).

To define an explicit Interface ID for a T1 trunk (that is different from the default), use the following parameters:

- ISDNIBehavior_x = 512 (x = 0 to the maximum number of trunks identifying the device's physical trunk)
- ISDNNFASInterfaceID_x = ID (x = 0 to 255)

Notes:

- Usually the Interface Identifier is included in the Q.931 Setup/Channel Identification IE only on T1 trunks that doesn’t contain the D-channel. Calls initiated on B-channels of the Primary T1 trunk, by default, don’t contain the Interface Identifier. Setting the parameter ISDNIBehavior_x to 2048’ forces the inclusion of the Channel Identifier parameter also for the Primary trunk.
- The parameter ISDNNFASInterfaceID_x = ID can define the ‘Interface ID’ for any Primary T1 trunk, even if the T1 trunk is not a part of an NFAS group. However, to include the Interface Identifier in Q.931 Setup/Channel Identification IE configure ISDNIBehavior_x = 2048 in the *ini* file.

25.6.2 Working with DMS-100 Switches

The DMS-100 switch requires the following NFAS Interface ID definitions:

- InterfaceID #0 for the Primary trunk
- InterfaceID #1 for the Backup trunk
- InterfaceID #2 for a 24 B-channel T1 trunk
- InterfaceID #3 for a 24 B-channel T1 trunk, and so on for subsequent T1 trunks

For example, if four T1 trunks on a device are configured as a single NFAS group with Primary and Backup T1 trunks that is used with a DMS-100 switch, the following parameters should be used:

```
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0    ;Primary T1 trunk
DchConfig_1 = 1    ;Backup T1 trunk
DchConfig_2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

If there is no NFAS Backup trunk, the following configuration should be used:

```
ISDNNFASInterfaceID_0 = 0
ISDNNFASInterfaceID_1 = 2
ISDNNFASInterfaceID_2 = 3
ISDNNFASInterfaceID_3 = 4
ISDNIBehavior = 512   ;This parameter should be added because of
;ISDNNFASInterfaceID coniguration above
NFASGroupNumber_0 = 1
NFASGroupNumber_1 = 1
NFASGroupNumber_2 = 1
NFASGroupNumber_3 = 1
DchConfig_0 = 0    ;Primary T1 trunk
DchConfig_1 = 2    ;B-Channel NFAS trunk
DchConfig_2 = 2    ;B-Channel NFAS trunk
DchConfig_3 = 2    ;B-channel NFAS trunk
```

25.6.3 Creating an NFAS-Related Trunk Configuration

The procedures for creating and deleting an NFAS group must be performed in the correct order, as described below.

➢ **To create an NFAS Group:**

1. If there's a backup ('secondary') trunk for this group, it must be configured first.
2. Configure the primary trunk before configuring any NFAS ('slave') trunk.
3. Configure NFAS ('slave') trunks.

➢ **To stop / delete an NFAS Group:**

1. Stop or delete (by setting ProtocolType to 0, i.e., 'None') all NFAS ('slave') trunks.
2. Stop or delete (by setting ProtocolType to 0, i.e., 'None') the backup trunk if a backup trunk exists.
3. Stop or delete (by setting ProtocolType to 0, i.e., 'None') the primary trunk.

Notes:

- All trunks in the group must be configured with the same values for trunk parameters TerminationSide, ProtocolType, FramingMethod, and LineCode.
- After stopping or deleting the backup trunk, delete the group and then reconfigure it.

25.6.4 Performing Manual D-Channel Switchover in NFAS Group

If an NFAS group is configured with two D-channels (Primary and Backup), you can do a manual switchover between these D-channels.

➢ **To manually switchover from active to standby D-channel:**

1. Open the NFAS Group & D-Channel Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **NFAS Group & D-Channel Status**).
2. Select the required NFAS group, and then click the **Switch Activity** button.

Notes:

- The **Switch Activity** button is unavailable (i.e, grayed out) if a switchover cannot be done due to, for example, alarms or unsuitable states.
- This feature is applicable only to T1 ISDN protocols supporting NFAS, and only if the NFAS group is configured with two D-channels.

25.7 ISDN Overlap Dialing

Overlap dialing is a dialing scheme used by several ISDN variants to send and/or receive called number digits one after the other (or several at a time). This is in contrast to en-bloc dialing in which a complete number is sent in one message.

The device supports the following ISDN overlap dialing methods:

- Collects ISDN called party number digits and then sends the SIP INVITE to the IP side with the complete destination number (see "Collecting ISDN Digits and Sending Complete Number in SIP" on page 363)
- Interworks ISDN overlap dialing with SIP, according to RFC 3578 (see "Interworking

ISDN Overlap Dialing with SIP According to RFC 3578" on page 364)

> **Note:** ISDN overlap dialing is applicable to PRI and BRI.

25.7.1 Collecting ISDN Digits and Sending Complete Number in SIP

The device can support an overlap dialing mode whereby the device collects the called party number digits from ISDN Q.931 Information messages or DTMF signals, and then sends a SIP INVITE message to the IP side containing the complete destination number.

ISDN overlap dialing for incoming ISDN calls can be configured for the entire device or per ISDN trunk. This is configured using the global, ISDNRxOverlap parameter or the ISDNRxOverlap_x parameter (where *x* denotes the trunk number), respectively.

By default (see the ISDNINCallsBehavior parameter), the device plays a dial tone to the ISDN user side when it receives an empty called number from the ISDN. In this scenario, the device includes the Progress Indicator in the SetupAck ISDN message that it sends to the ISDN side.

The device can also mute in-band DTMF detection until it receives the complete destination number from the ISDN. This is configured using the MuteDTMFInOverlap parameter. The Information digits can be sent in-band in the voice stream, or out-of-band using Q.931 Information messages. If Q.931 Information messages are used, the DTMF in-band detector must be disabled. Note that when at least one digit is received in the ISDN Setup message, the device stops playing a dial tone.

The device stops collecting digits (from the ISDN) upon the following scenarios:

- The device receives a Sending Complete IE in the ISDN Setup or Information messages, indicating no more digits.
- The timeout between received digits expires (configured by the TimeBetweenDigits parameter).
- The maximum number of received digits has been reached (configured by the MaxDigits parameter).
- A match is found with the defined digit map (configured by the DigitMapping parameter).

Relevant parameters (described in "PSTN Parameters" on page 907):

- ISDNRxOverlap_x = 1 (can be configured per trunk)
- TimeBetweenDigits
- MaxDigits
- MuteDTMFInOverlap
- DigitMapping

For configuring ISDN overlap dialing using the Web interface, see "Configuring Trunk Settings" on page 347.

25.7.2 Interworking ISDN Overlap Dialing with SIP According to RFC 3578

The device supports the interworking of ISDN overlap dialing to SIP and vice versa, according to RFC 3578.

- **Interworking ISDN overlap dialing to SIP (Tel to IP):** The device sends collected digits each time it receives them (initially from the ISDN Setup message and then from subsequent Q.931 Information messages) to the IP side, using subsequent SIP INVITE messages. You can also define the minimum number of overlap digits to collect before sending the first SIP message (INVITE) for routing the call, using the MinOverlapDigitsForRouting parameter.
- **Interworking SIP to ISDN overlap dialing (IP to Tel):** For each received SIP INVITE pertaining to the same dialog session, the device sends an ISDN Setup message (and subsequent Q.931 Information messages) with the collected digits to the Tel side. For all subsequent INVITEs received, the device sends a SIP 484 "Address Incomplete" response to the IP in order to maintain the current dialog session and to receive additional digits from subsequent INVITEs.

Relevant parameters (described in "PSTN Parameters" on page 907):

- ISDNRxOverlap = 2
- ISDNTxOverlap
- ISDNOutCallsBehavior = 2
- MinOverlapDigitsForRouting
- TimeBetweenDigits
- MaxDigits
- DigitMapping
- MuteDTMFInOverlap

For configuring ISDN overlap dialing using the Web interface, see "Configuring Trunk Settings" on page 347.

25.8 Redirect Number and Calling Name (Display)

The following tables define the device's redirect number and calling name (Display) support for various ISDN variants according to NT (Network Termination) / TE (Termination Equipment) interface direction:

Table 25-2: Calling Name (Display)

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	Yes	Yes	Yes
TE-to-NT	Yes	Yes	Yes	No	Yes

Table 25-3: Redirect Number

NT/TE Interface	DMS-100	NI-2	4/5ESS	Euro ISDN	QSIG
NT-to-TE	Yes	Yes	Yes	Yes	Yes
TE-to-NT	Yes	Yes	Yes	Yes*	Yes

***** When using ETSI DivertingLegInformation2 in a Facility IE (not Redirecting Number IE).

26 Trunk Group

This section describes the configuration of the device's channels, which includes assigning them to Trunk Groups.

26.1 Configuring Trunk Group Table

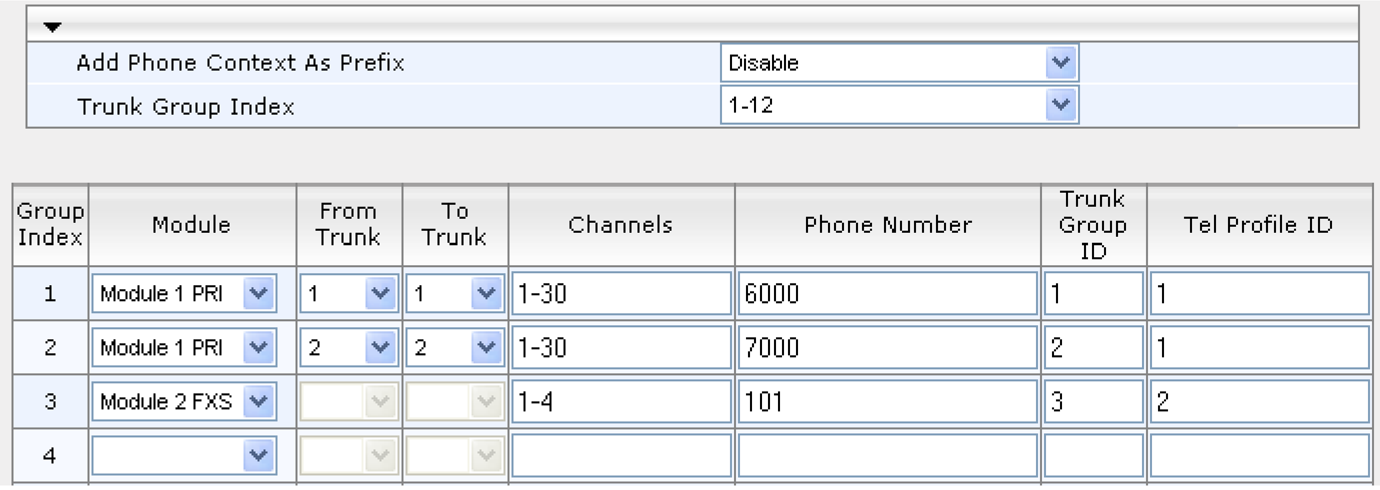
The Trunk Group table lets you configure up to 120 Trunk Groups. A Trunk Group is a logical group of physical trunks and channels. A Trunk Group can include multiple trunks and ranges of channels. To enable and activate the channels of the device, Trunk Groups need to be configured and assigned with telephone numbers. Channels that are not configured in this table are disabled.

Once you have configured your Trunk Groups, you need to use them for routing incoming IP calls to the Tel side, which is represented by a specific Trunk Group (ID). For configuring IP-to-Tel routing rules, see "Configuring Inbound IP Routing Table" on page 406. You can also use Trunk Groups for routing Tel calls to the IP side. For configuring Tel-to-IP routing rules, see "Configuring Outbound IP Routing Table" on page 397.

The following procedure describes how to configure Trunk Groups in the Web interface. You can also configure this using the table ini file parameter, TrunkGroup_x or CLI command, configure voip > gw hunt-or-trunk-group trunk-group.

➢ To configure a Trunk Group:

1. Open the Trunk Group Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Trunk Group** > **Trunk Group**).



Add Phone Context As Prefix	Disable
Trunk Group Index	1-12

Group Index	Module	From Trunk	To Trunk	Channels	Phone Number	Trunk Group ID	Tel Profile ID
1	Module 1 PRI	1	1	1-30	6000	1	1
2	Module 1 PRI	2	2	1-30	7000	2	1
3	Module 2 FXS			1-4	101	3	2
4							

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

You can also register all your Trunk Groups. The registration method per Trunk Group is configured by the 'Registration Mode' parameter in the Trunk Group Settings page (see "Configuring Hunt Group Settings" on page 367).

- To register the Trunk Groups, click the **Register** button, located below the Trunk Group table.
- To unregister the Trunk Groups, click the **Unregister** button, located below the Trunk Group table.

Table 26-1: Trunk Group Table Parameter Descriptions

Parameter	Description
Module CLI: module [TrunkGroup_Module]	Defines the telephony interface module for which you want to define the Trunk Group.
From Trunk CLI: first-trunk-id **[TrunkGroup_FirstTrunkId]**	Defines the starting physical Trunk number in the Trunk Group. The number of listed Trunks depends on the device's hardware configuration. **Note:** This parameter is applicable only to PRI and BRI modules.
To Trunk CLI: last-trunk-id **[TrunkGroup_LastTrunkId]**	Defines the ending physical Trunk number in the Trunk Group. The number of listed Trunks depends on the device's hardware configuration. **Note:** This parameter is applicable only to PRI and BRI modules.
Channels CLI: first-b-channel **[TrunkGroup_FirstBChannel]** CLI: last-b-channel **[TrunkGroup_LastBChannel]**	Defines the device's channels/ports (analog module) or Trunk B-channels (digital module). To enable channels, enter the channel numbers. You can enter a range of channels by using the syntax *n-m*, where *n* represents the lower channel number and *m* the higher channel number. For example, "1-4" specifies channels 1 through 4. To represent all the Trunk's B-channels, enter a single asterisk (*). **Note:** The number of defined channels must not exceed the maximum number of the Trunk's B-channels.
Phone Number CLI: first-phone-number **[TrunkGroup_FirstPhoneNumber]**	Defines the telephone number(s) of the channels. The valid value can be up to 50 characters. For a range of channels, enter only the first telephone number. Subsequent channels are assigned the next consecutive telephone number. For example, if you enter 400 for channels 1 to 4, then channel 1 is assigned phone number 400, channel 2 is assigned phone number 401, and so on. These numbers are also used for channel allocation for IP-to-Tel calls if the Trunk Group's 'Channel Select Mode' parameter is set to **By Dest Phone Number**. **Notes:** ▪ If this field includes alphabetical characters and the phone number is defined for a range of channels (e.g., 1-4), then the phone number must end with a number (e.g., 'user1'). ▪ This field is optional. The logical numbers defined in this field are used when an incoming Tel call doesn't contain the calling number or called number (the latter being determined by the ReplaceEmptyDstWithPortNumber parameter). These numbers are used to replace them. ▪ This field is ignored if routing of IP-to-Tel calls is done according to the Supp Services table, where multiple line extension numbers are configured per port (see "Configuring Multi-Line Extensions and Supplementary Services" on page 465). For this routing method, the 'Channel Select Mode' must be set to **Select Trunk By Supplementary Services Table** in the Trunk Group Settings table (see "Configuring Hunt Group Settings" on

Parameter	Description
	page 367).
Trunk Group ID CLI: trunk-group-id **[TrunkGroup_TrunkGroupNum]**	Defines the Trunk Group ID for the specified channels. The same Trunk Group ID can be assigned to more than one group of channels. If an IP-to-Tel call is assigned to a Trunk Group, the IP call is routed to the channel(s) pertaining to that Trunk Group ID. The valid value can be 0 to 119.
Tel Profile ID CLI: tel-profile-id **[TrunkGroup_ProfileId]**	Assigns a Tel Profile ID to the Trunk Group. **Note:** For configuring Tel Profiles, see "Configuring Tel Profiles" on page 311.

26.2 Configuring Hunt Group Settings

The Hunt Group Settings lets you configure various settings per Trunk Group. The main settings include:

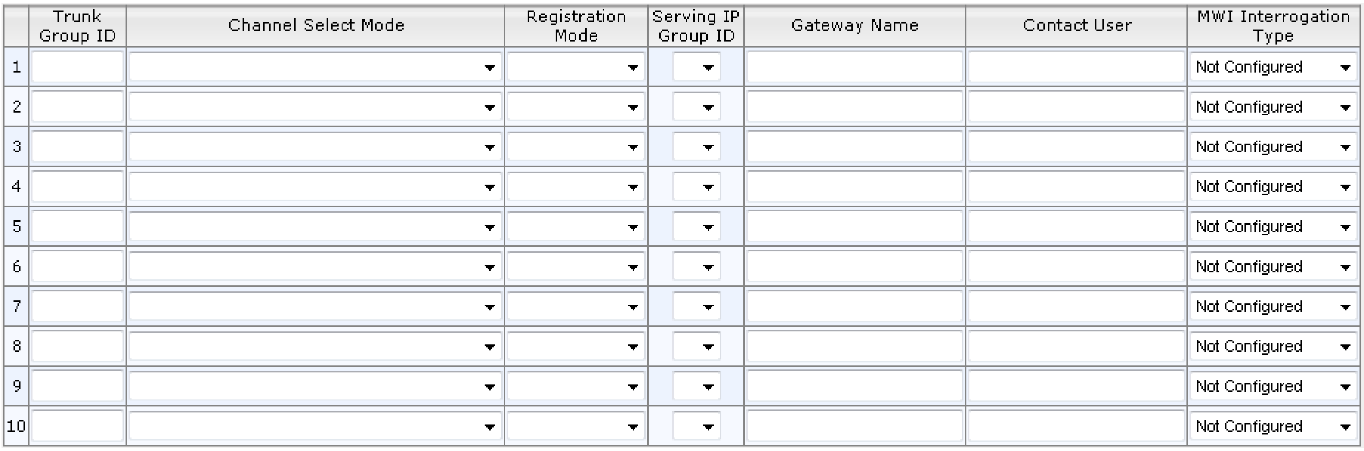
- Channel select method, which defines how the device allocates IP-to-Tel calls to the channels of a Trunk Group.
- Registration method for registering Trunk Groups to remote IP servers (*Serving IP Group*).

For configuring Trunk Groups, see Configuring Trunk Group Table on page 365.

The following procedure describes how to configure settings for Trunk Groups in the Web interface. You can also configure this using the table ini file parameter, TrunkGroupSettings or CLI command, configure voip/gw hunt-or-trunk-group trunk-group-setting.

➢ To configure settings for Hunt Group Settings:

1. Open the Hunt Group Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Trunk Group** > **Hunt Group Settings**).



	Trunk Group ID	Channel Select Mode	Registration Mode	Serving IP Group ID	Gateway Name	Contact User	MWI Interrogation Type
1							Not Configured
2							Not Configured
3							Not Configured
4							Not Configured
5							Not Configured
6							Not Configured
7							Not Configured
8							Not Configured
9							Not Configured
10							Not Configured

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

Table 26-2: Hunt Group Settings Parameter Descriptions

Parameter	Description
Trunk Group ID CLI: trunk-group-id **[TrunkGroupSettings_TrunkGroupId]**	Defines the Trunk Group ID that you want to configure.
Channel Select Mode CLI: channel-select-mode **[TrunkGroupSettings_ChannelSelectMode]**	Defines the method by which IP-to-Tel calls are assigned to the channels of the Trunk Group. ▪ **[0]** By Dest Phone Number = The channel is selected according to the called (destination) number. If the number is not located, the call is released. If the channel is unavailable (e.g., busy), the call is put on call waiting (if call waiting is enabled and no other call is on call waiting); otherwise, the call is released. ▪ **[1]** Cyclic Ascending = The next available channel in the Trunk Group, in ascending cyclic order is selected. After the device reaches the highest channel number in the Trunk Group, it selects the lowest channel number in the Trunk Group, and then starts ascending again. ▪ **[2]** Ascending = The lowest available channel in the Trunk Group is selected, and if unavailable, the next higher channel is selected. ▪ **[3]** Cyclic Descending = The next available channel in descending cyclic order is selected. The next lower channel number in the Trunk Group is always selected. When the device reaches the lowest channel number in the Trunk Group, it selects the highest channel number in the Trunk Group, and then starts descending again. ▪ **[4]** Descending = The highest available channel in the Trunk Group is selected, and if unavailable, the next lower channel is selected. ▪ **[5]** Dest Number + Cyclic Ascending = The channel is selected according to the called number. If the called number isn't found, the next available channel in ascending cyclic order is selected. **Note:** If the called number is located, but the port associated with the number is busy, the call is released. ▪ **[6]** By Source Phone Number = The channel is selected according to the calling number. ▪ [7] Trunk Cyclic Ascending = The channel from the first channel of the next trunk (adjacent to the trunk from which the previous channel was selected) is selected. This option is applicable only to digital interfaces. ▪ [8] Trunk & Channel Cyclic Ascending = The device implements the Trunk Cyclic Ascending and Cyclic Ascending methods to select the channel. This method selects the next physical trunk in the Trunk Group, and then selects the B-channel of this trunk according to the Cyclic Ascending method (i.e., selects the channel after the last allocated channel). This option is applicable only to digital interfaces. For example, if the Trunk Group includes two physical trunks, 0 and 1: ✓ For the first incoming call, the first channel of Trunk 0 is selected.

Parameter	Description
	✓ For the second incoming call, the first channel of Trunk 1 is selected. ✓ For the third incoming call, the second channel of Trunk 0 is selected. ▪ [9] Ring to Hunt Group = The device allocates IP-to-Tel calls to all the FXS ports (channels) in the Hunt Group. When a call is received for the Hunt Group, all telephones connected to the FXS ports belonging to the Hunt Group start ringing. The call is eventually received by whichever telephone first answers the call (after which the other phones stop ringing). This option is applicable only to FXS interfaces. ▪ [10] Select Trunk by Supplementary Services Table = The BRI port/module is selected according to the settings in the ISDN Supplementary Services table (see Configuring Multi-Line Extensions and Supplementary Services on page 465), allowing the routing of IP-to-Tel calls to specific BRI endpoints according to extension number. This option is applicable only to FXS and BRI interfaces. ▪ **[11]** Dest Number + Ascending = The device allocates a channels to incoming IP-to-Tel calls as follows: **a.** The device attempts to route the call to the channel that is associated with the destination (called) number. If located, the call is sent to that channel. **b.** If the number is not located or the channel is unavailable (e.g., busy), the device searches in ascending order for the next available channel in the Trunk Group. If located, the call is sent to that channel. **c.** If all the channels are unavailable, the call is released. **Note:** If this parameter is not configured, the Trunk Group's channel select method is according to the global parameter, ChannelSelectMode.
Registration Mode CLI: registration-mode **[TrunkGroupSettings_RegistrationMode]**	Defines the registration method for the Trunk Group: ▪ **[1]** Per Gateway = (Default) Single registration for the entire device. This is applicable only if a default Proxy or Registrar IP is configured and Registration is enabled (i.e., parameter IsRegisterUsed is set to 1). In this mode, the SIP URI user part in the From, To, and Contact headers is set to the value of the global registration parameter, GWRegistrationName or username if GWRegistrationName is not configured. ▪ **[0]** Per Endpoint = Each channel in the Trunk Group registers individually. The registrations are sent to the 'Serving IP Group ID' if defined in the table, otherwise, it is sent to the default Proxy, and if no default Proxy, then to the Registrar IP. ▪ **[4]** Don't Register = No registrations are sent by endpoints pertaining to the Trunk Group. For example, if the device is configured globally to register all its endpoints (using the parameter ChannelSelectMode), you can exclude some endpoints from being registered

Parameter	Description

	by assigning them to a Trunk Group and configuring the Trunk Group registration mode to 'Don't Register'.
	- **[5]** Per Account = Registrations are sent (or not) to an IP Group, according to the settings in the Account table (see "Configuring Registration Accounts" on page 291).
	An example is shown below of a REGISTER message for registering endpoint "101" using the registration Per Endpoint mode:
	`REGISTER sip:SipGroupName SIP/2.0`
	`Via: SIP/2.0/UDP`
	`10.33.37.78;branch=z9hG4bKac862428454`
	`From: <sip:101@GatewayName>;tag=1c862422082`
	`To: <sip:101@GatewayName>`
	`Call-ID: 9907977062512000232825@10.33.37.78`
	`CSeq: 3 REGISTER`
	`Contact: <sip:101@10.33.37.78>;expires=3600`
	`Expires: 3600`
	`User-Agent: Sip-Gateway/v.6.80A.227.005`
	`Content-Length: 0`
	The "SipGroupName" in the Request-URI is configured in the IP Group table (see "Configuring IP Groups" on page 270).
	Notes:
	- If this parameter is not configured, the registration is performed according to the global registration parameter, ChannelSelectMode.
	- To enable Trunk Group registration, set the global parameter, IsRegisterNeeded to 1. This is unnecessary for 'Per Account' registration mode.
	- If the device is configured globally to register Per Endpoint and an channel group includes four channels to register Per Gateway, the device registers all channels except the first four channels. The group of these four channels sends a single registration request.

Serving IP Group ID CLI: serving-ip-group **[TrunkGroupSettings_ServingIPGroup]**	Assigns an IP Group to where the device sends INVITE messages for this Trunk Group. The actual destination to where these INVITE messages are sent is according to the Proxy Set ID associated with the IP Group. The Request-URI host name in the INVITE and REGISTER messages (except for 'Per Account' registration modes) is set to the value of the 'SIP Group Name' parameter configured in the IP Group table (see "Configuring IP Groups" on page 270). **Notes:** - If this parameter is not configured, the INVITE messages are sent to the default Proxy or according to the Outbound IP Routing table (see "Configuring Outbound IP Routing Table" on page 397). - If the PreferRouteTable parameter is set to 1 (see "Configuring Proxy and Registration Parameters" on page 296), the routing rules in the Outbound IP Routing table take precedence over the selected Serving IP Group ID.

Parameter	Description
Gateway Name CLI: gateway-name **[TrunkGroupSettings_GatewayName]**	Defines the host name for the SIP From header in INVITE messages and for the From/To headers in REGISTER requests. **Note:** If this parameter is not configured, the global parameter, SIPGatewayName is used.
Contact User CLI: contact-user **[TrunkGroupSettings_ContactUser]**	Defines the user part for the SIP Contact URI in INVITE messages and for the From, To, and Contact headers in REGISTER requests. **Notes:** ▪ This parameter is applicable only if the 'Registration Mode' parameter is set to 'Per Account' and registration through the Account table is successful. ▪ If registration fails, the user part in the INVITE Contact header is set to the source party number. ▪ The 'Contact User' parameter in the Account table overrides this parameter (see "Configuring Registration Accounts" on page 291).
Trunk Group Name CLI: trunk-group-name **[TrunkGroupSettings_TrunkGroupName]**	Defines a name for the Trunk Group. This name represents the Trunk Group in the SIP 'tgrp' parameter of the outgoing INVITE messages (according to RFC 4904). For example: `sip:+16305550100;tgrp=`**`TG-1`**`;trunk-context=+1-630@isp.example.net;user=phone` The valid value can be a string of up to 20 characters. By default, no name is configured. **Notes:** ▪ If this parameter is not configured, the Trunk Group decimal number is used in the SIP 'tgrp' parameter. ▪ This feature is enabled by any of the following parameters: ✓ UseSIPtgrp ✓ UseBroadsoftDTG ▪ Currently, this parameter can only be configured using the ini file.

Parameter	Description
MWI Interrogation Type CLI: mwi-interrogation-type [TrunkGroupSettings_MWIInterrogationType]	Defines MWI QSIG-to-IP interworking for interrogating MWI supplementary services: ▪ [255] Not Configured ▪ [0] None = Disables the feature. ▪ [1] Use Activate Only = MWI Interrogation messages are not sent and only "passively" responds to MWI Activate requests from the PBX. ▪ [2] Result Not Used = MWI Interrogation messages are sent, but the result is not used. Instead, the device waits for MWI Activate requests from the PBX. ▪ [3] Use Result = MWI Interrogation messages are sent, its results are used, and the MWI Activate requests are used. MWI Activate requests are interworked to SIP NOTIFY MWI messages. The SIP NOTIFY messages are sent to the IP Group defined by the NotificationIPGroupID parameter. **Note:** This parameter appears in the table only if the VoiceMailInterface parameter is set to 3 (QSIG). Configuring Voice Mail on page 472.

27 Manipulation

This section describes the configuration of various manipulation processes.

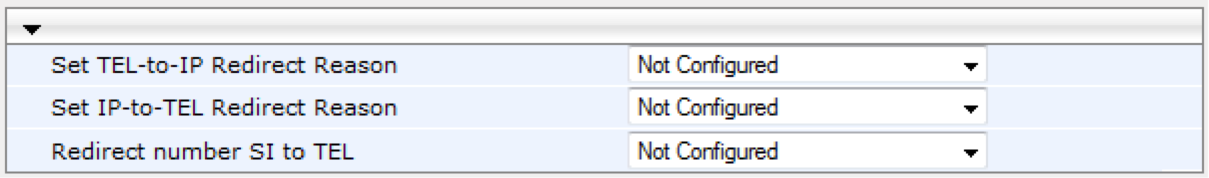
27.1 Configuring General Settings

The General Settings page allows you to configure general manipulation parameters. For a description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 763.

- **To configure the general manipulation parameters:**

1. Open the General Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** >**General Settings**).

Figure 27-1: General Settings Page



Set TEL-to-IP Redirect Reason	Not Configured
Set IP-to-TEL Redirect Reason	Not Configured
Redirect number SI to TEL	Not Configured

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

27.2 Configuring Source/Destination Number Manipulation Rules

The number manipulation tables let you configure rules for manipulating source and destination telephone numbers for IP-to-Tel and Tel-to-IP calls. The number manipulation tables include the following:

- **Tel-to-IP calls:**
  - Source Phone Number Manipulation Table for Tel > IP Calls (up to 120 entries)
  - Destination Phone Number Manipulation Table for Tel > IP Calls (up to 120 entries)
- **IP-to-Tel calls:**
  - Source Phone Number Manipulation Table for IP > Tel Calls (up to 120 entries)
  - Destination Phone Number Manipulation Table for IP > Tel Calls (up to 120 entries)

The number manipulation tables provide two configuration areas:

- Matching characteristics (*Rule*) of incoming call, for example, prefix of destination number.
- Manipulation operation (*Action*), for example, remove user-defined number of digits from the left of the number.

If the incoming call matches the characteristics of a rule, its manipulation action is applied. The device searches a matching manipulation rule starting from the first entry (i.e., top of the table). In other words, a rule at the top of the table takes precedence over a rule defined lower down in the table. Thus, define more specific rules above more generic rules. For example, if you enter 551 in Index 1 and 55 in Index 2, the device applies rule 1 to numbers that start with 551 and applies rule 2 to numbers that start with 550, 552, 553, and so on until 559. However, if you enter 55 in Index 1 and 551 in Index 2, the device applies rule 1 to all numbers that start with 55, including numbers that start with 551.

You can perform a second "round" (additional) of source and destination number manipulations for IP-to-Tel calls, on an already manipulated number. The initial and additional number manipulation rules are both configured in the number manipulation tables for Ip-to-Tel calls. The additional manipulation is performed on the initially manipulated number. Thus, for complex number manipulation schemes, you only need to configure relatively few manipulation rules in these tables (that would otherwise require many rules). To enable this additional manipulation, use the following parameters:

- Source number manipulation - PerformAdditionalIP2TELSourceManipulation
- Destination number manipulation - PerformAdditionalIP2TELDestinationManipulation

Telephone number manipulation can be useful, for example, for the following:

- Stripping or adding dialing plan digits from or to the number, respectively. For example, a user may need to first dial 9 before dialing the phone number to indicate an external line. This number 9 can then be removed by number manipulation before the call is setup.
- Allowing or blocking Caller ID information according to destination or source prefixes. For more information on Caller ID, see Configuring Caller Display Information on page 480.
- For digital modules only: Assigning Numbering Plan Indicator (NPI) and Type of Numbering (TON) to IP-to-Tel calls. The device can use a single global setting for NPI/TON classification or it can use the setting in the manipulation tables on a call-by-call basis.

Number manipulation can occur before or after a routing decision is made. For example, you can route a call to a specific Trunk Group according to its original number, and then you can remove or add a prefix to that number before it is routed. To determine when number manipulation is performed, use the 'IP to Tel Routing Mode' parameter (RouteModeIP2Tel) described in "Configuring Inbound IP Routing Table" on page 406, and 'Tel to IP Routing Mode' parameter (RouteModeTel2IP) described in "Configuring Outbound IP Routing Table" on page 397.

The device manipulates the number in the following order: 1) strips digits from the left of the number, 2) strips digits from the right of the number, 3) retains the defined number of digits, 4) adds the defined prefix, and then 5) adds the defined suffix.

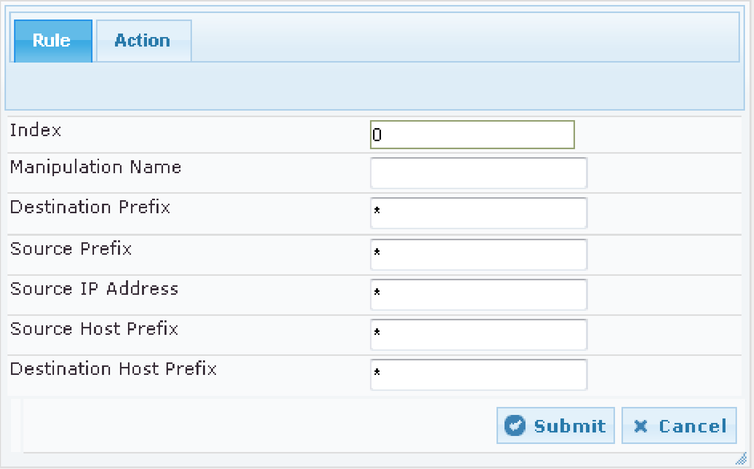
The following procedure describes how to configure number manipulation rules in the Web interface. You can also configure this using the following management tools:

- **Destination Phone Number Manipulation Table for IP > Tel Calls table:** ini file table parameter, NumberMapIP2Tel or CLI command, configure voip/gw manipulations dst-number-map-ip2tel
- **Destination Phone Number Manipulation Table for Tel > IP Calls table:** ini file table parameter, NumberMapTel2IP or CLI command, configure voip/gw manipulations dst-number-map-tel2ip
- **Source Phone Number Manipulation Table for IP > Tel Calls table:** ini file table parameter, SourceNumberMapIP2Tel or CLI command, configure voip/gw manipulations src-number-map-ip2tel
- **Source Phone Number Manipulation Table for Tel > IP Calls table:** ini file table parameter, SourceNumberMapTel2IP or CLI command, configure voip/gw manipulations src-number-map-tel2ip

➢ To configure a number manipulation rule:

1. Open the required Number Manipulation page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Dest Number IP->Tel**, **Dest Number Tel->IP**, **Source Number IP->Tel**, or **Source Number Tel->IP**); the relevant Manipulation table page is displayed.
2. Click **Add**; the following dialog box appears:

Figure 27-2: Number Manipulation Table - Add Dialog Box



Index	0
Manipulation Name	
Destination Prefix	*
Source Prefix	*
Source IP Address	*
Source Host Prefix	*
Destination Host Prefix	*

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

The table below shows configuration examples of Tel-to-IP source phone number manipulation rules, where:

- **Rule 1:** When the destination number has the prefix 03 (e.g., 035000), source number prefix 201 (e.g., 20155), and from source IP Group ID 2, the source number is changed to, for example, 97120155.
- **Rule 2:** When the source number has prefix 1001 (e.g., 1001876), it is changed to 587623.
- **Rule 3:** When the source number has prefix 123451001 (e.g., 1234510012001), it is changed to 20018.
- **Rule 4:** When the source number has prefix from 30 to 40 and a digit (e.g., 3122), it is changed to 2312.
- **Rule 5:** When the destination number has the prefix 6, 7, or 8 (e.g., 85262146), source number prefix 2001, it is changed to 3146.

Parameter	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Source IP Group	2	0	-	-	-
Destination Prefix	03		*	*	[6,7,8]
Source Prefix	201	1001	123451001#	[30-40]x	2001
Stripped Digits from Left	-	4	-	-	5
Stripped Digits from Right	-	-	-	1	-

Parameter	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5
Prefix to Add	971	5	-	2	3
Suffix to Add	-	23	8	-	-
Number of Digits to Leave	-	-	4	-	-
Presentation	Allowed	Restricted	-	-	-

Table 27-1: Number Manipulation Tables Parameter Descriptions

Parameter	Description
Index **[_Index]**	Defines an index number for the new table record.
Manipulation Name **[_ManipulationName]**	Defines an arbitrary name to easily identify the manipulation rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Characteristics (Rule)	
Web: Destination Prefix EMS: Prefix CLI: dst-prefix **[_DestinationPrefix]**	Defines the destination (called) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a called number. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761.
Web/EMS: Source Prefix CLI: src-prefix **[_SourcePrefix]**	Defines the source (calling) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a calling number. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761.
Web/EMS: Source IP Address CLI: src-ip-address **[_SourceAddress]**	Defines the source IP address of the caller. This is obtained from the Contact header in the INVITE message. **Notes:** ▪ This parameter is applicable only to the number manipulation tables for IP-to-Tel calls. ▪ The source IP address can include the 'x' wildcard to represent single digits. For example: 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99. ▪ The source IP address can include the asterisk (*) wildcard to represent any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255.
Web: Source Host Prefix CLI: src-host-prefix **[_SrcHost]**	Defines the URI host name prefix of the incoming SIP INVITE message in the From header. **Notes:** ▪ This parameter is applicable only to the number manipulation tables for IP-to-Tel calls. ▪ The asterisk (*) wildcard can be used to denote any prefix. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-

Parameter	Description
	Asserted-Identity URI host name (instead of the From header).
Web: Destination Host Prefix CLI: dst-host-prefix **[_DestHost]**	Defines the Request-URI host name prefix of the incoming SIP INVITE message. **Notes:** ▪ This parameter is applicable only to the number manipulation tables for IP-to-Tel calls. ▪ The asterisk (*) wildcard can be used to denote any prefix.
Web: Source Trunk Group CLI: src-trunk-group-id **[_SrcTrunkGroupID]**	Defines the source Trunk Group ID for Tel-to-IP calls. To denote all Trunk Groups, leave this field empty. **Notes:** ▪ The value -1 indicates that this field is ignored in the rule. ▪ This parameter is applicable only to the number manipulation tables for Tel-to-IP calls. ▪ For IP-to-IP call routing, this parameter is not required (i.e., leave the field empty).
Web: Source IP Group CLI: src-ip-group-id **[_SrcIPGroupID]**	Defines the IP Group from where the IP call originated. Typically, the IP Group of an incoming INVITE is determined or classified using the Inbound IP Routing table. If not used (i.e., any IP Group), leave the field empty. **Notes:** ▪ The value -1 indicates that this field is ignored. ▪ This parameter is applicable only to the number manipulation tables for Tel-to-IP calls. ▪ If this Source IP Group has a Serving IP Group, then all calls from this Source IP Group are sent to the Serving IP Group. In this scenario, this table is used only if the PreferRouteTable parameter is set to 1.
Web: Destination IP Group CLI: dst-ip-group-id **[_DestIPGroupID]**	Defines the IP Group to where the call is sent. **Notes:** ▪ The value -1 indicates that this field is ignored. ▪ This parameter is applicable only to the Destination Phone Number Manipulation Table for Tel -> IP Calls.
Operation (Action)	
Web: Stripped Digits From Left EMS: Number Of Stripped Digits CLI: remove-from-left **[_RemoveFromLeft]**	Defines the number of digits to remove from the left of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 1234.
Web: Stripped Digits From Right EMS: Number Of Stripped Digits CLI: remove-from-right **[RemoveFromRight]**	Defines the number of digits to remove from the right of the telephone number prefix. For example, if you enter 3 and the phone number is 5551234, the new phone number is 5551.
Web: Prefix to Add EMS: Prefix/Suffix To Add CLI: prefix-to-add **[Prefix2Add]**	Defines the number or string that you want added to the front of the telephone number. For example, if you enter 9 and the phone number is 1234, the new number is 91234.

Parameter	Description
Web: Suffix to Add EMS: Prefix/Suffix To Add CLI: suffix-to-add **[Suffix2Add]**	Defines the number or string that you want added to the end of the telephone number. For example, if you enter 00 and the phone number is 1234, the new number is 123400.
Web/EMS: Number of Digits to Leave CLI: num-of-digits-to-leave **[LeaveFromRight]**	Defines the number of digits that you want to keep from the right of the phone number. For example, if you enter 4 and the phone number is 00165751234, then the new number is 1234.
Web: NPI EMS: Number Plan CLI: npi [NumberPlan]	Defines the Numbering Plan Indicator (NPI). ▪ [0] Unknown (default) ▪ [9] Private ▪ [1] E.164 Public ▪ [-1] Not Configured = value received from PSTN/IP is used **Notes:** ▪ This parameter is applicable only to number manipulation tables for IP-to-Tel calls. ▪ NPI can be used in the SIP Remote-Party-ID header by using the EnableRPIHeader and AddTON2RPI parameters. ▪ For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 395.
Web: TON EMS: Number Type CLI: ton [NumberType]	Defines the Type of Number (TON). ▪ If you selected 'Unknown' for the NPI, you can select Unknown [0]. ▪ If you selected 'Private' for the NPI, you can select Unknown [0], Level 2 Regional [1], Level 1 Regional [2], PISN Specific [3] or Level 0 Regional (Local) [4]. ▪ If you selected 'E.164 Public' for the NPI, you can select Unknown [0], International [1], National [2], Network Specific [3], Subscriber [4] or Abbreviated [6]. The default is 'Unknown'. **Notes:** ▪ This parameter is applicable only to number manipulation tables for IP-to-Tel calls. ▪ TON can be used in the SIP Remote-Party-ID header by using the EnableRPIHeader and AddTON2RPI parameters. ▪ For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 395.
Web: Presentation EMS: Is Presentation Restricted CLI: is-presentation-restricted **[IsPresentationRestricted]**	Enables caller ID. ▪ Not Configured = Privacy is determined according to the Caller ID table (see Configuring Caller Display Information on page 480). ▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination/source prefixes. ▪ **[1]** Restricted = Restricts Caller ID information for these prefixes. **Notes:** ▪ This field is applicable only to number manipulation tables for source phone number manipulation. ▪ If this field is set to **Restricted** and the 'Asserted Identity Mode' (AssertedIdMode) parameter is set to **Add P-Asserted-Identity**, the From header in the INVITE message includes the following: From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.

27.3 Manipulating Number Prefix

The device supports a notation for adding a prefix where part of the prefix is first extracted from a user-defined location in the original destination or source number. This notation is entered in the 'Prefix to Add' field in the Number Manipulation tables (see "Configuring Source/Destination Number Manipulation" on page 373): x[n,l]y...

where,

- *x* = any number of characters/digits to add at the beginning of the number (i.e. first digits in the prefix).
- *[n,l]* = defines the location in the original destination or source number where the digits *y* are added:
  - *n* = location (number of digits counted from the left of the number) of a specific string in the original destination or source number.
  - *l* = number of digits that this string includes.
- y = prefix to add at the specified location.

For example, assume that you want to manipulate an incoming IP call with destination number +5492028888888 (area code 202 and phone number 8888888) to the number 0202158888888. To perform such a manipulation, the following configuration is required in the Number Manipulation table:

1. The following notation is used in the 'Prefix to Add' field:

   0[5,3]15

   where,

   - 0 is the number to add at the beginning of the original destination number.
   - [5,3] denotes a string that is located after (and including) the fifth character (i.e., the first '2' in the example) of the original destination number, and its length being three digits (i.e., the area code 202, in the example).
   - 15 is the number to add immediately after the string denoted by [5,3] - in other words, 15 is added after (i.e. to the right of) the digits 202.
2. The first seven digits from the left are removed from the original number, by entering "7" in the 'Stripped Digits From Left' field.

Table 27-2: Example of Configured Rule for Manipulating Prefix using Special Notation

Parameter	Rule 1
Destination Prefix	+5492028888888
Source Prefix	*
Source IP Address	*
Stripped Digits from Left	7
Prefix to Add	0[5,3]15

In this configuration example, the following manipulation process occurs:

1. The prefix is calculated as 020215.
2. The first seven digits from the left are removed from the original number, thereby changing the number to 8888888.
3. The prefix that was previously calculated is then added.

27.4 SIP Calling Name Manipulations

The calling name manipulation tables lets you configure up to 120 manipulation rules for manipulating the calling name (i.e., caller ID) in SIP messages, for IP-to-Tel and Tel-to-IP calls. Manipulation includes modifying or removing the calling name. The calling name manipulation tables include the following:

- Calling Name Manipulation Table for IP-to-Tel Calls table
- Calling Name Manipulation Table for Tel-to-IP Calls table

For example, assume that an incoming SIP INVITE message includes the following header:

```
P-Asserted-Identity: "company:john" sip:6666@78.97.79.104
```

Using the Calling Name Manipulation Table for IP-to-Tel table, the text "company" can be changed to "worker" in the outgoing INVITE, as shown below:

```
P-Asserted-Identity: "worker:john" sip:996666@10.13.83.10
```

The calling name manipulation tables provide two configuration areas:

- Matching characteristics (*Rule*) of incoming call, for example, prefix of destination number.
- Manipulation operation (*Action*), for example, remove user-defined number of digits from the left of the calling name.

If the incoming call matches the characteristics of a rule, its manipulation action is applied.

> **Note:** For using the Calling Name Manipulation Table for Tel-to-IP Calls table for retrieving the calling name (display name) from an Active Directory using LDAP queries, see Querying the AD for Calling Name on page 233.

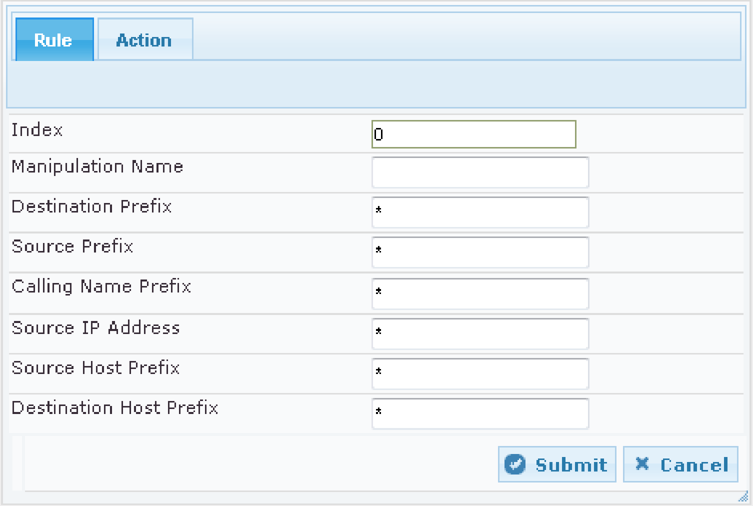
The following procedure describes how to configure calling name manipulation rules in the Web interface. You can also configure these rules using the the following management tools:

- Calling Name Manipulation Table for Tel-to-IP Calls table: table *ini* file parameter, CallingNameMapTel2Ip or CLI command, configure voip/gw manipulations calling-name-map-tel2ip
- Calling Name Manipulation Table for IP-to-Tel Calls table: table *ini* file parameter, CallingNameMapIp2Tel or CLI command, configure voip/gw manipulations calling-name-map-ip2tel

➢ **To configure calling name manipulation rules:**

1. Open the required calling name manipulations page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Calling Name IP->Tel** or **Calling Name Tel->IP**).
2. Click **Add**; the following dialog box appears:

Figure 27-3: Calling Name Manipulation IP2Tel - Rule Tab



Index	0
Manipulation Name	
Destination Prefix	*
Source Prefix	*
Calling Name Prefix	*
Source IP Address	*
Source Host Prefix	*
Destination Host Prefix	*

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

Table 27-3: Calling Name Manipulation Tables Parameter Descriptions

Parameter	Description
Index **[_Index]**	Defines an index number for the new table record.
Manipulation Name CLI: manipulation-name **[_ManipulationName]**	Defines an arbitrary name to easily identify the manipulation rule. The valid value is a string of up to 20 characters.
Matching Characteristics (Rule)	
Web: Destination Prefix CLI: dst-prefix **[_DestinationPrefix]**	Defines the destination (called) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a called number. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761. The default value is the asterisk (*) symbol (i.e., any destination prefix).
Web/EMS: Source Prefix CLI: src-prefix **[_SourcePrefix]**	Defines the source (calling) telephone number prefix and/or suffix. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. You can also use the $ sign to denote calls without a calling number. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761. The default value is the asterisk (*) symbol (i.e., any source prefix).

Parameter	Description
Web: Calling Name Prefix CLI: calling-name-prefix **[_CallingNamePrefix]**	Defines the caller name (i.e., caller ID) prefix. You can use special notations for denoting the prefix. For example, to denote calls without a calling name, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761. The default value is the asterisk (*) symbol (i.e., any calling name prefix).
Web: Source Trunk Group ID CLI: src-trunk-group-id **[_SrcTrunkGroupID]**	Defines the source Trunk Group ID from where the Tel-to-IP call was received. The default value is -1, which denotes any Trunk Group. **Note:** This parameter is applicable only to the Calling Name Manipulation Table for Tel-to-IP Calls table.
Web: Source IP Group ID CLI: src-ip-group-id [_SrcIPGroupID]	Defines the IP Group from where the IP call was received. The default value is -1, which denotes any IP Group. **Notes:** ▪ This parameter is applicable only to the IP-to-IP application. ▪ This parameter is applicable only to the Calling Name Manipulation Table for Tel-to-IP Calls table.
Web/EMS: Source IP Address CLI: src-ip-address **[_SourceAddress]**	Defines the source IP address of the caller, for IP-to-Tel calls. The source IP address appears in the SIP Contact header in the INVITE message. The default value is the asterisk (*) symbol (i.e., any IP address). The source IP address can include the following wildcards: ▪ "x" wildcard: represents single digits. For example, 10.8.8.xx represents all IP addresses between 10.8.8.10 to 10.8.8.99. ▪ "*" (asterisk) wildcard: represents any number between 0 and 255. For example, 10.8.8.* represents all IP addresses between 10.8.8.0 and 10.8.8.255. **Note:** This parameter is applicable only to the Calling Name Manipulation Table for IP-to-Tel Calls table.
Web: Source Host Prefix CLI: src-host-prefix **[_SrcHost]**	Defines the URI host name prefix of the incoming SIP INVITE message in the From header. The default value is the asterisk (*) symbol (i.e., any source host prefix). **Notes:** ▪ This parameter is applicable only to the Calling Name Manipulation Table for IP-to-Tel Calls table. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-Asserted-Identity URI host name (instead of the From header).
Web: Destination Host Prefix CLI: dst-host-prefix **[_DestHost]**	Defines the Request-URI host name prefix of the incoming SIP INVITE message. The default value is the asterisk (*) symbol (i.e., any destination host prefix). **Note:** This parameter is applicable only to the Calling Name Manipulation Table for IP-to-Tel Calls table.
Operation (Action)	
Web: Stripped Digits From Left EMS: Number Of Stripped	Defines the number of characters to remove from the left of the calling name. For example, if you enter 3 and the calling name is

Parameter	Description
Digits CLI: remove-from-left **[_RemoveFromLeft]**	"company:john", the new calling name is "pany:john".
Web: Stripped Digits From Right EMS: Number Of Stripped Digits CLI: remove-from-right **[_RemoveFromRight]**	Defines the number of characters to remove from the right of the calling name. For example, if you enter 3 and the calling name is "company:name", the new name is "company:n".
Web/EMS: Number of Digits to Leave CLI: num-of-digits-to-leave **[LeaveFromRight]**	Defines the number of characters that you want to keep from the right of the calling name. For example, if you enter 4 and the calling name is "company:name", the new name is "name".
Web: Prefix to Add EMS: Prefix/Suffix To Add CLI: prefix-to-add **[_Prefix2Add]**	Defines the number or string to add at the front of the calling name. For example, if you enter ITSP and the calling name is "company:name", the new name is ITSPcompany:john".
Web: Suffix to Add EMS: Prefix/Suffix To Add CLI: suffix-to-add **[_Suffix2Add]**	Defines the number or string to add at the end of the calling name. For example, if you enter 00 and calling name is "company:name", the new name is "company:name00".

27.5 Configuring Redirect Number IP to Tel

The redirect number manipulation tables let you configure rules for manipulating the redirect number received in SIP messages. The redirect number manipulation tables include:

- Redirect Number IP-to-Tel table: This table defines IP-to-Tel redirect number manipulation. You can manipulate the value of the received SIP Diversion, Resource-Priority, or History-Info headers, which is then added to the Redirecting Number Information Element (IE) in the ISDN Setup message sent to the Tel side. This also includes the reason for the call redirection. This is configured in the Redirect Number IP > Tel table.
- Redirect Number Tel to IP table: This table defines Tel-to-IP redirect number manipulation. You can manipulate the prefix of the redirect number, received from the Tel side, in the outgoing SIP Diversion, Resource-Priority, or History-Info headers sent to the IP side. This is configured in the Redirect Number Tel > IP table.

The redirect number manipulation tables provide two configuration areas:

- Matching characteristics (*Rule*) of incoming call, for example, prefix of redirect number.
- Manipulation operation (*Action*), for example, remove user-defined number of digits from the left of the redirect number.

If the incoming call matches the characteristics of a rule, its manipulation action is applied.

Notes:

- If the device copies the received destination number to the outgoing SIP redirect number (enabled by the CopyDest2RedirectNumber parameter), no redirect number Tel-to-IP manipulation is done.
- The manipulation rules are done in the following order: Stripped Digits From Left, Stripped Digits From Right, Number of Digits to Leave, Prefix to Add, and then Suffix to Add.
- The device uses the 'Redirect Prefix' parameter before it manipulates the prefixi.

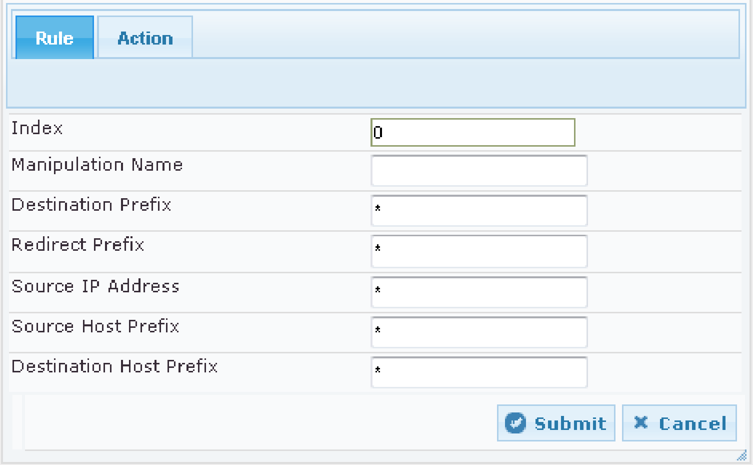
The following procedure describes how to configure redirect number manipulation rules in the Web interface. You can also configure these rules using the following management tools:

- Redirect Number IP to Tel table: ini file parameter, RedirectNumberMapIp2Tel or CLI command, configure voip/gw manipulations redirect-number-map-ip2tel
- Redirect Number Tel to IP table: ini file parameter, RedirectNumberMapTel2Ip or CLI command, configure voip/gw manipulations redirect-number-map-tel2ip (CLI)

➢ To configure a redirect number manipulation rule:

1. Open the redirect number manipulation table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Redirect Number Tel > IP** or Redirect Number IP > Tel).
2. Click **Add**; the following dialog box appears (e.g., Redirect Number Tel-to-IP table):

Figure 27-4: Redirect Number Manipulation (e.g., Tel to IP)



Index	0
Manipulation Name	
Destination Prefix	*
Redirect Prefix	*
Source IP Address	*
Source Host Prefix	*
Destination Host Prefix	*

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

Table 27-4: Redirect Number Manipulation Tables Parameter Description

Parameter	Description
Index **[_Index]**	Defines an index number for the new table record.
Manipulation Name CLI: manipulation-name **[_ManipulationName]**	Defines an arbitrary name to easily identify the manipulation rule. The valid value is a string of up to 20 characters.
Matching Characteristics - Rule	

Parameter	Description
Web/EMS: Destination Prefix CLI: dst-prefix **[_DestinationPrefix]**	Defines the destination (called) telephone number prefix. The default value is the asterisk (*) symbol (i.e., any number). For manipulating the diverting and redirected numbers for call diversion, you can use the strings "DN" and "RN" to denote the destination prefix of these numbers. For more information, see Manipulating Redirected and Diverted Numbers for Call Diversion on page 387.
Web/EMS: Redirect Prefix CLI: redirect-prefix **[_RedirectPrefix]**	Defines the redirect telephone number prefix. The default value is the asterisk (*) symbol (i.e., any number prefix).
Web: Source Trunk Group ID CLI: src-trunk-group-id **[_SrcTrunkGroupID]**	Defines the Trunk Group from where the Tel call is received. To denote any Trunk Group, leave this field empty. The value -1 indicates that this field is ignored in the rule. **Notes:** ▪ This parameter is applicable only to the Redirect Number Tel > IP table. ▪ This parameter is not applicable to IP-to-IP call routing.
Source IP Group ID CLI: src-ip-group-id [_SrcIPGroupID]	Defines the IP Group from where the IP call originated. Typically, the IP Group of an incoming INVITE is determined or classified by the Inbound IP Routing table. If not used (i.e., any IP Group), leave the field empty. The value -1 indicates that it is ignored in the rule. **Notes:** ▪ This parameter is applicable only to the Redirect Number Tel > IP table. ▪ This parameter is applicable only to the IP-to-IP application.
Web/EMS: Source IP Address CLI: src-ip-address [_SourceAddress]	Defines the IP address of the caller. The IP address appears in the SIP Contact header of the incoming INVITE message. The default value is the asterisk (*) symbol (i.e., any IP address). The value can include the following wildcards: ▪ "x": represents single digits, for example, 10.8.8.xx denotes all addresses between 10.8.8.10 and 10.8.8.99. ▪ "*": represents any number between 0 and 255, for example, 10.8.8.* denotes all addresses between 10.8.8.0 and 10.8.8.255. **Note:** This parameter is applicable only to the Redirect Number IP-to-Tel table.
Web: Source Host Prefix CLI: src-host-prefix [_SrcHost]	Defines the URI host name prefix of the caller. The host name appears in the SIP From header of the incoming SIP INVITE message. The default value is the asterisk (*) symbol (i.e., any host name prefix). **Notes:** ▪ This parameter is applicable only to the Redirect Number IP-to-Tel table. ▪ If the P-Asserted-Identity header is present in the incoming INVITE message, then the value of this parameter is compared to the P-Asserted-Identity URI host name (instead of to the From header).
Web: Destination Host Prefix	Defines the Request-URI host name prefix, which appears in the

Parameter	Description
CLI: dst-host-prefix [_DestHost]	incoming SIP INVITE message. The default value is the asterisk (*) symbol (i.e., any prefix). **Note:** This parameter is applicable only to the Redirect Number IP-to-Tel table.
Operation (Action)	
Web: Stripped Digits From Left EMS: Remove From Left CLI: remove-from-left **[_RemoveFromLeft]**	Defines the number of digits to remove from the left of the redirect number prefix. For example, if you enter 3 and the redirect number is 5551234, the new number is 1234.
Web: Stripped Digits From Right EMS: Remove From Right CLI: remove-from-right **[_RemoveFromRight]**	Defines the number of digits to remove from the right of the redirect number prefix. For example, if you enter 3 and the redirect number is 5551234, the new number is 5551.
Web/EMS: Number of Digits to Leave CLI: num-of-digits-to-leave **[_LeaveFromRight]**	Defines the number of digits that you want to retain from the right of the redirect number.
Web/EMS: Prefix to Add CLI: prefix-to-add **[_Prefix2Add]**	Defines the number or string that you want added to the front of the redirect number. For example, if you enter 9 and the redirect number is 1234, the new number is 91234.
Web/EMS: Suffix to Add CLI: suffix-to-add **[_Suffix2Add]**	Defines the number or string that you want added to the end of the redirect number. For example, if you enter 00 and the redirect number is 1234, the new number is 123400.
Web: Presentation EMS: Is Presentation Restricted CLI: is-presentation-restricted **[_IsPresentationRestricted]**	Enables caller ID. ▪ Not Configured = Privacy is determined according to the Caller ID table (see Configuring Caller Display Information on page 480). ▪ **[0]** Allowed = Sends Caller ID information when a call is made using these destination / source prefixes. ▪ **[1]** Restricted = Restricts Caller ID information for these prefixes. **Note:** If this parameter is set to **Restricted** and the 'AssertedIdMode' parameter is set to **Add P-Asserted-Identity,** the From header in the INVITE message includes the following: `From: 'anonymous' <sip: anonymous@anonymous.invalid> and 'privacy: id' header.`
Web: TON EMS: Number Type CLI: ton [_NumberType]	Defines the Type of Number (TON). The default is Not Configured [-1]. ▪ If NPI is set to Unknown, you can set TON to Unknown [0]. ▪ If NPI is set to Private, you can set TON to Unknown [0], International [1], National [2], Network Specific [3] or Subscriber [4]. ▪ If NPI is set to E.164 Public, you can set TON to Unknown [0], International [1], National [2], Network Specific [3], Subscriber [4] or Abbreviated [6]. For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 395.

Parameter	Description
Web: NPI EMS: Number Plan CLI: npi [_NumberPlan]	Defines the Numbering Plan Indicator (NPI). ▪ [-1] Not Configured = (Default) Value received from PSTN/IP is used ▪ [0] Unknown ▪ [1] E.164 Public ▪ [9] Private For more information on available NPI/TON values, see Numbering Plans and Type of Number on page 395.

27.6 Manipulating Redirected and Diverted Numbers for Call Diversion

You can configure manipulation rules to manipulate the Diverted-to and Diverting numbers received in the incoming Call Redirection Facility message for call diversion, which is interworked to outgoing SIP 302 responses. This feature is applicable to the Euro ISDN and QSIG variants, and to IP-to-Tel calls.

The incoming redirection Facility message includes, among other parameters, the Diverted-to number and Diverting number. The Diverted-to number (i.e., new destination) is mapped to the user part in the Contact header of the SIP 302 response. The Diverting number is mapped to the user part in the Diversion header of the SIP 302 response.

These two numbers can be manipulated by entering the following special strings in the 'Destination Prefix' field of the Redirect Number Tel-to-IP table:

- "RN" - used in the rule to manipulate the Redirected number (i.e., originally called number or Diverting number).
- "DN" - used in the rule to manipulate the Diverted-to number (i.e., the new called number or destination). This manipulation is done on the user part in the Contact header of the SIP 302 response.

For example, assume the following required manipulation:

- Manipulate Redirected number 6001 (originally called number) to 6005
- Manipulate Diverted-to number 8002 (the new called number or destination) to 8005

The configuration in the Redirect Number Tel-to-IP table is as follows:

Table 27-5: Redirect Number Configuration Example

Parameter	Rule 1	Rule 2
Destination Prefix	RN	DN
Redirect Prefix	6	8
Stripped Digits From Right	1	1
Suffix to Add	5	5
Number of Digits to Leave	5	-

After the above manipulation is done, the device sends the following outgoing SIP 302 response:

```
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/TLS 10.33.45.68;branch=z9hG4bKac54132643;alias
From: "MP118 1" <sip:8001@10.33.45.68>;tag=1c54119560
```

```
To: <sip:6001@10.33.45.69;user=phone>;tag=1c664560944
Call-ID: 541189832710201115142@10.33.45.68
CSeq: 1 INVITE
Contact: <sip:8005@10.33.45.68;user=phone>
Supported: em,timer,replaces,path,early-session,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE,UPDATE
Diversion: <tel:6005>;reason=unknown;counter=1
Server: Audiocodes-Sip-Gateway-IPmedia 260_UN/v.6.80A.227.005
Reason: SIP ;cause=302 ;text="302 Moved Temporarily"
Content-Length: 0
```

27.7 Mapping NPI/TON to SIP Phone-Context

The Phone Context table lets you configure rules for mapping the Numbering Plan Indication (NPI) and Type of Number (TON) to the SIP 'phone-context' parameter, and vice versa. The 'phone-context' parameter appears in the standard SIP headers where a phone number is used (i.e., Request-URI, To, From, and Diversion). When a call is received from the ISDN/Tel side, the NPI and TON are compared against the table and the matching 'phone-context' value is used in the outgoing SIP INVITE message. The same mapping occurs when an INVITE with a 'phone-context' parameter is received.

For example, for a Tel-to-IP call with NPI/TON set as E164 National (values 1/2), the device can send the following SIP INVITE URI:

```
sip:12365432;phone-context= na.e.164.nt.com
```

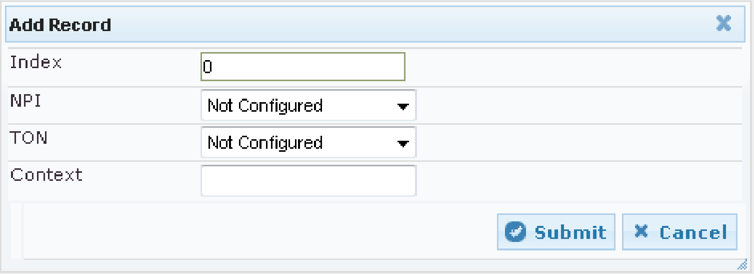
For an IP-to-Tel call, if the incoming INVITE contains this 'phone-context' (e.g. "phone-context= na.e.164.nt.com"), the NPI/TON of the called number in the outgoing Setup message is changed to E164 National.

The following procedure describes how to configure NPI/TON-SIP phone-context mapping rules in the Web interface. You can also configure this using the table ini file parameter, PhoneContext or CLI command, configure voip > gw manipulations phone-context-table.

➢ **To configure NPI/TON-SIP phone-context mapping rules:**

1. Open the Phone Context Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Phone Context**).
2. Click **Add**; the following dialog box appears:

Figure 27-5: Phone Context Table Page



Add Record

Index	0
NPI	Not Configured
TON	Not Configured
Context	

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

Note: You can configure multiple rows with the same NPI/TON or same SIP 'phone-context'. In such a configuration, a Tel-to-IP call uses the first matching rule in the table.

Table 27-6: Phone Context Table Parameter Description

Parameter	Description
Index **[PhoneContext_Index]**	Defines an index number for the new table record.
Add Phone Context As Prefix CLI: configure voip > gw manipulations general-setting > add-ph-cntxt-as-pref **[AddPhoneContextAsPrefix]**	Determines whether the received SIP 'phone-context' parameter is added as a prefix to the outgoing ISDN Setup message (for digital interfaces) with called and calling numbers. ▪ **[0]** Disable (default) ▪ **[1]** Enable
NPI CLI: npi **[PhoneContext_Npi]**	Defines the Number Plan Indicator (NPI). ▪ **[0]** Unknown (default) ▪ **[1]** E.164 Public ▪ **[9]** Private For a detailed list of the available NPI/TON values, see Numbering Plans and Type of Number on page 395.
TON CLI: ton **[PhoneContext_Ton]**	Defines the Type of Number (TON). ▪ If you selected Unknown as the NPI, you can select Unknown **[0]**. ▪ If you selected Private as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** Level 2 Regional ✓ **[2]** Level 1 Regional ✓ **[3]** PSTN Specific ✓ **[4]** Level 0 Regional (Local) ▪ If you selected E.164 Public as the NPI, you can select one of the following: ✓ **[0]** Unknown ✓ **[1]** International ✓ **[2]** National ✓ **[3]** Network Specific ✓ **[4]** Subscriber ✓ **[6]** Abbreviated
Phone Context CLI: context **[PhoneContext_Context]**	Defines the SIP 'phone-context' URI parameter.

27.8 Configuring Release Cause Mapping

The release cause mapping tables let you configure rules to map up to 12 different ISDN ITU-T Q.850 release cause codes (call failure) to SIP response codes, and vice versa. These tables allow you to override the default ISDN-SIP release cause mappings, described in "Fixed Mapping of ISDN Release Reason to SIP Response" on page 392 and "Fixed Mapping of SIP Response to ISDN Release Reason" on page 391.

The release cause mapping tables include:

- **Release Cause Mapping from SIP to ISDN table:** Maps SIP release codes to ISDN cause codes. When a SIP response is received from the IP side, the device searches this table for a match. If the SIP response is found, the Q.850 Release Cause assigned to it is sent to the PSTN. If no match is found, the default static mapping is used.
- **Release Cause Mapping from ISDN to SIP table:** Maps ISDN cause codes to SIP release codes. When a Release Cause is received from the PSTN, the device searches this table for a match. If the Q.850 Release Cause is found, the SIP response assigned to it is sent to the IP side. If no match is found, the default static mapping is used.

> **Note:** For Tel-to-IP calls, you can also map the less commonly used SIP responses to a single default ISDN Release Cause, using the DefaultCauseMapISDN2IP parameter. This parameter defines a default ISDN Cause that is always used except when the following Release Causes are received: Normal Call Clearing (16), User Busy (17), No User Responding (18) or No Answer from User (19).

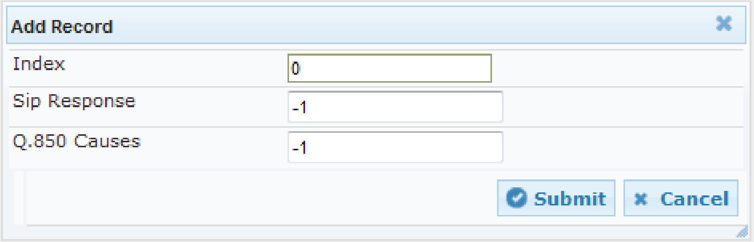
The following procedure describes how to configure ISDN-SIP release cause mapping in the Web interface. You can also configure this mapping using the following management platforms:

- Release Cause Mapping from ISDN to SIP table: ini file parameter, CauseMapISDN2SIP or CLI command, configure voip > gw manipulations cause-map-isdn2sip
- Release Cause Mapping from SIP to ISDN table:ini file parameter, CauseMapSIP2ISDN or CLI command, configure voip > gw manipulations cause-map-sip2isdn

➢ **To configure an ISDN-SIP release cause mapping rule:**

1. Open the required release cause mapping table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Release Cause SIP > ISDN** or **Release Cause ISDN > SIP**).
2. Click **Add**; the following dialog box appears:

Figure 27-6: Release Cause Mapping - Add Record



Add Record

Index	0
Sip Response	-1
Q.850 Causes	-1

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

Table 27-7: Release Cause Mapping Tables Parameter Descriptions

Parameter	Description
Index **[CauseMapSip2Isdn_Index]** **[CauseMapIsdn2Sip_Index]**	Defines an index number for the new table record.
Sip Response CLI: sip-response **[CauseMapSip2Isdn_SipResponse]** **[CauseMapIsdn2Sip_SipResponse]**	Defines the SIP response code (e.g., 406).
Q.850 Causes CLI: q850-causes **[CauseMapSip2Isdn_IsdnReleaseCause]** **[CauseMapIsdn2Sip_IsdnReleaseCause]**	Defines the ISDN Q.850 cause code (e.g., 6).

27.8.1 Fixed Mapping of SIP Response to ISDN Release Reason

The following table describes the mapping of SIP response to ISDN release reason.

Table 27-8: Mapping of SIP Response to ISDN Release Reason

SIP Response	Description	ISDN Release Reason	Description
400*	Bad request	31	Normal, unspecified
401	Unauthorized	21	Call rejected
402	Payment required	21	Call rejected
403	Forbidden	21	Call rejected
404	Not found	1	Unallocated number
405	Method not allowed	63	Service/option unavailable
406	Not acceptable	79	Service/option not implemented
407	Proxy authentication required	21	Call rejected
408	Request timeout	102	Recovery on timer expiry
409	Conflict	41	Temporary failure
410	Gone	22	Number changed w/o diagnostic
411	Length required	127	Interworking
413	Request entity too long	127	Interworking
414	Request URI too long	127	Interworking
415	Unsupported media type	79	Service/option not implemented
420	Bad extension	127	Interworking
480	Temporarily unavailable	18	No user responding
481*	Call leg/transaction doesn't exist	127	Interworking
482*	Loop detected	127	Interworking

SIP Response	Description	ISDN Release Reason	Description
483	Too many hops	127	Interworking
484	Address incomplete	28	Invalid number format
485	Ambiguous	1	Unallocated number
486	Busy here	17	User busy
488	Not acceptable here	31	Normal, unspecified
500	Server internal error	41	Temporary failure
501	Not implemented	38	Network out of order
502	Bad gateway	38	Network out of order
503	Service unavailable	41	Temporary failure
504	Server timeout	102	Recovery on timer expiry
505*	Version not supported	127	Interworking
600	Busy everywhere	17	User busy
603	Decline	21	Call rejected
604	Does not exist anywhere	1	Unallocated number
606*	Not acceptable	38	Network out of order

***** Messages and responses were created because the 'ISUP to SIP Mapping' draft does not specify their cause code mapping.

27.8.2 Fixed Mapping of ISDN Release Reason to SIP Response

The following table describes the mapping of ISDN release reason to SIP response.

Table 27-9: Mapping of ISDN Release Reason to SIP Response

ISDN Release Reason	Description	SIP Response	Description
1	Unallocated number	404	Not found
2	No route to network	404	Not found
3	No route to destination	404	Not found
6	Channel unacceptable	406*	Not acceptable
7	Call awarded and being delivered in an established channel	500	Server internal error
16	Normal call clearing	-*	BYE
17	User busy	486	Busy here
18	No user responding	408	Request timeout
19	No answer from the user	480	Temporarily unavailable
21	Call rejected	403	Forbidden
22	Number changed w/o diagnostic	410	Gone

ISDN Release Reason	Description	SIP Response	Description
26	Non-selected user clearing	404	Not found
27	Destination out of order	502	Bad gateway
28	Address incomplete	484	Address incomplete
29	Facility rejected	501	Not implemented
30	Response to status enquiry	501*	Not implemented
31	Normal unspecified	480	Temporarily unavailable
34	No circuit available	503	Service unavailable
38	Network out of order	503	Service unavailable
41	Temporary failure	503	Service unavailable
42	Switching equipment congestion	503	Service unavailable
43	Access information discarded	502*	Bad gateway
44	Requested channel not available	503*	Service unavailable
47	Resource unavailable	503	Service unavailable
49	QoS unavailable	503*	Service unavailable
50	Facility not subscribed	503*	Service unavailable
55	Incoming calls barred within CUG	403	Forbidden
57	Bearer capability not authorized	403	Forbidden
58	Bearer capability not presently available	503	Service unavailable
63	Service/option not available	503*	Service unavailable
65	Bearer capability not implemented	501	Not implemented
66	Channel type not implemented	480*	Temporarily unavailable
69	Requested facility not implemented	503*	Service unavailable
70	Only restricted digital information bearer capability is available	503*	Service unavailable
79	Service or option not implemented	501	Not implemented
81	Invalid call reference value	502*	Bad gateway
82	Identified channel does not exist	502*	Bad gateway
83	Suspended call exists, but this call identity does not	503*	Service unavailable
84	Call identity in use	503*	Service unavailable
85	No call suspended	503*	Service unavailable
86	Call having the requested call identity has been cleared	408*	Request timeout
87	User not member of CUG	503	Service unavailable
88	Incompatible destination	503	Service unavailable

ISDN Release Reason	Description	SIP Response	Description
91	Invalid transit network selection	502*	Bad gateway
95	Invalid message	503	Service unavailable
96	Mandatory information element is missing	409*	Conflict
97	Message type non-existent or not implemented	480*	Temporarily not available
98	Message not compatible with call state or message type non-existent or not implemented	409*	Conflict
99	Information element non-existent or not implemented	480*	Not found
100	Invalid information elements contents	501*	Not implemented
101	Message not compatible with call state	503*	Service unavailable
102	Recovery of timer expiry	408	Request timeout
111	Protocol error	500	Server internal error
127	Interworking unspecified	500	Server internal error

***** Messages and responses were created because the ‘ISUP to SIP Mapping’ draft doesn’t specify their cause code mapping.

27.8.3 Reason Header

The device supports the SIP Reason header according to RFC 3326. The Reason header conveys information describing the disconnection cause of a call:

- **Sending Reason header:** If a call is disconnected from the Tel side (ISDN), the Reason header is set to the received Q.850 cause in the appropriate message (BYE/CANCEL/final failure response) and sent to the SIP side. If the call is disconnected because of a SIP reason, the Reason header is set to the appropriate SIP response.
- **Receiving Reason header:** If a call is disconnected from the IP side and the SIP message includes the Reason header, it is sent to the Tel side according to the following logic:
    - If the Reason header includes a Q.850 cause, it is sent as is.
    - If the Reason header includes a SIP response:
        - If the message is a final response, the response status code is translated to Q.850 format and passed to ISDN.
        - If the message isn’t a final response, it is translated to a Q.850 cause.
    - When the Reason header is received twice (i.e., SIP Reason and Q.850), the Q.850 takes precedence over the SIP reason and is sent to the Tel side.

27.8.4 Mapping PSTN Release Cause to SIP Response

The device's FXO interface interoperates between the SIP network and the PSTN/PBX. This interoperability includes the mapping of PSTN/PBX Call Progress tones to SIP 4xx or 5xx responses for IP-to-Tel calls. The converse is also true - for Tel-to-IP calls, the SIP 4xx or 5xx responses are mapped to tones played to the PSTN/PBX.

When establishing an IP-to-Tel call, the following rules are applied:

- If the remote party (PSTN/PBX) is busy and the FXO device detects a busy tone, it sends a SIP 486 Busy response to IP. If it detects a reorder tone, it sends a SIP 404 Not Found (no route to destination) to IP. In both cases the call is released. Note that if the 'Disconnect Call on Busy Tone Detection' parameter is set to **Disable**, the FXO device ignores the detection of busy and reorder tones and does not release the call.
- For all other FXS/FXO release types such as:
  - no free channels in the Trunk Group,
  - an appropriate call routing rule to a Trunk Group doesn’t exist, or
  - the phone number isn’t found

  then the device sends a SIP response to the IP according to the 'Default Release Cause' parameter. This parameter defines Q.931 release causes. Its default value is **3**, which is mapped to the SIP 404 response. By changing its value to **34**, the SIP 503 response is sent. Other causes can be used as well.

27.9 Numbering Plans and Type of Number

The IP-to-Tel destination or source number manipulation tables allow you to classify numbers by their Numbering Plan Indication (NPI) and Type of Number (TON). The device supports all NPI/TON classifications used in the ETSI ISDN variant, as shown in the table below:

Table 27-10: NPI/TON Values for ETSI ISDN Variant

NPI	TON	Description
Unknown [0]	Unknown [0]	A valid classification, but one that has no information about the numbering plan.
E.164 Public [1]	Unknown [0]	A public number in E.164 format, but no information on what kind of E.164 number.
	International [1]	A public number in complete international E.164 format, e.g., 16135551234.
	National [2]	A public number in complete national E.164 format, e.g., 6135551234.
	Network Specific [3]	The type of number "network specific number" is used to indicate administration / service number specific to the serving network, e.g., used to access an operator.
	Subscriber [4]	A public number in complete E.164 format representing a local subscriber, e.g., 5551234.
	Abbreviated [6]	The support of this code is network dependent. The number provided in this information element presents a shorthand representation of the complete number in the specified numbering plan as supported by the network.

NPI	TON	Description
Private [9]	Unknown [0]	A private number, but with no further information about the numbering plan.
	Level 2 Regional [1]	
	Level 1 Regional [2]	A private number with a location, e.g., 3932200.
	PISN Specific [3]	
	Level 0 Regional (local) [4]	A private local extension number, e.g., 2200.

For NI-2 and DMS-100 ISDN variants, the valid combinations of TON and NPI for calling and called numbers include (Plan/Type):

- 0/0 - Unknown/Unknown
- 1/1 - International number in ISDN/Telephony numbering plan
- 1/2 - National number in ISDN/Telephony numbering plan
- 1/4 - Subscriber (local) number in ISDN/Telephony numbering plan
- 9/4 - Subscriber (local) number in Private numbering plan

28 Routing

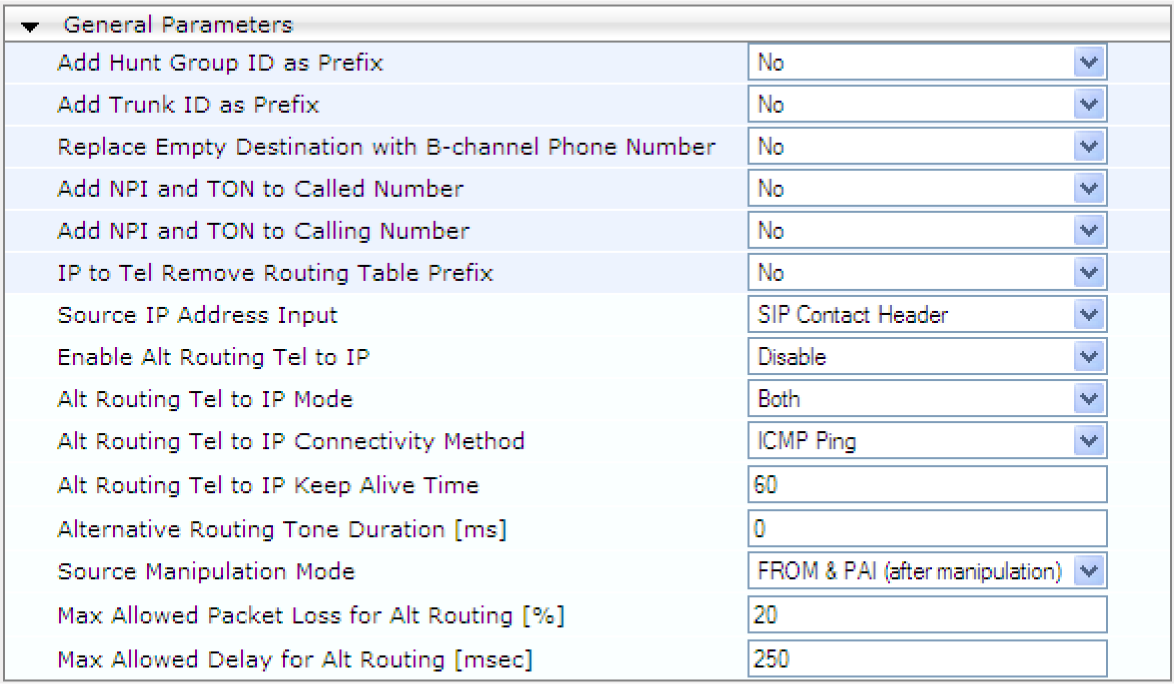
This section describes the configuration of call routing rules.

28.1 Configuring General Routing Parameters

The Routing General Parameters page allows you to configure general routing parameters. For a description of these parameters, see "Configuration Parameters Reference" on page 763.

➢ **To configure general routing parameters:**

1. Open the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **General Parameters**).



General Parameters

Add Hunt Group ID as Prefix	No
Add Trunk ID as Prefix	No
Replace Empty Destination with B-channel Phone Number	No
Add NPI and TON to Called Number	No
Add NPI and TON to Calling Number	No
IP to Tel Remove Routing Table Prefix	No
Source IP Address Input	SIP Contact Header
Enable Alt Routing Tel to IP	Disable
Alt Routing Tel to IP Mode	Both
Alt Routing Tel to IP Connectivity Method	ICMP Ping
Alt Routing Tel to IP Keep Alive Time	60
Alternative Routing Tone Duration [ms]	0
Source Manipulation Mode	FROM & PAI (after manipulation)
Max Allowed Packet Loss for Alt Routing [%]	20
Max Allowed Delay for Alt Routing [msec]	250

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

28.2 Configuring Outbound IP Routing Table

The Outbound IP Routing table lets you configure up to 180 Tel-to-IP or outbound IP call routing rules. The device uses these rules to route calls from the Tel or IP to a user-defined IP destination. You can also configure the device to process the routing rule before or after it has performed number manipulation, if necessary.

The Outbound IP Routing table provides two configuration areas:

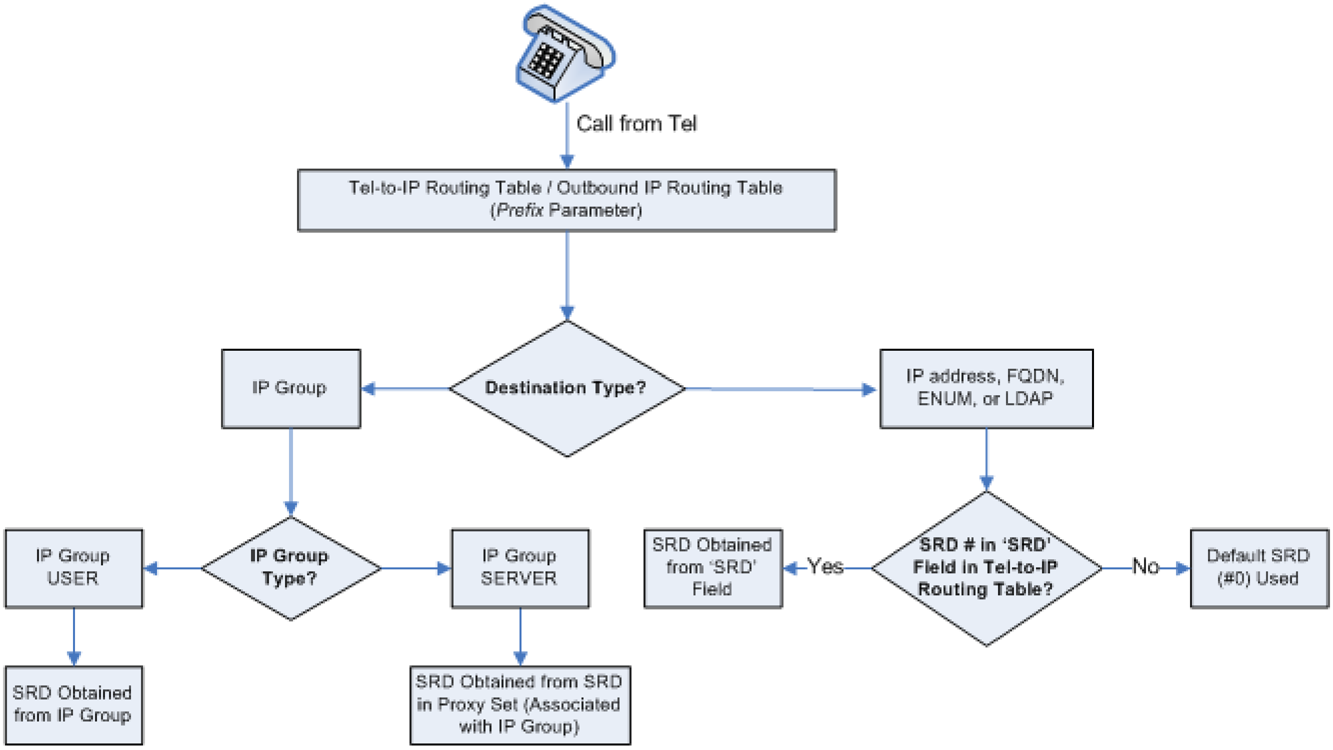
- **Matching Characteristics:** Characteristics of the incoming call. If the call characteristics match a table entry, the routing rule is used to route the call to the specified destination. One or more characteristics can be defined for the rule:
  - Source IP Group (to which the call belongs)
  - Source and destination Request-URI host name prefix
  - Source Trunk Group (from where the call is received)
  - Source (calling) and destination (called) telephone number prefix and suffix

- Source and destination Request-URI host name prefix

- **Destination:** If the call matches the configured characteristics, the device routes the call to an IP destination. If no characteristics match is found in the table, the call is rejected. The destination can be any of the following:
  - IP address in dotted-decimal notation.
  - Fully Qualified Domain Name (FQDN).
  - E.164 Telephone Number Mapping (ENUM service).
  - Lightweight Directory Access Protocol (LDAP). For a description, see Routing Based on LDAP Active Directory Queries on page 210.
  - IP Group, where the call is routed to the IP address configured for the Proxy Set or SRD associated with the IP Group (configured in "Configuring IP Groups" on page 270). If the device is configured with multiple SRDs, you can also indicate (in the table's 'Dest. SRD' field) the destination SRD for routing to one of the following destination types - IP address, FQDN, ENUM, or LDAP. If the SRD is not specified, then the default SRD (0) is used. In scenarios where routing is to an IP Group, the destination SRD is obtained from the SRD associated with the IP Group (in the IP Group table). The specified destination SRD determines the:
    - Destination SIP interface (SIP port and control IP interface) - important when using multiple SIP control VLANs
    - Media Realm (port and IP interface for media / RTP voice)
    - Other SRD-related interfaces and features on which the call is routed

  Since each call must have a destination IP Group (even in cases where the destination type is not to an IP Group), in cases when the IP Group is not specified, the SRD's default IP Group is used, which is the first configured IP Group that belongs to the SRD.

Figure 28-1: Locating SRD


Call from Tel
Tel-to-IP Routing Table / Outbound IP Routing Table (Prefix Parameter)
Destination Type?
IP Group
IP address, FQDN, ENUM, or LDAP
IP Group Type?
IP Group USER
IP Group SERVER
SRD Obtained from IP Group
SRD Obtained from SRD in Proxy Set (Associated with IP Group)
SRD # in 'SRD' Field in Tel-to-IP Routing Table?
Yes
No
SRD Obtained from 'SRD' Field
Default SRD (#0) Used

Note: When using a proxy server, you do not need to configure this table, unless you require one of the following:

- Fallback (alternative) routing if communication is lost with the proxy server.
- IP security, whereby the device routes only received calls whose source IP addresses are defined in this table. IP security is enabled using the SecureCallsFromIP parameter.
- Filter Calls to IP feature: the device checks this table before a call is routed to the proxy server. However, if the number is not allowed, i.e., the number does not exist in the table or a Call Restriction (see below) routing rule is applied, the call is released.
- Obtain different SIP URI host names (per called number).
- Assign IP Profiles to calls.
- For this table to take precedence over a proxy for routing calls, you need to set the parameter PreferRouteTable to 1. The device checks the 'Destination IP Address' field in this table for a match with the outgoing call; a proxy is used only if a match is not found.

In addition to basic outbound IP routing, this table supports the following features:

- **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 234. If two routing rules have identical costs, then the rule appearing higher up in the table (i.e., first-matched rule) is used. If a selected route is unavailable, the device uses the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched routing rules with Cost Groups, according to the settings of the LCR parameter, LCRDefaultCost (see "Enabling LCR and Configuring Default LCR" on page 236).
- **Call Forking:** If the Tel-to-IP Call Forking feature is enabled, the device can send a Tel call to multiple IP destinations. An incoming Tel call with multiple matched routing rules (e.g., all with the same source prefix numbers) can be sent (forked) to multiple IP destinations if all these rules are configured with a Forking Group. The call is established with the first IP destination that answers the call.
- **Call Restriction:** Rejects calls whose matching routing rule is configured with the destination IP address of 0.0.0.0.
- **Always Use Routing Table:** Even if a proxy server is used, the SIP Request-URI host name in the outgoing INVITE message is obtained from this table. Using this feature, you can assign a different SIP URI host name for different called and/or calling numbers. This feature is enabled using the AlwaysUseRouteTable parameter.
- **IP Profiles:** IP Profiles can be assigned to destination addresses (also when a proxy is used).
- **Alternative Routing (when a proxy isn't used):** An alternative IP destination (alternative routing rule) can be configured for specific calls ("main" routing rule). When the "main" route fails (e.g., busy), the device can send the call to the alternative route. You must configure the alternative routing rules in table rows (indices) that are located anywhere below the "main" routing rule. For example, if you configure a "main" routing rule in Index 4, the alternative routing rule can be configured in Index 6. In addition, you must configure the alternative routing rules with identical matching characteristics (e.g., destination prefix number) to the "main" routing rule, but assigned with different destination IP addresses. Instead of an IP address, you can use an FQDN to resolve into two IP addresses. For more information on alternative routing, see "Alternative Routing for Tel-to-IP Calls" on page 412.
- **Advice of Charge (AOC):** AOC is a pre-billing feature that tasks the rating engine

with calculating the cost of using a service (Tel-to-IP call) and relaying that information to the customer. AOC, which is configured in the Charge Codes table, can be applied per Tel-to-IP routing rule.

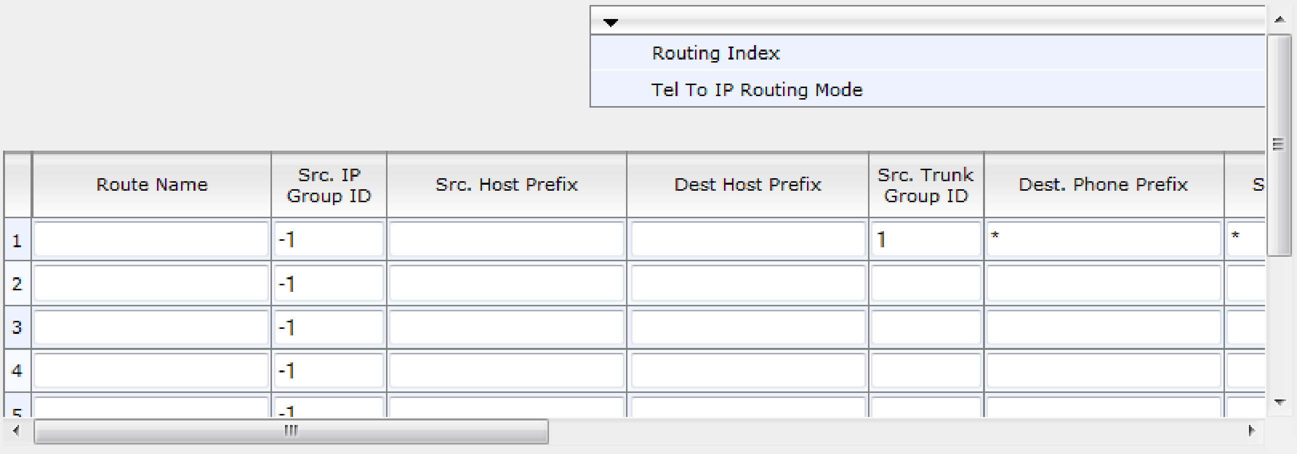
Note: You can configure up to three alternative routing rules per "main" routing rule in the Inbound IP Routing table.

The following procedure describes how to configure Tel-to-IP or outbound IP routing rules in the Web interface. You can also configure these rules using the table ini file parameter, Prefix or CLI command, configure voip > gw routing tel2ip-routing.

➢ To configure Tel-to-IP or outbound IP routing rules:

1. Open the Outbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **Tel to IP Routing**).

Figure 28-2: Outbound IP Routing Page



	Route Name	Src. IP Group ID	Src. Host Prefix	Dest Host Prefix	Src. Trunk Group ID	Dest. Phone Prefix	S
1		-1			1	*	*
2		-1					
3		-1					
4		-1					
5		-1					

2. From the 'Routing Index' drop-down list, select the range of entries that you want to add.
3. Configure a routing rule according to the parameters described in the table below.
4. Click **Submit**, and then save ("burn") your settings to flash memory.

The table below shows configuration examples of Tel-to-IP or outbound IP routing rules, where:

- **Rule 1 and 2 (Least Cost Routing rule):** For both rules, the called (destination) phone number prefix is 10, the caller's (source) phone number prefix is 100, and the call is assigned IP Profile ID 1. However, Rule 1 is assigned a cheaper Cost Group than Rule 2, and therefore, the call is sent to the destination IP address (10.33.45.63) associated with Rule 1.
- **Rule 3 (IP Group destination rule):** For all callers (*), if the called phone number prefix is 20, the call is sent to IP Group 1 (whose destination is the IP address configured for its associated Proxy Set ID).
- **Rule 4 (domain name destination rule):** If the called phone number prefix is 5, 7, 8, or 9 and the caller belongs to Trunk Group ID 1, the call is sent to domain.com.
- **Rule 5 (block rule):** For all callers (*), if the called phone number prefix is 00, the call is rejected (discarded).
- **Rule 6, 7, and 8 (Forking Group rule):** For all callers (*), if the called phone number prefix is 100, the call is sent to Rule 7 and 9 (belonging to Forking Group "1"). If their destinations are unavailable and alternative routing is enabled, the call is sent to Rule 8 (Forking Group "2").

- Rule 9 (IP-to-IP rule): If an incoming IP call from Source IP Group 2 with domain.com as source host prefix in its SIP Request-URI, the IP call is sent to IP address 10.33.45.65.

Table 28-1: Example of Tel-to-IP Source Phone Number Manipulation Rules

Parameter	Rule 1	Rule 2	Rule 3	Rule 4	Rule 5	Rule 6	Rule 7	Rule 8	**Rule 9**
Src. Trunk Group ID	*	0	1	-	-	-	-	-	-
Src. Trunk Group ID	-	-	*	1	-	*	*	*	-
Dest. Phone Prefix	10	10	20	[5,7-9]	00	100	100	100	*
Source Phone Prefix	100	100	*	*	*	*	*	*	*
Dest. IP Address	10.33.45.63	10.33.45.50	-	domain.com	0.0.0.0	10.33.45.68	10.33.45.67	domain.com	10.33.45.65
Dest IP Group ID	-	-	1	-	-	-	-	-	-
IP Profile ID	1	1	-	-	-	-	-	-	-
Cost Group ID	Weekend	Weekend_B	-	-	-	-	-	-	-
Forking Group			-	-	-	1	2	1	-

Table 28-2: Outbound IP Routing Table Parameter Descriptions

Parameter	Description
Index **[PREFIX_Index]**	Defines an index number for the new table record.
Web/EMS: Tel to IP Routing Mode CLI: configure voip > gw routing general-setting > tel2ip-rte-mode **[RouteModeTel2IP]**	Determines whether to route received calls to an IP destination before or after manipulation of the destination number. ▪ **[0]** Route calls before manipulation = Calls are routed before the number manipulation rules are applied (default). ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied. **Notes:** ▪ This parameter is not applicable if outbound proxy routing is

Parameter	Description
	used. - For number manipulation, see "Configuring Source/Destination Number Manipulation" on page 373.
Route Name CLI: route-name **[PREFIX_RouteName]**	Defines an arbitrary name to easily identify the routing rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Call Characteristics	
Web: Src. IP Group ID EMS: Source IP Group ID CLI: src-ip-group-id [PREFIX_SrcIPGroupID]	Defines the IP Group from where the incoming IP call is received. Typically, the IP Group of an incoming INVITE is determined according to the Inbound IP Routing table. **Notes:** - This parameter is applicable only to the Gateway's IP-to-IP routing application. - To denote all IP Groups, leave this field empty. - If this IP Group has a Serving IP Group (configured in the IP Group table), calls from the IP Group are sent to the Serving IP Group and therefore, this routing rule is ignored. However, if you want the call to be sent to a destination configured for this routing rule instead, set the PreferRouteTable parameter to 1.
Web: Src. Host Prefix EMS: Source Host Prefix CLI: src-host-prefix [PREFIX_SrcHostPrefix]	Defines the prefix of the SIP Request-URI host name in the From header of the incoming SIP INVITE message. To denote any prefix, use the asterisk (*) symbol. **Note:** This parameter is applicable only to the Gateway's IP-to-IP routing application.
Web: Dest. Host Prefix EMS: Destination Host Prefix CLI: dst-host-prefix [PREFIX_DestHostPrefix]	Defines the SIP Request-URI host name prefix of the incoming SIP INVITE message. To denote any prefix, use the asterisk (*) symbol. **Note:** This parameter is applicable only to the Gateway's IP-to-IP routing application.
Web: Src. Trunk Group ID EMS: Source Trunk Group ID CLI: src-trunk-group-id **[PREFIX_SrcTrunkGroupID]**	Defines the Trunk Group from where the call is received. To denote any Trunk Group, use the asterisk (*) symbol.
Web: Dest. Phone Prefix EMS: Destination Phone Prefix CLI: dst-phone-prefix **[PREFIX_DestinationPrefix]**	Defines the prefix and/or suffix of the called (destination) telephone number. The suffix is enclosed in parenthesis after the suffix value. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a called number, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761. The number can include up to 50 digits. **Notes:** - For LDAP-based routing, enter the LDAP query keyword as the prefix number to denote the IP domain: ✓ "PRIVATE" = Private number ✓ "OCS" = Lync / OCS client number ✓ "PBX" = PBX / IP PBX number ✓ "MOBILE" = Mobile number

Parameter	Description
	✓ "LDAP_ERR" = LDAP query failure For more information, see Routing Based on LDAP Active Directory Queries on page 210. ▪ If you want to configure re-routing of ISDN Tel-to-IP calls to fax destinations, you need to enter the value string "FAX" (case-sensitive) as the destination phone prefix. For more information regarding this feature, see the FaxReroutingMode parameter.
Web/EMS: Source Phone Prefix CLI: src-phone-prefix **[PREFIX_SourcePrefix]**	Defines the prefix and/or suffix of the calling (source) telephone number. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a calling number, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761. The number can include up to 50 digits.
Call Setup Rules Set ID CLI: call-setup-rules-set-id **[PREFIX_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 241.
Operation (IP Destination)	
Web: Dest. IP Address EMS: Address CLI: dst-ip-address **[PREFIX_DestAddress]**	Defines the IP address (in dotted-decimal notation or FQDN) to where the call is sent. If an FQDN is used (e.g., domain.com), DNS resolution is done according to the DNSQueryType parameter. The IP address can include the following wildcards: ▪ "x": represents single digits. For example, 10.8.8.xx denotes all addresses between 10.8.8.10 and 10.8.8.99. ▪ "*": represents any number between 0 and 255. For example, 10.8.8.* denotes all addresses between 10.8.8.0 and 10.8.8.255. For ENUM-based routing, enter the string "ENUM". The device sends an ENUM query containing the destination phone number to an external DNS server, configured in the Interface table. The ENUM reply includes a SIP URI which is used as the Request-URI in the subsequent outgoing INVITE and for routing (if a proxy is not used). To configure the type of ENUM service (e.g., e164.arpa), use the EnumService parameter. For LDAP-based routing, enter the string value "LDAP" for denoting the IP address of the LDAP server. For more information, see Routing Based on LDAP Active Directory Queries on page 210. **Notes:** ▪ This field and any value assigned to it is ignored if you have configured a destination IP Group for this routing rule (in the 'Dest IP Group ID' field). ▪ To reject calls, enter the IP address 0.0.0.0. For example, if you want to prohibit international calls, then in the 'Dest Phone

Parameter	Description
	Prefix' field, enter 00 and in the 'Dest IP Address' field, enter 0.0.0.0. - For routing calls between phones connected to the device (i.e., local routing), enter the device's IP address. - When the device's IP address is unknown (e.g., when DHCP is used), enter IP address 127.0.0.1. - When using domain names, enter the DNS server's IP address or alternatively, configure these names in the Internal DNS table (see "Configuring the Internal DNS Table" on page 144).
Web: Port EMS: Destination Port CLI: dst-port **[PREFIX_DestPort]**	Defines the destination port to where you want to route the call.
Web/EMS: Transport Type CLI: transport-type **[PREFIX_TransportType]**	Defines the transport layer type for sending the IP call: - **[-1]** Not Configured - **[0]** UDP - **[1]** TCP - **[2]** TLS **Note:** When set to Not Configured (-1), the transport type defined by the SIPTransportType parameter is used.
Web: Dest. IP Group ID EMS: Destination IP Group ID CLI: dst-ip-group-id **[PREFIX_DestIPGroupID]**	Defines the IP Group to where you want to route the call. The SIP INVITE message is sent to the IP address defined for the Proxy Set ID associated with the IP Group. **Notes:** - If you select an IP Group, you do not need to configure a destination IP address. However, if both parameters are configured in this table, the INVITE message is sent only to the IP Group (and not the defined IP address). - If the source IP Group, configured in the 'Src. IP Group ID' parameter above, has a Serving IP Group (configured in the IP Group table), calls from the IP Group are sent to the Serving IP Group and therefore, this routing rule is ignored. However, if you want this routing rule to be used instead and sent to this destination IP Group, set the PreferRouteTable parameter to 1. - If the destination is a User-type IP Group, the device searches for a match between the Request-URI (of the received INVITE) to an AOR registration record in the device's database. The INVITE is then sent to the IP address of the registered contact. - If the parameter AlwaysUseRouteTable is set to 1 (see "Configuring IP Groups" on page 270), then the Request-URI host name in the INVITE message is set to the value defined for the parameter 'Dest. IP Address' (above); otherwise, if no IP address is defined, it is set to the value of the parameter 'SIP Group Name' (defined in the IP Group table). - This parameter is used as the 'Serving IP Group' in the Account table for acquiring authentication user/password for this call (see "Configuring Registration Accounts" on page 291). - For defining Proxy Set ID's, see "Configuring Proxy Sets" on page 282.
Dest SRD CLI: dst-srd [PREFIX_DestSRD]	Defines the SRD to where you want to route the call. The actual destination is defined by the Proxy Set associated with the SRD. This allows you to route the call to a specific SIP Media Realm and

Parameter	Description
	SIP Interface. To configure SRD's, see Configuring SRDs on page 264.
IP Profile ID CLI: ip-profile-id **[PREFIX_ProfileId]**	Assigns an IP Profile ID to this IP destination call. This allows you to assign numerous configuration attributes (e.g., voice codes) per routing rule. To configure IP Profiles, see "Configuring IP Profiles" on page 314.
Status	(Read-only field) Displays the connectivity status of the routing rule's IP destination. If there is connectivity with the destination, this field displays "OK" and the device uses this routing rule if required. The routing rule is not used if any of the following is displayed: ▪ "n/a" = The destination IP Group is unavailable ▪ "No Connectivity" = No connection with the destination (no response to the SIP OPTIONS). ▪ "QoS Low" = Poor Quality of Service (QoS) of the destination. ▪ "DNS Error" = No DNS resolution. This status is applicable only when a domain name is used (instead of an IP address). ▪ "Unavailable" = The destination is unreachable due to networking issues.
Web/EMS: Charge Code CLI: charge-code **[PREFIX_MeteringCode]**	Assigns a Charge Code to the routing rule. To configure Charge Codes, see "Configuring Charge Codes Table" on page 470. **Note:** This parameter is applicable only to FXS interfaces and E1 Euro ISDN trunks.
Cost Group ID CLI: cost-group-id **[PREFIX_CostGroup]**	Assigns a Cost Group with the routing rule for determining the cost of the call. To configure Cost Groups, see "Configuring Cost Groups" on page 238.
Forking Group CLI: forking-group **[PREFIX_ForkingGroup]**	Defines a forking group ID for the routing rule. This enables forking of incoming Tel calls to multiple IP destinations. The device sends simultaneous INVITE messages and handles multiple SIP dialogs until one of the calls is answered. When a call is answered, the other calls are dropped. If all matched routing rules belong to the same Forking Group number, the device sends an INVITE to all the destinations belonging to this group and according to the following logic: ▪ If matched routing rules belong to different Forking Groups, the device sends the call to the Forking Group of the first matched routing rule. If the call cannot be established with any of the destinations associated with this Forking Group and alternative routing is enabled, the device forks the call to the Forking Group of the next matched routing rules as long as the Forking Group is defined with a **higher** number than the previous Forking Group. For example: ▪ Table index entries 1 and 2 are defined with Forking Group "1", and index entries 3 and 4 with Forking Group "2": The device first sends the call according to index entries 1 and 2, and if unavailable and alternative routing is enabled, sends the call according to index entries 3 and 4. ▪ Table index entry 1 is defined with Forking Group "2", and index entries 2, 3, and 4 with Forking Group "1": The device sends the call according to index entry 1 only and ignores the other index entries even if the destination is unavailable and

Parameter	Description
	alternative routing is enabled. This is because the subsequent index entries are defined with a Forking Group number that is lower than that of index entry 1. ▪ Table index entry 1 is defined with Forking Group "1", index entry 2 with Forking Group "2", and index entries 3 and 4 with Forking Group "1": The device first sends the call according to index entries 1, 3, and 4 (all belonging to Forking Group "1"), and if the destination is unavailable and alternative routing is enabled, the device sends the call according to index entry 2. ▪ Table index entry 1 is defined with Forking Group "1", index entry 2 with Forking Group "3", index entry 3 with Forking Group "2", and index entry 4 with Forking Group "1": The device first sends the call according to index entries 1 and 4 (all belonging to Forking Group "1"), and if the destination is unavailable and alternative routing is enabled, the device sends the call according to index entry 2 (Forking Group "3"). Even if index entry 2 is unavailable and alternative routing is enabled, the device ignores index entry 3 because it belongs to a Forking Group that is lower than index entry 2. **Notes:** ▪ To enable Tel-to-IP call forking, set the 'Tel2IP Call Forking Mode' (*Tel2IPCallForkingMode*) parameter to **Enable**. ▪ You can configure the device to immediately send the INVITE message to the first member of the Forking Group (as in normal operation) and then only after a user-defined interval, send (simultaneously) the INVITE messages to the other members. If the device receives a SIP 4xx or 5xx in response to the first INVITE, it immediately sends INVITEs to all the other members, regardless of the interval. To configure this feature, use the ForkingDelayTimeForInvite ini file parameter. ▪ You can implement Forking Groups when the destination is an LDAP server or a domain name using DNS. In such scenarios, the INVITE is sent to all the queried LDAP or resolved IP addresses, respectively. You can also use LDAP routing rules with standard routing rules for Forking Groups. ▪ When the UseDifferentRTPportAfterHold parameter is enabled, every forked call is sent with a different RTP port. Thus, ensure that the device has sufficient available RTP ports for these forked calls.

28.3 Configuring Inbound IP Routing Table

The Inbound IP Routing table lets you configure up to 120 inbound call routing rules:

- For IP-to-IP routing: The table is used to identify an incoming call as an IP-to-IP call and subsequently, to assign the call to an IP Group, referred to as a source IP Group. These IP-to-IP calls can later be routed to an outbound destination IP Group (see Configuring Outbound IP Routing Table on page 397).
- For IP-to-Tel routing: This table is used to route incoming IP calls to Trunk Groups. The specific channel pertaining to the Trunk Group to which the call is routed is determined according to the Trunk Group's channel selection mode. The channel selection mode can be defined per Trunk Group (see "Configuring Hunt Group Settings" on page 367) or for all Trunk Groups using the global parameter ChannelSelectMode.

The Inbound IP Routing table provides two configuration areas:

- Matching characteristics of incoming IP call, for example, prefix of destination number.
- Operation (destination), for example, sends to a specific Trunk Group.

If the incoming call matches the characteristics of a rule, the call is sent to the destination configured for that rule.

The device also supports alternative routing if the IP-to-Tel call cannot be routed to the Trunk Group:

- **Routing to an Alternative Trunk Group:** If a call release reason (cause) code (e.g., 17 for User Busy) is received from the Tel side for a specific IP-to-Tel call and you have configured this reason code in the Reasons for IP-to-Tel Alternative Routing table, the device re-routes the call to an alternative Trunk Group if an alternative routing rule has been configured. You must configure the alternative routing rules in table rows (indices) that are located anywhere below the "main" routing rule. For example, if you configure a "main" routing rule in Index 4, the alternative routing rule can be configured in Index 6. In addition, you must configure the alternative routing rules with identical matching characteristics (e.g., destination prefix number) to the "main" routing rule, but assigned with different destinations (i.e., Trunk Groups). For more information on IP-to-Tel alternative routing and for configuring call release reasons for alternative routing, see "Alternative Routing to Trunk upon Q.931 Call Release Cause Code" on page 417.
- **Routing to an IP Destination (i.e., Call Redirection):** The device can re-route the IP-to-Tel call to an alternative IP destination, using SIP 3xx responses. For more information, see "Alternative Routing to IP Destinations upon Busy Trunk" on page 418.
- **Routing to an Alternative Physical FXO port or Trunk Within the Same Trunk Group:** The device can re-route an IP-to-Tel call to a different physical FXO port or trunk if the destined trunk or FXO port within the same Trunk Group is out of service (e.g., physically disconnected). When the physical FXO port or trunk is disconnected, the device sends the SNMP trap, GWAPP_TRAP_BUSYOUT_LINK notifying of the out-of-service state for the specific trunk number. When the physical FXO port or trunk is physically reconnected, this trap is sent notifying of the back-to-service state.

Notes:

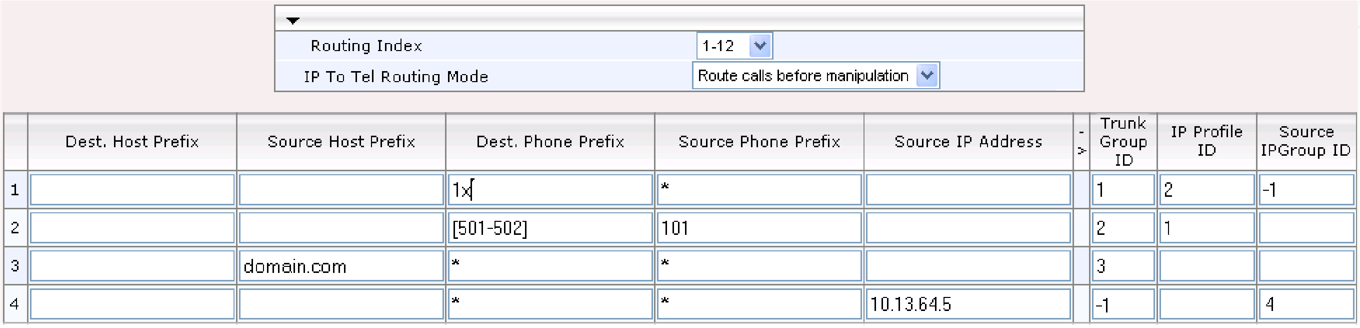
- You can configure up to three alternative routing rules per "main" routing rule in the Inbound IP Routing table.
- If your deployment includes calls of many different called (source) and/or calling (destination) numbers that need to be routed to the same destination, you can employ user-defined prefix tags to represent these numbers. Thus, instead of configuring many routing rules, you need to configure only one routing rule using the prefix tag as the source and destination number matching characteristics, and a destination for the calls. For more information on prefix tags, see "Dial Plan Prefix Tags for IP-to-Tel Routing" on page 608.

The following procedure describes how to configure Inbound IP Routing rules in the Web interface. You can also configure Inbound IP Routing rules using the table ini file parameter, PSTNPrefix or CLI command, configure voip > gw routing ip2tel-routing.

➢ To configure IP-to-Tel or inbound IP routing rules:

1. Open the Inbound IP Routing Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **IP to** Trunk **Group Routing**).

Figure 28-3: Inbound IP Routing Table



Routing Index: 1-12

IP To Tel Routing Mode: Route calls before manipulation

	Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	->	Trunk Group ID	IP Profile ID	Source IPGroup ID
1			1x	*			1	2	-1
2			[501-502]	101			2	1	
3		domain.com	*	*			3		
4			*	*	10.13.64.5		-1		4

The previous figure displays the following configured routing rules:

- **Rule 1:** If the incoming IP call destination phone prefix is between 10 and 19, the call is assigned settings configured for IP Profile ID 2 and routed to Trunk Group ID 1.
- **Rule 2:** If the incoming IP call destination phone prefix is between 501 and 502 and source phone prefix is 101, the call is assigned settings configured for IP Profile ID 1 and routed to Trunk Group ID 2.
- **Rule 3:** If the incoming IP call has a From URI host prefix as domain.com, the call is routed to Trunk Group ID 3.
- Rule 4: If the incoming IP call has IP address 10.13.64.5 in the INVITE's Contact header, the call is identified as an IP-to-IP call and assigned to Source IP Group 4. This call is routed according to the outbound IP routing rules for this Source IP Group configured in the Outbound IP Routing table.

2. Configure a routing rule according to the parameters described in the table below.
3. Click **Submit**.

Table 28-3: IP-to-Tel or Inbound IP Routing Table Parameter Description

Parameter	Description
IP to Tel Routing Mode CLI: configure voip/gw routing general-setting/ip2tel-rte-mode **[RouteModeIP2Tel]**	Determines whether to route the incoming IP call before or after manipulation of destination number, configured in "Configuring Source/Destination Number Manipulation" on page 373. ▪ **[0]** Route calls before manipulation = (Default) Incoming IP calls are routed before number manipulation. ▪ **[1]** Route calls after manipulation = Incoming IP calls are routed after number manipulation.
Index **[PstnPrefix_Index]**	Defines an index number for the new table record.
Route Name CLI: route-name **[PstnPrefix_RouteName]**	Defines an arbitrary name to easily identify the routing rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Characteristics	
Web: Dest. Host Prefix CLI: dst-phone-prefix **[PstnPrefix_DestPrefix]**	Defines the Request-URI host name prefix of the incoming SIP INVITE message. By default, no value is defined (i.e., not used in routing rule). To denote any prefix, use the asterisk (*) wildcard.

Parameter	Description
Web: Source Host Prefix CLI: src-host-prefix **[PstnPrefix_SrcHostPrefix]**	Defines the prefix of the URI host name in the From header of the incoming SIP INVITE message. By default, no value is defined (i.e., not used in routing rule). To denote any prefix, use the asterisk (*) wildcard. **Note:** If the P-Asserted-Identity header is present in the incoming INVITE message, the value of this parameter is compared to the P-Asserted-Identity URI host name (and not the From header).
Web: Dest. Phone Prefix CLI: dst-host-prefix **[PstnPrefix_DestHostPrefix]**	Defines the prefix or suffix of the called (destined) telephone number. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol or to denote calls without a called number, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761. The prefix can include up to 49 digits.
Web: Source Phone Prefix CLI: src-phone-prefix **[PstnPrefix_SourcePrefix]**	Defines the prefix or suffix of the calling (source) telephone number. The prefix can include up to 49 digits. You can use special notations for denoting the prefix. For example, [100-199](100,101,105) denotes a number that starts with 100 to 199 and ends with 100, 101 or 105. To denote any prefix, use the asterisk (*) symbol. To denote calls without a calling number, use the $ sign. For a description of available notations, see "Dialing Plan Notation for Routing and Manipulation Tables" on page 761. **Note:** If the P-Asserted-Identity header is present in the incoming INVITE message, the value of this parameter is compared to the P-Asserted-Identity URI host name (and not the From header).
Web: Source IP Address CLI: src-ip-address **[PstnPrefix_SourceAddress]**	Defines the source IP address of the incoming IP call that can be used for routing decisions. The IP address can be configured in dotted-decimal notation (e.g., 10.8.8.5) or as an FQDN. If the address is an FQDN, DNS resolution is done according to the DNSQueryType parameter. **Notes:** ▪ The source IP address is obtained from the Contact header in the INVITE message. ▪ You can configure from where the source IP address is obtained, using the SourceIPAddressInput parameter. ▪ The source IP address can include the following wildcards: ✓ "x": denotes single digits. For example, 10.8.8.xx represents all the addresses between 10.8.8.10 and 10.8.8.99. ✓ "*": denotes any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255.
Web: Source SRD ID CLI: src-srd-id	Defines the SRD from where the incoming packet is received. **Note:** When the incoming INVITE matches the SRD in the

Parameter	Description
[PstnPrefix_SrcSRDID]	routing rule, if the 'Source IP Group ID' parameter (see below) is defined and it is associated with a different SRD, the incoming SIP call is rejected. If the 'Source IP Group ID' parameter is not defined, the SRD's default IP Group is used. If there is no valid source IP Group, the call is rejected.
Call Setup Rules Set ID CLI: call-setup-rules-set-id **[PstnPrefix_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 241.
Operation (Destination)	
Web: Trunk Group ID CLI: trunk-group-id **[PstnPrefix_TrunkGroupId]**	For IP-to-Tel calls: Defines the Trunk Group to where the incoming SIP call is sent. For IP-to-IP calls: Identifies the call as an IP-to-IP call if this parameter is set to -1.
Web: Trunk ID CLI: trunk-id [PstnPrefix_TrunkId]	Defines the Trunk to where the incoming SIP call is sent. **Notes:** ▪ If both 'Trunk Group ID' and 'Trunk ID' parameters are configured in the table, the routing is done according to the 'Trunk Group ID' parameter. ▪ The method for selecting the trunk's channel to which the IP call is sent is configured by the global parameter, ChannelSelectMode. ▪ Currently, this field can only be configured using the ini file.
Web: IP Profile ID CLI: ip-profile-id **[PstnPrefix_ProfileId]**	Assigns an IP Profile (configured in "Configuring IP Profiles" on page 314) to the call.
Web: Source IP Group ID CLI: src-ip-group-id **[PstnPrefix_SrcIPGroupID]**	For IP-to-Tel calls: Defines the IP Group associated with the incoming IP call. This is the IP Group that sent the INVITE message. This IP Group can later be used as the 'Serving IP Group' in the Account table for obtaining authentication user name/password for this call. For IP-to-IP calls: Assigns the IP Group to the incoming IP call. This IP Group can later be used for outbound IP routing and as the 'Serving IP Group' in the Account table for obtaining authentication user name/password for this call. For configuring registration accounts in the Account table, see "Configuring Account Table" on page 291.

28.4 IP Destinations Connectivity Feature

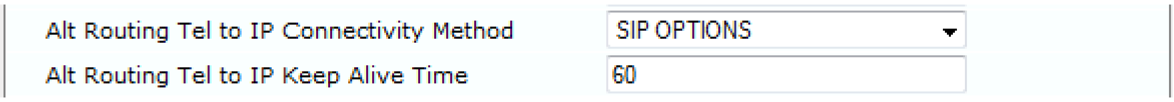
The device can be configured to check the integrity of the connectivity to IP destinations of Tel-to-IP routing rules in the Outbound IP Routing table. The IP Connectivity feature can be used for the Alternative Routing feature, whereby the device attempts to re-route calls from unavailable Tel-to-IP routing destinations to available ones (see "Alternative Routing Based on IP Connectivity" on page 412).

The device supports the following methods for checking the connectivity of IP destinations:

- **Network Connectivity:** The device checks the network connectivity of the IP destination configured by the 'Alt Routing Tel to IP Connectivity Method' parameter:
  - **SIP OPTIONS:** The device sends "keep-alive" SIP OPTIONS messages to the IP destination. If the device receives a SIP 200 OK in response, it considers the destination as available. If the destination does not respond to the OPTIONS message, then it is considered unavailable. You can configure the time interval for sending these OPTIONS messages, using the 'Alt Routing Tel to IP Keep Alive Time' parameter.

  These parameters are configured in the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **General Parameters**), as shown below:

Figure 28-4: IP Connectivity Method in Routing General Parameters Page

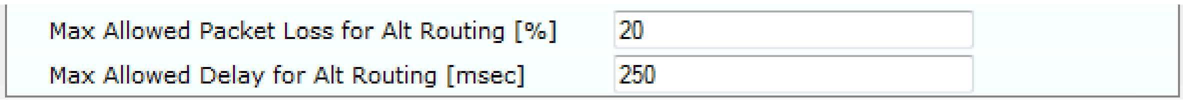


Alt Routing Tel to IP Connectivity Method: SIP OPTIONS
Alt Routing Tel to IP Keep Alive Time: 60

- **Quality of Service (QoS):** You can enable the device to check the QoS of IP destinations. The device measures the QoS according to RTCP statistics of previously established calls with the IP destination. The RTCP includes packet delay (in milliseconds) and packet loss (in percentage). If these measured statistics exceed a user-defined threshold, the destination is considered unavailable. Note that if call statistics is not received within two minutes, the QoS data is reset. These thresholds are configured using the following parameters:
  - 'Max Allowed Packet Loss for Alt Routing' (IPConnQoSMaxAllowedPL): defines the threshold value for packet loss after which the IP destination is considered unavailable.
  - 'Max Allowed Delay for Alt Routing' (IPConnQoSMaxAllowedDelay): defines the threshold value for packet delay after which the IP destination is considered unavailable

  These parameters are configured in the Routing General Parameters page, as shown below:

Figure 28-5: IP QoS Thresholds in Routing General Parameters Page



Max Allowed Packet Loss for Alt Routing [%]: 20
Max Allowed Delay for Alt Routing [msec]: 250

- **DNS Resolution:** When a host name (FQDN) is used (instead of an IP address) for the IP destination, it is resolved into an IP address by a DNS server. The device checks network connectivity and QoS of the resolved IP address. If the DNS host name is unresolved, the device considers the connectivity of the IP destination as unavailable.

You can view the connectivity status of IP destinations in the following Web interface pages:

- **Outbound IP Routing Table:** The connectivity status of the IP destination per routing rule is displayed in the 'Status' column. For more information, see "Configuring Outbound IP Routing Table" on page 397.
- **IP Connectivity:** This page displays a more informative connectivity status of the IP destinations used in Tel-to-IP routing rules in the Outbound IP Routing table. For viewing this page, see "Viewing IP Connectivity" on page 687.

28.5 Alternative Routing for Tel-to-IP Calls

The device supports various alternative Tel-to-IP call routing methods, as described in this section.

28.5.1 Alternative Routing Based on IP Connectivity

You can configure the device to route Tel-to-IP calls to an alternative IP destination when the connectivity state of an IP destination is unavailable. The alternative routing rules are configured in the Outbound IP Routing table. These rules must be configured anywhere below the "main" routing rule and with identical matching characteristics (e.g., destination prefix number) to the "main" routing rule. The device uses the first alternative route that is available. For more information on configuring alternative Tel-to-IP routing rules in the Outbound IP Routing table, see "Configuring Outbound IP Routing Table" on page 397.

⚠ **Note:** Alternative routing based on IP connectivity is applicable only when a proxy server is not used.

The device searches for an alternative IP destination when any of the following connectivity states are detected with the IP destination of the initial Tel-to-IP routing rule:

- No response received from SIP OPTIONS messages. This depends on the chosen method for checking IP connectivity.
- Poor QoS according to the configured thresholds for packet loss and delay.
- Unresolved DNS, if the configured IP destination is a domain name (or FQDN). If the domain name is resolved into two IP addresses, the timeout for INVITE re-transmissions can be configured using the HotSwapRtx parameter. For example, if you set this parameter to 3, the device attempts up to three times to route the call to the first IP address and if unsuccessful, it attempts up to three times to re-route it to the second resolved IP address.

The connectivity status of the IP destination is displayed in the 'Status' column of the Outbound IP Routing table per routing rule. If it displays a status other than "ok", then the device considers the IP destination as unavailable and attempts to re-route the call to an alternative destination. For more information on the IP connectivity methods and on viewing IP connectivity status, see "IP Destinations Connectivity Feature" on page 410.

The table below shows an example of alternative routing where the device uses an available alternative routing rule in the Outbound IP Routing table to re-route the initial Tel-to-IP call.

Table 28-4: Alternative Routing based on IP Connectivity Example

	Destination Phone Prefix	IP Destination	IP Connectivity Status	Rule Used?
Main Route	40	10.33.45.68	"No Connectivity"	No
Alternative Route #1	40	10.33.45.70	"QoS Low"	No
Alternative Route #2	40	10.33.45.72	"ok"	Yes

The steps for configuring alternative Tel-to-IP routing based on IP connectivity are summarized below.

➢ **To configure alternative Tel-to-IP routing based on IP connectivity:**

1. In the Outbound IP Routing table, add alternative Tel-to-IP routing rules for specific calls.
2. In the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **General Parameters**), do the following:
   a. Enable alternative routing based on IP connectivity, by setting the 'Enable Alt Routing Tel to IP AltRouting' (Tel2IPEnable) parameter to **Enable**.
   b. Configure the IP connectivity reason for triggering alternative routing, by setting the 'Alt Routing Tel to IP Mode' parameter (AltRoutingTel2IPMode) to one of the following:
      - SIP OPTIONS failure
      - Poor QoS
      - SIP OPTIONS failure, poor QoS, or unresolved DNS
   c. The device plays a tone to the Tel endpoint (for analog interfaces) whenever an alternative route is used. This tone is played for a user-defined time configured by the 'Alternative Routing Tone Duration' parameter.

28.5.2 Alternative Routing Based on SIP Responses

The device can perform alternative routing based on the received SIP response code (i.e., 4xx, 5xx, 6xx, or 8xx). If you have configured this response code in the Reasons for Tel-to-IP Alternative Routing table, the device attempts to re-route the call to an alternative destination, if configured. You can configure up to 10 SIP response codes in the Reasons for Tel-to-IP Alternative Routing table.

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). You can also configure the device to perform alternative routing for the following proprietary response codes that are issued by the device itself:

- **805 IP Profile Call Limit:** The device generates this response code when Call Admission Control (CAC) limits are exceeded for an IP Group. The CAC rules are configured in the IP Profile table (see "Configuring IP Profiles" on page 314). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity.
- **806 Media Limits Exceeded:** The device generates this response code when the call is terminated due to crossed thresholds of QoE metrics such as MOS, packet delay, and packet loss (configured in the Quality of Experience Profile table) and/or media bandwidth (configured in the Bandwidth profile table). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity. This is configured by 1) assigning an IP Group a QoE and/or Bandwidth profile that rejects calls if the threshold is crossed, 2) configuring 806 in the Reasons for Tel-to-IP Alternative Routing table and 3) configuring an alternative routing rule.

Note: The device also plays a tone to the endpoint whenever an alternative route is used. This tone is played for a user-defined time, configured by the AltRoutingToneDuration parameter.

Depending on configuration, the alternative routing is done using one of the following configuration entities:

- **Outbound IP Routing Rules:** You configure alternative routing rules for a specific routing rule in the Outbound IP Routing table. If the destination of the "main" routing rule is unavailable, the device searches the table for the next matching rule (e.g., destination phone number), and if available attempts to re-route the call to the IP destination configured for this alternative routing rule. For more information on configuring alternative Tel-to-IP routing rules, see "Configuring Outbound IP Routing Table" on page 397. The table below shows an example of alternative routing where the device uses the first available alternative routing rule to re-route the initial, unsuccessful Tel-to-IP call destination:

Table 28-5: Alternative Routing based on SIP Response Code Example

	Destination Phone Prefix	IP Destination	SIP Response	Rule Used?
Main Route	40	10.33.45.68	408 Request Timeout	No
Alternative Route #1	40	10.33.45.70	486 Busy Here	No
Alternative Route #2	40	10.33.45.72	200 OK	Yes

- **Proxy Sets:** Proxy Sets are used for Server-type IP Groups (e.g., an IP PBX or proxy), which define the address (IP address or FQDN) of the server (see "Configuring Proxy Sets" on page 282). As you can configure multiple IP destinations per Proxy Set, the device supports proxy redundancy, which works together with the alternative routing feature. If the destination of a routing rule in the Outbound IP Routing table is an IP Group, the device routes the call to the IP destination configured for the Proxy Set associated with the IP Group. If the first IP destination of the Proxy Set is unavailable, the device attempts to re-route the call to the next proxy destination, and so on until an available IP destination is located. To enable the Proxy Redundancy feature for a Proxy Set, set the IsProxyHotSwap parameter to 1 and the EnableProxyKeepAlive parameter to 1.

  When the Proxy Redundancy feature is enabled, the device continually monitors the connection with the proxies by using keep-alive messages (SIP OPTIONS). The device sends these messages every user-defined interval (ProxyKeepAliveTime parameter). If the first (primary) proxy in the list replies with a SIP response code that you have also configured by the 'Keep-Alive Failure Responses' parameter, the device considers the Proxy as down; otherwise, the device considers the proxy as "alive". If the proxy is still considered down after a user-defined number of re-transmissions (configured by the HotSwapRtx parameter), the device attempts to communicate (using the same INVITE) with the next configured (redundant) proxy in the list, and so on until an available redundant proxy is located. Once an available proxy is located, the device can operate in one of the following modes (configured by the ProxyRedundancyMode parameter):

  - **Parking mode:** The device continues operating with the redundant proxy (now active) until the next failure occurs, after which it switches to the next redundant proxy.
  - **Homing mode:** The device always attempts to operate with the primary proxy. In other words, it switches back to the primary proxy whenever it's available again.

  If none of the proxy servers respond, the device goes over the list again.

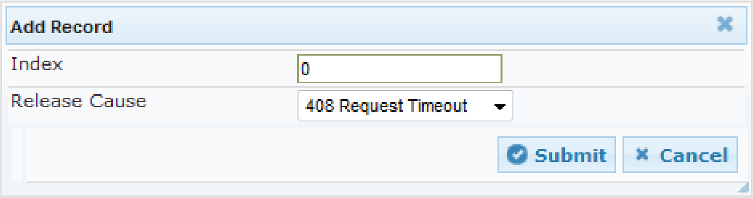
Note: The device assumes that all the proxy servers belonging to the Proxy Set are synchronized with regards to registered users. Thus, when the device locates an available proxy using the Hot Swap feature, it does not re-register the users; new registration (refresh) is done as normal.

The steps for configuring alternative Tel-to-IP routing based on SIP response codes are summarized below.

➢ **To configure alternative Tel-to-IP routing based on SIP response codes:**

1. Configure SIP response codes (call failure reasons) that invoke alternative Tel-to-IP routing:
   a. Open the Reasons for Tel-to-IP Alternative Routing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **Alternative Reasons** > **Reasons for Tel-to-IP**).
   b. Click **Add**; the following dialog box appears:

Figure 28-6: Reasons for Tel-to-IP Alternative Routing Table - Add Record



Add Record

Index: 0

Release Cause: 408 Request Timeout

   c. Configure a SIP response code for alternative routing according to the parameters described in the table below.
   d. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 28-6: Reasons for Tel-to-IP Alternative Routing Table Parameter Descriptions

Parameter	Description
Index **[AltRouteCauseTel2Ip_Index]**	Defines an index number for the new table record.
Release Cause CLI: rel-cause **[AltRouteCauseTel2Ip_ReleaseCause]**	Defines a SIP response code that if received, the device attempts to route the call to an alternative destination (if configured).

2. Enable alternative routing based on SIP responses, by setting the 'Redundant Routing Mode' parameter in the Proxy & Registration page to one of the following:
   - **Routing Table:** Outbound IP Routing table is used for alternative routing.
   - **Proxy:** Proxy Set redundancy feature is used for alternative routing.
3. If you are using the Outbound IP Routing table, configure alternative routing rules with identical call matching characteristics, but with different IP destinations.
4. If you are using the Proxy Set, configure redundant proxies.

28.5.3 Alternative Routing upon SIP 3xx with Multiple Contacts

You can configure how the device handles received SIP 3xx responses that contain multiple alternative contacts. The 3xx response indicates that the original destination is unavailable (e.g., 301 Moved Permanently – user cannot be found) and that the call can be redirected to alternative destinations specified in the SIP Contact headers.

Configured by the '3xx Use Alt Route Reasons' parameter, the device can handle the receipt of 3xx responses using one of the following methods:

- The device tries each contact sequentially, listed in the Contact headers, until a successful destination is found. If a contact responds with a SIP 486 or 600, the device does not try to redirect the call to the next contact and drops the call.
- The device tries each contact sequentially, listed in the Contact headers. If a SIP 6xx Global Failure response is received during this process (e.g., 600 Busy Everywhere),

the device does not try to redirect the call to the next contact and drops the call.

- The device redirects the call to the first contact listed in the Contact header. If the contact responds with a SIP response that is configured in the Reasons for Tel-to-IP Alternative Routing table (see "Alternative Routing Based on SIP Responses" on page 413), the device tries to redirect the call to the next contact, and so on. If a contact responds with a response that is not configured in the table, the device does not try to redirect the call to the next contact and drops the call.

> **Note:** If a SIP 401 or 407 response is received from a contact, the device does not try to redirect the call to the next contact. Instead, the device continues with the regular authentication process, as indicated by these response types.

28.5.4 PSTN Fallback

The PSTN Fallback feature enables the device to re-route a Tel-to-IP call to the legacy PSTN using one of its trunks if the IP destination is unavailable. For example, if poor voice quality is detected over the IP network, the device attempts to re-route the call to the PSTN.

The steps for configuring alternative Tel-to-IP routing to the legacy PSTN are summarized below.

➢ **To configure alternative Tel-to-IP routing to the legacy PSTN:**

1. Configure an alternative routing rule in the Outbound IP Routing table with the same call matching characteristics (e.g., phone number destination), but where the destination is the IP address of the device itself.
2. Configure an IP-to-Tel routing rule in the Inbound IP Routing table to route calls received from the device (i.e., its IP address) to a specific Trunk Group connected to the PSTN. This configuration is necessary as the re-routed call is now considered an IP-to-Tel call. For configuring IP-to-Tel routing rules, see "Configuring the Inbound IP Routing Table" on page 406.

> **Note:** The PSTN Fallback feature is applicable only to digital interfaces.

28.6 Alternative Routing for IP-to-Tel Calls

The device supports alternative IP-to-Tel call routing, as described in this section.

28.6.1 Alternative Routing to Trunk upon Q.931 Call Release Cause Code

You can configure up to 10 ISDN Q.931 release cause codes, which if received from the Tel side, the device routes the IP-to-Tel call to an alternative Trunk Group, if configured. Alternative IP-to-Tel routing rules are configured in the Inbound IP Routing table. These rules must be configured anywhere below the "main" routing rule and with identical matching characteristics (e.g., destination prefix number) to the "main" routing rule. The device uses the first alternative route that is available. For more information on configuring alternative IP-to-Tel routing rules in the Inbound IP Routing table, see "Configuring Inbound IP Routing Table" on page 406.

A release cause code indicates that the IP-to-Tel call has been rejected or disconnected on the Tel side. The release cause codes are configured in the Reasons for IP-to-Tel Alternative Routing table. For example, you can configure alternative IP-to-Tel routing for scenarios where the initial Tel destination is busy and a Q.931 Cause Code No. 17 is received (or for other call releases that issue the default Cause Code No. 3).

You can configure a default release cause code that the device issues itself upon the following scenarios:

- The device initiates a call release whose cause is unknown.
- No free channels (i.e., busy) in the Trunk Group.
- No appropriate routing rule located in the Inbound IP Routing table.
- Phone number is not located in the Inbound IP Routing table.

This default release code is set to Cause Code No. 3 (No Route to Destination).You can change the code number using the 'Default Release Cause' parameter, located on the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**).

Notes:

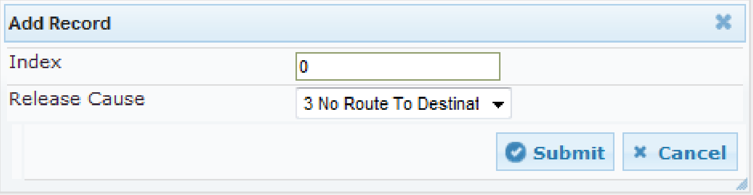
- If a Trunk is disconnected or not synchronized, the device issues itself the internal Cause Code No. 27. This cause code is mapped (by default) to SIP 502.
- The default release cause is described in the Q.931 notation and translated to corresponding SIP 40x or 50x values (e.g., Cause Code No. 3 to SIP 404, and Cause Code No. 34 to SIP 503).
- For analog interfaces: For information on mapping PSTN release causes to SIP responses, see PSTN Release Cause to SIP Response Mapping on page 395.
- For mapping SIP-to-Q.931 and Q.931-to-SIP release causes, see Configuring Release Cause Mapping on page 390.

The following procedure describes how to configure alternative routing reasons for IP-to-Tel calls in the Web interface. You can also configure alternative routing reasons for IP-to-Tel calls using the table ini file parameter, AltRouteCauseIP2Tel or CLI command, configure voip/gw routing alt-route-cause-ip2tel.

➢ To configure alternative Trunk Group routing based on Q.931 cause codes:

1. In the Proxy & Registration page, set the 'Redundant Routing Mode' parameter to **Routing Table** so that the device uses the Inbound IP Routing table for alternative routing.
2. In the Inbound IP Routing table, configure alternative routing rules with the same call matching characteristics, but with different Trunk Group destinations.
3. Configure Q.931 cause codes that invoke alternative IP-to-Tel routing:
   a. Open the Reasons for IP-to-Tel Alternative Routing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **Alternative Routing Reasons** > **Reasons for IP-to-Tel**).
   b. Click **Add**; the following dialog box appears:

Figure 28-7: IP to Tel Reasons - Reasons for Alternative Routing Page



Add Record

Index: 0

Release Cause: 3 No Route To Destinat

   c. Configure a Q.931 release cause code for alternative routing according to the parameters described in the table below.
   d. Click **Submit**, and then save ("burn") your settings to flash memory.

Table 28-7: Reasons for IP-to-Tel Alternative Routing Table Parameter Descriptions

Parameter	Description
Index **[AltRouteCauseIP2Tel_Index]**	Defines an index number for the new table record.
Release Cause CLI: rel-cause **[AltRouteCauseIP2Tel_ReleaseCause]**	Defines a Q.931 release code that if received, the device attempts to route the call to an alternative destination (if configured).

28.6.2 Alternative Routing to an IP Destination upon a Busy Trunk

The Forward on Busy Trunk Destination table lets you configure alternative routing rules for forwarding (i.e., call redirection) IP-to-Tel calls to an alternative IP destination using SIP 3xx responses. These rules are used upon the following:

- For digital interfaces: Trunk Group has no free channels (i.e., “busy”).
- For analog interfaces: Unavailable FXS / FXO Trunk Group. This feature can be used, for example, to forward the call to another FXS / FXO device.

This feature is configured per Trunk Group. The alternative destination can be defined as a host name or as a SIP Request-URI user name and host part (i.e., user@host). For example, the below configuration forwards IP-to-Tel calls to destination user “112” at host IP address 10.13.4.12, port 5060, using transport protocol TCP, if Trunk Group ID 2 is unavailable:

```
ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp;
```

When configured with user@host, the original destination number is replaced by the user part.

The device forwards calls using this table only if no alternative IP-to-Tel routing rule has been configured in the Inbound IP Routing table or alternative routing fails and the following reason(s) in the SIP Diversion header of 3xx messages exists:

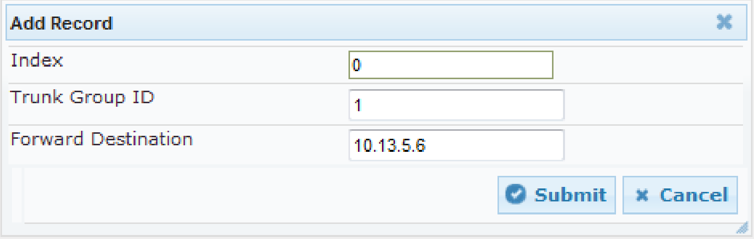
- For digital interfaces: “out-of-service” - all trunks are unavailable/disconnected
- "unavailable":
  - For digital interfaces: All trunks are busy or unavailable
  - For analog interfaces: All FXS / FXO lines pertaining to a Trunk Group are busy or unavailable

The following procedure describes how to configure Forward on Busy Trunks in the Web interface. You can also configure this using the table ini file parameter, ForwardOnBusyTrunkDest or CLI command, configure voip/gw routing fwd-on-bsy-trk-dst.

➢ To configure a Forward on Busy Trunk Destination rule:

1. Open the Forward on Busy Trunk Destination page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **Forward on Busy Trunk**).
2. Click **Add**; the following dialog box appears:

Figure 28-8: Forward on Busy Trunk Destination Page - Add Record



Add Record

Index: 0

Trunk Group ID: 1

Forward Destination: 10.13.5.6

The figure above displays a configuration that forwards IP-to-Tel calls destined for Trunk Group ID 1 to destination IP address 10.13.5.67 if the conditions mentioned earlier exist.

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

Table 28-8: Forward on Busy Trunk Destination Parameter Descriptions

Parameter	Description
Trunk Group ID CLI: trunk-group-id **[ForwardOnBusyTrunkDest_TrunkGroupId]**	Defines the Trunk Group ID to which the IP call is destined to.
Forward Destination CLI: forward-dst **[ForwardOnBusyTrunkDest_ForwardDestination]**	Defines the alternative IP destination for the call used if the Trunk Group is busy or unavailable. The valid value can be an IP address in dotted-decimal notation, an FQDN, or a SIP Request-URI user name and host part (i.e., user@host). The following syntax can also be used: host:port;transport=xxx (i.e., IP address, port and transport type). **Note:** When configured with a user@host, the original destination number is replaced by the user part.

28.6.3 Alternative Routing upon ISDN Disconnect

You can configure when the device sends a call to an alternative route if it receives an ISDN Q.931 Disconnect message with a Progress Indicator (PI) IE from the Tel side for IP-to-Tel calls. The Disconnect message indicates that the call cannot be established due to, for example, a busy state on the Tel side. Using the DisconnectCallwithPIifAlt ini file parameter, you can configure the following modes of operation:

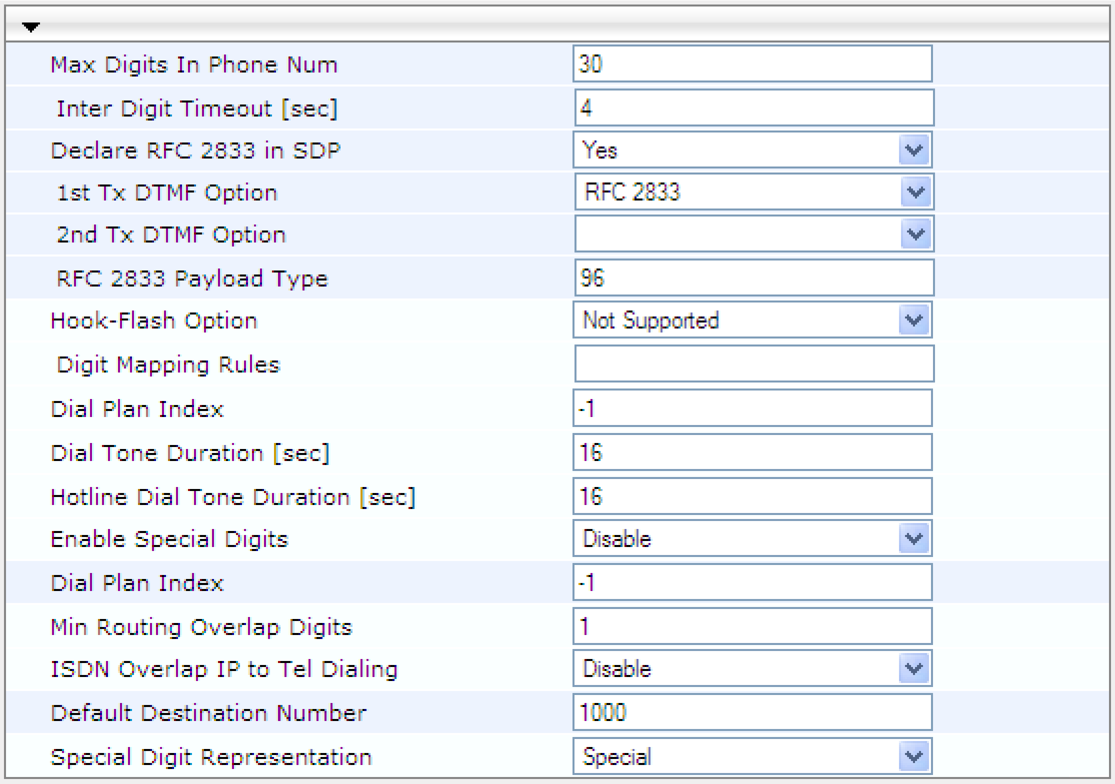
- The device does not immediately disconnect the call. Instead, it waits for any subsequent media from the Tel side (e.g., "this number is currently busy") and forwards it to the IP side (SIP 183 for early media). Only when it receives a Q.931 Release message, does the device disconnect the call (sends a SIP BYE message to the IP side). If you have configured an alternative route, the device sends the IP call to the alternative route.
- The device immediately sends the IP call to an alternative route, if you have configured one. If no alternative route has been configured and the Disconnect message is received with PI, the device forwards the subsequent early media to the IP side. The device disconnects the IP call only upon receipt of the subsequent Release message.

29 Configuring DTMF and Dialing

The DTMF & Dialing page is used to configure parameters associated with dual-tone multi-frequency (DTMF) and dialing. For a description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 763.

➢ **To configure the DTMF and dialing parameters:**

1. Open the DTMF & Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF & Supplementary** > **DTMF & Dialing**).



Max Digits In Phone Num	30
Inter Digit Timeout [sec]	4
Declare RFC 2833 in SDP	Yes
1st Tx DTMF Option	RFC 2833
2nd Tx DTMF Option	
RFC 2833 Payload Type	96
Hook-Flash Option	Not Supported
Digit Mapping Rules	
Dial Plan Index	-1
Dial Tone Duration [sec]	16
Hotline Dial Tone Duration [sec]	16
Enable Special Digits	Disable
Dial Plan Index	-1
Min Routing Overlap Digits	1
ISDN Overlap IP to Tel Dialing	Disable
Default Destination Number	1000
Special Digit Representation	Special

2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

29.1 Dialing Plan Features

This section describes various dialing plan features supported by the device.

29.1.1 Digit Mapping

Digit map pattern rules are used for Tel-to-IP ISDN overlap dialing (by setting the ISDNRxOverlap parameter to 1) to reduce the dialing period (for digital interface). For more information on digit maps for ISDN overlapping, see ISDN Overlap Dialing on page 362. The device collects digits until a match is found in the user-defined digit pattern (e.g., for closed numbering schemes). The device stops collecting digits and starts sending the digits (collected number) upon any of the following scenarios:

- Maximum number of digits is received. You can define (using the MaxDigits parameter) the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side by the device. When the number of collected digits reaches the maximum (or a digit map pattern is matched), the device uses these digits for the called destination number.
- Inter-digit timeout expires (e.g., for open numbering schemes). This is defined using the TimeBetweenDigits parameter. This is the time that the device waits between each received digit. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number.
- The phone's pound (#) key is pressed.
- Digit string (i.e., dialed number) matches one of the patterns defined in the digit map.

Digit map (pattern) rules are defined using the DigitMapping parameter. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar ("|"). The maximum length of the entire digit pattern is 152 characters. The available notations are described in the table below:

Table 29-1: Digit Map Pattern Notations

Notation	Description
[n-m]	Range of numbers (not letters).
.	(single dot) Repeat digits until next notation (e.g., T).
x	Any single digit. **Note:** This notation does not apply to some scenarios when using the star (*) or hash (#) key. For example, the key sequence of ** must be presented in the dial plan as *x.s (instead of xx).
T	Dial timeout (configured by the TimeBetweenDigits parameter).
S	Short timer (configured by the TimeBetweenDigits parameter; default is two seconds) that can be used when a specific rule is defined after a more general rule. For example, if the digit map is 99|998, then the digit collection is terminated after the first two 9 digits are received. Therefore, the second rule of 998 can never be matched. But when the digit map is 99s|998, then after dialing the first two 9 digits, the device waits another two seconds within which the caller can enter the digit 8.

Below is an example of a digit map pattern containing eight rules:

```
DigitMapping = 11xS|00[1-
7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x|xx.T
```

In the example, the rule "00[1-7]xxx" denotes dialed numbers that begin with 00, and then any digit from 1 through 7, followed by three digits (of any number). Once the device receives these digits, it does not wait for additional digits, but starts sending the collected digits (dialed number) immediately.

Notes:

- If you want the device to accept/dial any number, ensure that the digit map contains the rule "xx.T"; otherwise, dialed numbers not defined in the digit map are rejected.
- If you are using an external Dial Plan file for dialing plans (see "Dialing Plans for Digit Collection" on page 606), the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map (configured by the DigitMapping parameter).
- It may be useful to configure both Dial Plan file and Digit Maps. For example, the Digit Map can be used for complex digit patterns (which are not supported by the Dial Plan) and the Dial Plan can be used for long lists of relatively simple digit patterns. In addition, as timeout between digits is not supported by the Dial Plan, the Digit Map can be used to define digit patterns (MaxDigits parameter) that are shorter than those defined in the Dial Plan, or left at default. For example, “xx.T” Digit Map instructs the device to use the Dial Plan and if no matching digit pattern, it waits for two more digits and then after a timeout (TimeBetweenDigits parameter), it sends the collected digits. Therefore, this ensures that calls are not rejected as a result of their digit pattern not been completed in the Dial Plan.

29.1.2 External Dial Plan File

The device can be loaded with a Dial Plan file with user-defined dialing plans. For more information, see ''Dial Plan File'' on page 606.

30 Configuring Supplementary Services

This section describes SIP supplementary services that can enhance your telephone service.

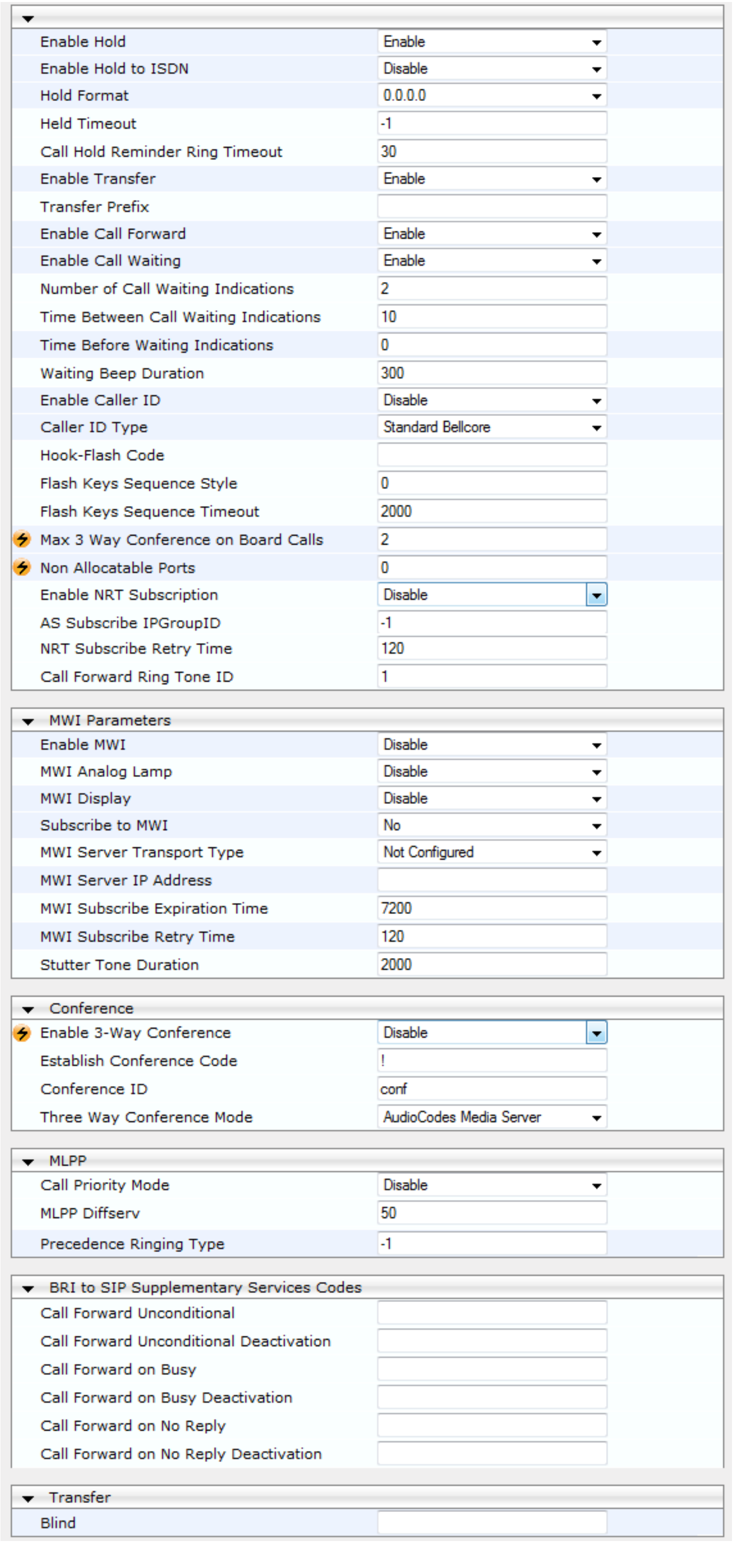
Notes:

- All call participants must support the specific supplementary service that is used.
- When working with certain application servers (such as BroadSoft’s BroadWorks) in client server mode (the application server controls all supplementary services and keypad features by itself), the device's supplementary services must be disabled.

The Supplementary Services page is used to configure many of the discussed supplementary services parameters. For a description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 763.

➢ To configure supplementary services parameters:

1. Open the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF & Supplementary** > **Supplementary Services**).



Enable Hold	Enable
Enable Hold to ISDN	Disable
Hold Format	0.0.0.0
Held Timeout	-1
Call Hold Reminder Ring Timeout	30
Enable Transfer	Enable
Transfer Prefix	
Enable Call Forward	Enable
Enable Call Waiting	Enable
Number of Call Waiting Indications	2
Time Between Call Waiting Indications	10
Time Before Waiting Indications	0
Waiting Beep Duration	300
Enable Caller ID	Disable
Caller ID Type	Standard Bellcore
Hook-Flash Code	
Flash Keys Sequence Style	0
Flash Keys Sequence Timeout	2000
⚡ Max 3 Way Conference on Board Calls	2
⚡ Non Allocatable Ports	0
Enable NRT Subscription	Disable
AS Subscribe IPGroupID	-1
NRT Subscribe Retry Time	120
Call Forward Ring Tone ID	1

MWI Parameters	
Enable MWI	Disable
MWI Analog Lamp	Disable
MWI Display	Disable
Subscribe to MWI	No
MWI Server Transport Type	Not Configured
MWI Server IP Address	
MWI Subscribe Expiration Time	7200
MWI Subscribe Retry Time	120
Stutter Tone Duration	2000

Conference	
⚡ Enable 3-Way Conference	Disable
Establish Conference Code	!
Conference ID	conf
Three Way Conference Mode	AudioCodes Media Server

MLPP	
Call Priority Mode	Disable
MLPP Diffserv	50
Precedence Ringing Type	-1

BRI to SIP Supplementary Services Codes	
Call Forward Unconditional	
Call Forward Unconditional Deactivation	
Call Forward on Busy	
Call Forward on Busy Deactivation	
Call Forward on No Reply	
Call Forward on No Reply Deactivation	

Transfer	
Blind	

2. Configure the parameters as required.

3. Click **Submit**, or click the **Subscribe to MWI** or **Unsubscribe to MWI** buttons to save your changes and to subscribe / unsubscribe to the MWI server.

4. To save the changes to flash memory, see "Saving Configuration" on page 592.

30.1 Call Hold and Retrieve

Initiating Call Hold and Retrieve:

- Active calls can be put on-hold by pressing the phone's hook-flash button.
- The party that initiates the hold is called the *holding* party; the other party is called the *held* party.
- After a successful Hold, the holding party hears a dial tone (HELD_TONE defined in the device's Call Progress Tones file).
- Call retrieve can be performed only by the holding party while the call is held and active.
- The holding party performs the retrieve by pressing the telephone's hook-flash button.
- After a successful retrieve, the voice is connected again.
- Hold is performed by sending a re-INVITE message with IP address 0.0.0.0 or a=sendonly in the SDP, according to the HoldFormat parameter.
- The hold and retrieve functionalities are implemented by re-INVITE messages. The IP address 0.0.0.0 as the connection IP address or the string ‘a=inactive’ in the received re-INVITE SDP cause the device to enter Hold state and to play the held tone (configured in the device) to the PBX/PSTN. If the string ‘a=sendonly’ is received in the SDP message, the device stops sending RTP packets, but continues to listen to the incoming RTP packets. Usually, the remote party plays, in this scenario, Music on Hold (MOH) and the device forwards the MOH to the held party.

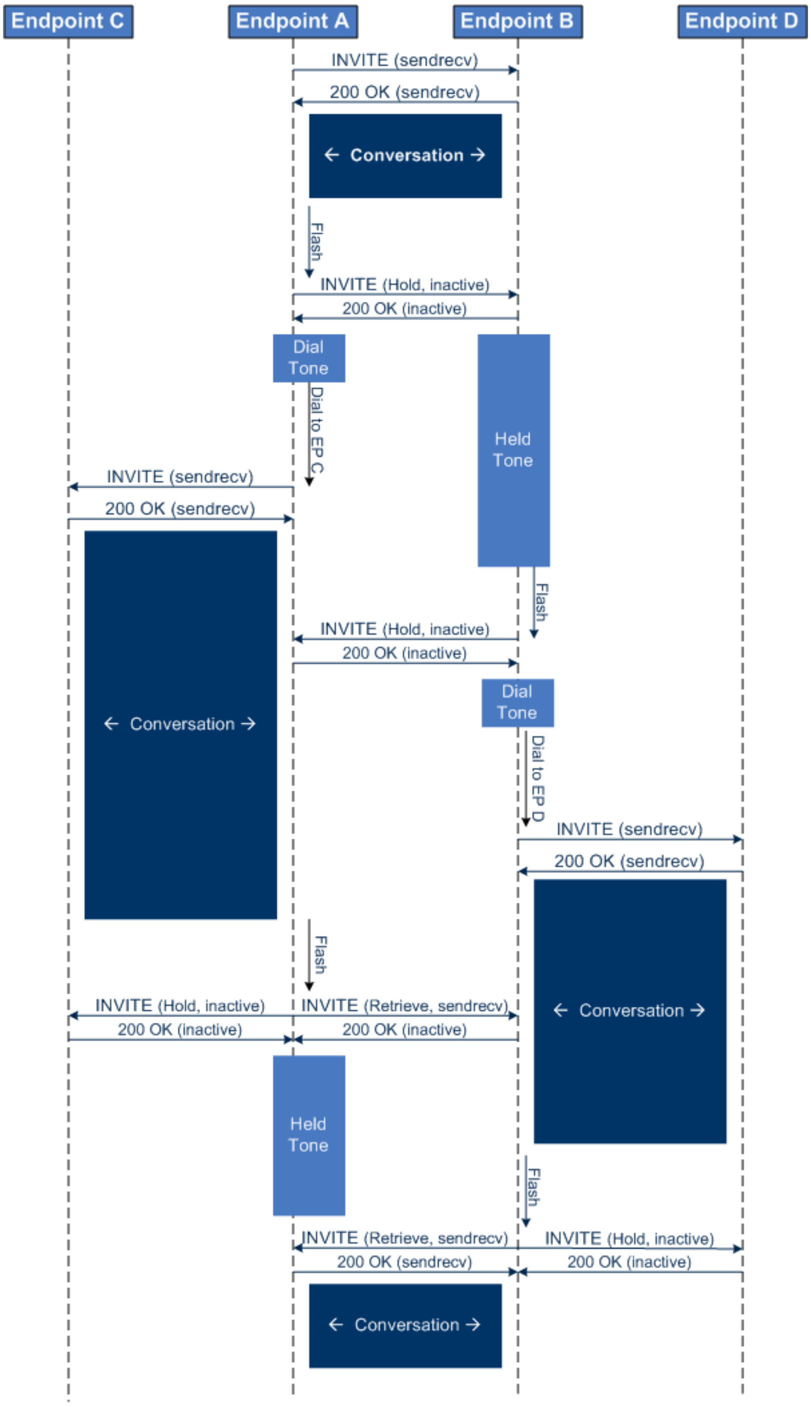
Receiving Hold/Retrieve:

- When an active call receives a re-INVITE message with IP address 0.0.0.0 or ‘inactive’ string in SDP, the device stops sending RTP and plays a local held tone.
- When an active call receives a re-INVITE message with the ‘sendonly’ string in SDP, the device stops sending RTP and listens to the remote party. In this mode, it is expected that music on-hold (or any other hold tone) is played (over IP) by the remote party.

You can also configure the device to keep a call on-hold for a user-defined time after which the call is disconnected, using the HeldTimeout parameter.

The device also supports "double call hold" for FXS interfaces where the called party, which has been placed on-hold by the calling party, can then place the calling party on hold as well and make a call to another destination. The flowchart below provides an example of this type of call hold:

Figure 30-1: Double Hold SIP Call Flow



The flowchart above describes the following "double" call-hold scenario:

1. A calls B and establishes a voice path.
2. A places B on hold; A hears a dial tone and B hears a held tone.
3. A calls C and establishes a voice path.
4. B places A on hold; B hears a dial tone.
5. B calls D and establishes a voice path.

6. A ends call with C; A hears a held tone.
7. B ends call with D.
8. B retrieves call with A.

Notes:

- If a party that is placed on hold (e.g., B in the above example) is called by another party (e.g., D), then the on-hold party receives a call waiting tone instead of the held tone.
- While in a Double Hold state, placing the phone on-hook disconnects both calls (i.e. call transfer is not performed).
- You can enable the device to handle incoming re-INVITE messages with "a=sendonly" in the SDP, in the same way as if "a=inactive" is received in the SDP. This is configured using the SIPHoldBehavior parameter. When enabled, the device plays a held tone to the Tel phone and responds with a 200 OK containing "a=recvonly" in the SDP.

30.2 Call Pickup

The device supports the Call Pick-Up feature, whereby the FXS user can answer someone else's telephone call by pressing a user-defined sequence of phone keys. When the user dials the user-defined digits (e.g., #77), the incoming call from the other phone is forwarded to the FXS user's phone. This feature is configured using the parameter KeyCallPickup.

Note: The Call Pick-Up feature is supported only for FXS endpoints pertaining to the same Trunk Group ID.

30.3 BRI Suspend and Resume

The device supports call suspend and resume services initiated by ISDN BRI phones connected to the device. During an ongoing call, the BRI phone user can suspend the call by typically pressing the phone's "P" button or a sequence of keys (depending on the phone), and then on-hooking the handset. To resume the call, the phone user typically presses the same keys or button again and then off-hooks the phone. During the suspended state, the device plays a howler tone to the remote party. This service is also supported when instead of pressing the call park button(s), the phone cable is disconnected (suspending the call) and then reconnected again (resuming the call).

If the phone user does not resume the call within a user-defined interval (configured using the HeldTimeout parameter), the device releases the call.

Note: Only one call can be suspended per trunk. If another suspend request is received from a BRI phone while there is already a suspended call (even if done by another BRI phone connected to the same trunk), the device rejects this suspend request.

30.4 Consultation Feature

The device's Consultation feature allows you to place one number on hold and make a second call to another party.

- After holding a call (by pressing hook-flash), the holding party hears a dial tone and can then initiate a new call, which is called a Consultation call.
- While hearing a dial tone, or when dialing to the new destination (before dialing is complete), the user can retrieve the held call by pressing hook-flash.
- The held call can't be retrieved while ringback tone is heard.
- After the Consultation call is connected, the user can toggle between the held and active call by pressing the hook-flash key.

Note: The Consultation feature is applicable only to FXS interfaces.

30.5 Call Transfer

This section describes the device's support for call transfer types.

30.5.1 Consultation Call Transfer

The device supports Consultation Call Transfer using the SIP REFER message and Replaces header. The common method to perform a consultation transfer is described in the following example, which assumes three call parties:

- Party A = transferring
- Party B = transferred
- Party C = transferred to

1. A Calls B.
2. B answers.
3. A presses the hook-flash button and places B on-hold (party B hears a hold tone).
4. A dials C.
5. After A completes dialing C, A can perform the transfer by on-hooking the A phone.
6. After the transfer is complete, B and C parties are engaged in a call.

The transfer can be initiated at any of the following stages of the call between A and C:

- Just after completing dialing C phone number - transfer from setup
- While hearing ringback – transfer from alert
- While speaking to C - transfer from active

Note: For FXS interfaces, the device can also handle call transfers using SIP INVITE and re-INVITE messages, instead of REFER messages. This is useful when communicating with SIP UAs that do not support the receipt of REFER messages. This feature is applicable to FXS interfaces. To enable this support, use the EnableCallTransferUsingReinvites parameter.

The device also supports attended (consultation) call transfer for BRI phones (user side) connected to the device and using the Euro ISDN protocol. BRI call transfer is according to

ETSI TS 183 036, Section G.2 (Explicit Communication Transfer – ECT). Call transfer is enabled using the EnableTransfer and EnableHoldtoISDN parameters.

The Explicit Call Transfer (ECT, according to ETS-300-367, 368, 369) supplementary service is supported for BRI and PRI trunks. This service provides the served user who has two calls to ask the network to connect these two calls together and release its connection to both parties. The two calls can be incoming or outgoing calls. This service is similar to NI-2 Two B-Channel Transfer (TBCT) Supplementary Service. The main difference is that in ECT one of the calls must be in HELD state. The ECT standard defines two methods - Implicit and Explicit. In implicit method, the two calls must be on the same trunk. BRI uses the implicit mechanism, and PRI the explicit mechanism.

30.5.2 Consultation Transfer for QSIG Path Replacement

The device can interwork consultation call transfer requests for ISDN QSIG-to-IP calls. When the device receives a request for a consultation call transfer from the PBX, the device sends a SIP REFER message with a Replaces header to the SIP UA to transfer it to another SIP UA. Once the two SIP UA parties are successfully connected, the device requests the PBX to disconnect the ISDN call, thereby freeing resources on the PBX.

For example, assume legacy PBX user "A" has two established calls connected through the device – one with remote SIP UA "B" and the other with SIP UA "C". In this scenario, user "A" initiates a consultation call transfer to connect "B" with "C". The device receives the consultation call transfer request from the PBX and then connects "B" with "C", by sending "B" a REFER message with a Replaces header (i.e., replace caller "A" with "C"). Upon receipt of a SIP NOTIFY 200 message in response to the REFER, the device sends a Q.931 Disconnect messages to the PBX, notifying the PBX that it can disconnect the ISDN calls (of user "A").

This feature is enabled by the QSIGPathReplacementMode parameter.

30.5.3 Blind Call Transfer

Blind call transfer is done (using SIP REFER messages) after a call is established between call parties A and B, and party A decides to immediately transfer the call to C without first speaking to C. The result of the transfer is a call between B and C (similar to consultation transfer, but skipping the consultation stage).

You can also use the ManipulateIP2PSTNReferTo parameter to manipulate the destination number according to the number received in the SIP Refer-To header. This is applicable to all types of blind transfers to the PSTN (e.g., TBCT, ECT, RLT, QSIG, FXO, and CAS). During blind transfer, the device initiates a new call to the PSTN and the destination number of this call can be manipulated if this parameter is enabled. The following is an example of such a blind transfer:

1. IP phone "A" calls PSTN phone "B", and the call is established.
2. "A" performs a blind transfer to PSTN phone "C". It does this as follows:
    a. "A" sends a SIP REFER message (with the phone number of "C" in the Refer-To header) to the device.
    b. The device sends a Q.931 Setup message to "C". This feature enables manipulating the called party number in this outgoing Setup message.

The manipulation is done as follows:

1. If you configure a value for the xferPrefix parameter, then this value (string) is added as a prefix to the number in the Refer-To header.
2. This called party number is then manipulated using the IP-to-Tel Destination Phone Number Manipulation table.
3. The source number of the transferred call is taken from the original call, according to its initial direction:

- Tel-to-IP call: source number of the original call.
- IP-to-Tel call: destination number of the original call.
- If the UseReferredByForCallingNumber parameter is set to 1, the source number is taken from the SIP Referred-By header if included in the received SIP REFER message.

This source number can also be used as the value for the 'Source Prefix' field in the IP-to-Tel Destination Phone Number Manipulation table. The local IP address is used as the value for the 'Source IP Address' field.

Note: Manipulation using the ManipulateIP2PSTNReferTo parameter does not affect IP-to-Trunk Group routing rules.

30.6 Call Forward

For digital interfaces: The device supports Call Deflection (ETS-300-207-1) for Euro ISDN and QSIG (ETSI TS 102 393) for Network and User sides, which provides IP-ISDN interworking of call forwarding (call diversion) when the device receives a SIP 302 response.

Call forward performed by the SIP side: Upon receipt of a Facility message with Call Rerouting IE from the PSTN, the device initiates a SIP transfer process by sending a SIP 302 (including the Call Rerouting destination number) to the IP in response to the remote SIP entity's INVITE message. The device then responds with a Disconnect message to the PSTN side.

Call forward performed by the PSTN side: When the device sends the INVITE message to the remote SIP entity and receives a SIP 302 response, the device sends a Facility message with the same IE mentioned above to the PSTN, and waits for the PSTN side to disconnect the call. This is configured using the CallReroutingMode.

For analog interfaces: The following methods of call forwarding are supported:

- Immediate: incoming call is forwarded immediately and unconditionally.
- Busy: incoming call is forwarded if the endpoint is busy.
- No Reply: incoming call is forwarded if it isn't answered for a specified time.
- On Busy or No Reply: incoming call is forwarded if the port is busy or when calls are not answered after a specified time.
- Do Not Disturb: immediately reject incoming calls. Upon receiving a call for a Do Not Disturb, the 603 Decline SIP response code is sent.

Three forms of forwarding parties are available:

- Served party: party configured to forward the call (FXS device).
- Originating party: party that initiates the first call (FXS or FXO device).
- Diverted party: new destination of the forwarded call (FXS or FXO device).

The served party (FXS interface) can be configured through the Web interface (see Configuring Call Forward on page 481) or ini file to activate one of the call forward modes. These modes are configurable per endpoint.

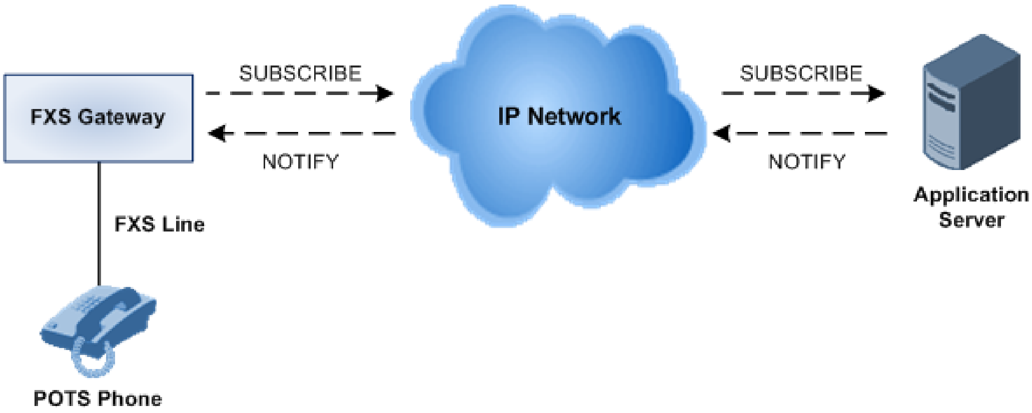
Notes:

- When call forward is initiated, the device sends a SIP 302 response with a contact that contains the phone number from the forward table and its corresponding IP address from the routing table (or when a proxy is used, the proxy's IP address).
- For receiving call forward, the device handles SIP 3xx responses for redirecting calls with a new contact.

30.6.1 Call Forward Reminder Ring

The device supports the Call Forward Reminder Ring feature for FXS interfaces, whereby the device's FXS endpoint emits a short ring burst, only in **onhook** state, when a third-party Application Server (e.g., softswitch) forwards an incoming call to another destination. This is important in that it notifies (audibly) the FXS endpoint user that a call forwarding service is currently being performed.

Figure 30-2: Call Forward Reminder with Application Server



The device generates a Call Forward Reminder ring burst to the FXS endpoint each time it receives a SIP NOTIFY message with a “reminder ring” xml body. The NOTIFY request is sent from the Application Server to the device each time the Application Server forwards an incoming call. The service is cancelled when an UNSUBSCRIBE request is sent from the device, or when the Subscription time expires.

The reminder-ring tone can be defined by using the parameter CallForwardRingToneID, which points to a ring tone defined in the Call Progress Tone file.

The following parameters are used to configure this feature:

- EnableNRTSubscription
- ASSubscribeIPGroupID
- NRTSubscribeRetryTime
- CallForwardRingToneID

30.6.2 Call Forward Reminder (Off-Hook) Special Dial Tone

The device plays a special dial tone (stutter dial tone - Tone Type #15) to a specific FXS endpoint when the phone is off-hooked and when a third-party Application server (AS), e.g., a softswitch is used to forward calls intended for the endpoint, to another destination. This is useful in that it reminds the FXS user of this service. This feature does not involve device subscription (SIP SUBSCRIBE) to the AS.

Activation/deactivation of the service is notified by the server. An unsolicited SIP NOTIFY request is sent from the AS to the device when the Call Forward service is activated or de-activated. Depending on this NOTIFY request, the device plays either the standard dial tone or the special dial tone for Call Forward.

For playing the special dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"

- Message body is the XML body and contains the “dial-tone-pattern” set to "special-condition-tone" (<ss:dial-tone-pattern>special-condition-tone</ss:dial-tone-pattern>), which is the special tone indication.

To cancel the special dial tone and playing the regular dial tone, the received SIP NOTIFY message must contain the following headers:

- **From and To:** contain the same information, indicating the specific endpoint
- **Event:** ua-profile
- **Content-Type:** "application/simservs+xml"
- Message body is the XML body containing the “dial-tone-pattern” set to "standard-condition-tone" (<ss:dial-tone-pattern>standard-condition-tone</ss:dial-tone-pattern>), which is the regular dial tone indication.

Therefore, the special dial tone is valid until another SIP NOTIFY is received that instructs otherwise (as described above).

Note: if the MWI service is active, the MWI dial tone overrides this special Call Forward dial tone.

30.6.3 Call Forward Reminder Dial Tone (Off-Hook) upon Spanish SIP Alert-Info

The device plays a special dial tone to FXS phones in off-hook state that are activated with the call forwarding service. The special dial tone is used as a result of the device receiving a SIP NOTIFY message from a third-party softswitch providing the call forwarding service with the following SIP Alert-Info header:

```
Alert-Info: <http://127.0.0.1/Tono-Espec-Invitacion>;lpi-
aviso=Desvio-Inmediato
```

This special tone is a stutter dial tone (Tone Type = 15), as defined in the CPT file.

The FXS phone user, connected to the device, activates the call forwarding service by dialing a special number (e.g., *21*xxxxx) and as a result, the device sends a regular SIP INVITE message to the softswitch. The softswitch later notifies of the activation of the forwarding service by sending an unsolicited NOTIFY message with the Alert-Info header, as mentioned above.

When the call forwarding service is de-activated, for example, by dialing #21# and sending an INVITE with this number, the softswitch sends another SIP NOTIFY message with the following Alert-Info header:

```
Alert-Info: <http://127.0.0.1/ Tono-Normal-Invitacion>; Aviso =
Desvió-Inmediato
```

From this point on, the device plays a normal dial tone to the FXS phone when it goes off-hook.

30.6.4 BRI Call Forwarding

The device supports call forwarding (CF) services initiated by ISDN Basic BRI phones connected to it. Upon receipt of an ISDN Facility message for call forward from the BRI phone, the device sends a SIP INVITE to the softswitch with a user-defined code in the SIP To header, representing the reason for the call forward.

The codes for the call forward can be defined using the following parameters:

- SuppServCodeCFU - Call Forward Unconditional
- SuppServCodeCFUDeact - Call Forward Unconditional Deactivation

- SuppServCodeCFB - Call Forward on Busy
- SuppServCodeCFBDeact - Call Forward on Busy Deactivation
- SuppServCodeCFNR - Call Forward on No Reply
- SuppServCodeCFNRDeact - Call Forward on No Reply Deactivation

Note: These codes must be defined according to the settings of the softswitch (i.e., the softswitch must recognize them).

Below is an example of an INVITE message sent by the device indicating an unconditional call forward ("*72") to extension number 100. This code is defined using the SuppServCodeCFU parameter.

```
INVITE sip:*72100@10.33.8.53;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.2.5:5060;branch=z9hG4bKWDSUKUHWFEXQSVOUVJGM
From: <sip:400@10.33.2.5;user=phone>;tag=DUOROSXSOYJJLNBFRQTG
To: <sip:*72100@10.33.8.53;user=phone>
Call-ID: GMNOVQRRXUUCYCQSFAHS@10.33.2.5
CSeq: 1 INVITE
Contact: <sip:400@10.33.2.5:5060>
Supported: em,100rel,timer,replaces
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUB
SCRIBE
User-Agent: Sip Message Generator V1.0.0.5
User-to-User: 31323334;pd=4
Content-Type: application/sdp
Content-Length: 155
```

30.7 Call Waiting

The Call Waiting feature enables FXS devices to accept an additional (second) call on busy endpoints. If an incoming IP call is designated to a busy port, the called party hears a call waiting tone (several configurable short beeps) and (for Bellcore and ETSI Caller IDs) can view the Caller ID string of the incoming call. The calling party hears a call waiting ringback tone. The called party can accept the new call using hook-flash, and can toggle between the two calls.

➢ To enable call waiting:

1. Set the parameter EnableCallWaiting to 1.
2. Set the parameter EnableHold to 1.
3. Define the Call Waiting indication and call waiting ringback tones in the Call Progress Tones file. You can define up to four call waiting indication tones (refer to the FirstCallWaitingToneID parameter).
4. To configure the call waiting indication tone cadence, modify the following parameters: NumberOfWaitingIndications, WaitingBeepDuration and TimeBetweenWaitingIndications.
5. To configure a delay interval before a Call Waiting Indication is played to the currently busy port, use the parameter TimeBeforeWaitingIndication. This enables the caller to hang up before disturbing the called party with Call Waiting Indications. Applicable only to FXS modules.

Both the calling and called sides are supported by FXS interfaces; FXO interfaces support only the calling side.

To indicate Call Waiting, the device sends a 182 Call Queued response. The device identifies Call Waiting when a 182 Call Queued response is received.

Note: The Call Waiting feature is applicable only to FXS/FXO interfaces.

30.8 Message Waiting Indication

The device supports Message Waiting Indication (MWI) according to IETF RFC 3842. The device also supports subscribing to an MWI server (using SIP SUBSCRIBE messages).

For analog interfaces: The FXS device can accept a SIP MWI NOTIFY message that indicates waiting messages or cleared messages. Users are informed of these messages by a stutter dial tone. You can define the stutter and confirmation tones in the CPT file. If the MWI display is configured, the number of waiting messages is also displayed. If the MWI lamp is configured, the phone’s lamp (on a phone that is equipped with an MWI lamp) is lit. The device can subscribe to the MWI server per port (usually used on FXS) or per device (used on FXO).

Note: For more information on configuring IP-based voice mail, refer to the *IP Voice Mail CPE Configuration Guide*.

To configure MWI, use the following parameters:

- EnableMWI
- MWIServerIP, or MWISubscribeIPGroupID and ProxySet
- MWIAnalogLamp
- MWIDisplay
- StutterToneDuration
- EnableMWISubscription
- MWIExpirationTime
- SubscribeRetryTime
- SubscriptionMode
- CallerIDType (determines the standard for detection of MWI signals)
- ETSIVMWITypeOneStandard
- BellcoreVMWITypeOneStandard
- VoiceMailInterface
- EnableVMURI

The device supports the following digital PSTN-based MWI features:

- ISDN BRI: The device supports MWI for its BRI phones, using the Euro ISDN BRI variant. When this feature is activated and a voice mail message is recorded to the mail box of a BRI extension, the softswitch sends a notification to the device. In turn, the device notifies the BRI extension and a red light flashes on the BRI extension’s phone. Once the voice message is retrieved, the MWI light on the BRI phone turns off. This is configured by setting the VoiceMailInterface parameter to 8 (“ETSI”) and enabled by the EnableMWI parameter.
- Euro-ISDN MWI: The device supports Euro-ISDN MWI for IP-to-Tel calls. The device interworks SIP MWI NOTIFY messages to Euro-ISDN Facility information element (IE)

MWI messages. This is configured by setting the VoiceMailInterface parameter to 8.

- ISDN PRI NI-2: The device support the interworking of the SIP MWI NOTIFY messages to ISDN PRI NI-2 Message Waiting Notification (MWN), sent in the ISDN Facility IE message. This is applicable when the device is connected to a PBX through an ISDN PRI trunk configured to NI-2. This is configured by setting the VoiceMailInterface parameter to [9].
- QSIG MWI: The device supports the interworking of QSIG MWI to IP (in addition to interworking of SIP MWI NOTIFY to QSIG Facility MWI messages). This provides interworking between an ISDN PBX with voice mail capabilities and a softswitch, which requires information on the number of messages waiting for a specific user. This support is configured using the TrunkGroupSettings_MWIInterrogationType parameter (in the Trunk Group Settings table), which determines the device's handling of MWI Interrogation messages. The process for sending the MWI status upon request from a softswitch is as follows:

  1. The softswitch sends a SIP SUBSCRIBE message to the device.
  2. The device responds by sending an empty SIP NOTIFY to the softswitch, and then sending an ISDN Setup message with Facility IE containing an MWI Interrogation request to the PBX.
  3. The PBX responds by sending to the device an ISDN Connect message containing Facility IE with an MWI Interrogation result, which includes the number of voice messages waiting for the specific user.
  4. The device sends another SIP NOTIFY to the softswitch, containing this MWI information.
  5. The SIP NOTIFY messages are sent to the IP Group defined by the NotificationIPGroupID parameter.

  When a change in the status occurs (e.g., a new voice message is waiting or the user has retrieved a message from the voice mail), the PBX initiates an ISDN Setup message with Facility IE containing an MWI Activate request, which includes the new number of voice messages waiting for the user. The device forwards this information to the softswitch by sending a SIP NOTIFY.

  Depending on PBX support, the MWIInterrogationType parameter can be configured to handle these MWI Interrogation messages in different ways. For example, some PBXs support only the MWI Activate request (and not MWI Interrogation request). Some support both these requests. Therefore, the device can be configured to disable this feature or enable it with one of the following support:

  - Responds to MWI Activate requests from the PBX by sending SIP NOTIFY MWI messages (i.e., does not send MWI Interrogation messages).
  - Send MWI Interrogation message, but don't use its result. Instead, wait for MWI Activate requests from the PBX.
  - Send MWI Interrogation message, use its result, and use the MWI Activate requests.

30.9 Caller ID

This section describes the device's Caller ID support.

> **Note:** The Caller ID feature is applicable only to FXS/FXO interfaces.

30.9.1 Caller ID Detection / Generation on the Tel Side

By default, generation and detection of Caller ID to the Tel side is disabled. To enable Caller ID, set the parameter EnableCallerID to 1. When the Caller ID service is enabled:

- For FXS: the Caller ID signal is sent to the device's port
- For FXO: the Caller ID signal is detected

The configuration for Caller ID is described below:

- Use the parameter CallerIDType to define the Caller ID standard. Note that the Caller ID standard that is used on the PBX or phone must match the standard defined in the device.
- Select the Bellcore caller ID sub standard using the parameter BellcoreCallerIDTypeOneSubStandard
- Select the ETSI FSK caller ID sub standard using the parameter ETSICallerIDTypeOneSubStandard
- Enable or disable (per port) the caller ID generation (for FXS) and detection (for FXO) using the 'Generate / Detect Caller ID to Tel' table (EnableCallerID). If a port isn't configured, its caller ID generation / detection are determined according to the global parameter EnableCallerID.
- EnableCallerIDTypeTwo: disables / enables the generation of Caller ID type 2 when the phone is off-hooked (used for call waiting).
- RingsBeforeCallerID: sets the number of rings before the device starts detection of caller ID (FXO only). By default, the device detects the caller ID signal between the first and second rings.
- AnalogCallerIDTimimgMode: determines the time period when a caller ID signal is generated (FXS only). By default, the caller ID is generated between the first two rings.
- PolarityReversalType: some Caller ID signals use reversal polarity and/or wink signals. In these scenarios, it is recommended to set PolarityReversalType to 1 (Hard) (FXS only).
- The Caller ID interworking can be changed using the parameters UseSourceNumberAsDisplayName and UseDisplayNameAsSourceNumber.

30.9.2 Debugging a Caller ID Detection on FXO

The following procedure describes debugging caller ID detection in FXO interfaces.

➢ **To debug a Caller ID detection on an FXO interface:**

1. Verify that the parameter EnableCallerID is set to 1.
2. Verify that the caller ID standard (and substandard) of the device matches the standard of the PBX (using the parameters CallerIDType, BellcoreCallerIDTypeOneSubStandard, and ETSICallerIDTypeOneSubStandard).

3. Define the number of rings before the device starts the detection of caller ID (using the parameter RingsBeforeCallerID).
4. Verify that the correct FXO coefficient type is selected (using the parameter CountryCoefficients), as the device is unable to recognize caller ID signals that are distorted.
5. Connect a phone to the analog line of the PBX (instead of to the device's FXO interface) and verify that it displays the caller ID.

If the above does not solve the problem, you need to record the caller ID signal (and send it to AudioCodes), as described below.

➢ **To record the caller ID signal using the debug recording mechanism:**

1. Access the FAE page (by appending "FAE" to the device's IP address in the Web browser's URL, for example, http://10.13.4.13/FAE).
2. Press the **Cmd Shell** link.
3. Enter the following commands:

```
dr
ait <IP address of PC to collect the debug traces sent from
the device>
AddChannelIdTrace ALL-WITH-PCM <port number, which starts from
0>
Start
```

4. Make a call to the FXO.
5. To stop the DR recording, at the CLI prompt, type **STOP**.

30.9.3 Caller ID on the IP Side

Caller ID is provided by the SIP From header containing the caller's name and "number", for example:

```
From: “John” <SIP:101@10.33.2.2>;tag=35dfsgasd45dg
```

If Caller ID is restricted (received from Tel or configured in the device), the From header is set to:

```
From: “anonymous” <anonymous@anonymous.invalid>; tag=35dfsgasd45dg
```

The P-Asserted (or P-Preferred) headers are used to present the originating party’s caller ID even when the caller ID is restricted. These headers are used together with the Privacy header.

- If Caller ID is restricted:
  - The From header is set to “anonymous” <anonymous@anonymous.invalid>
  - The ‘Privacy: id’ header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID
- If Caller ID is allowed:
  - The From header shows the caller ID
  - The ‘Privacy: none’ header is included
  - The P-Asserted-Identity (or P-Preferred-Identity) header shows the caller ID

The caller ID (and presentation) can also be displayed in the Calling Remote-Party-ID header.

The ‘Caller Display Information’ table (CallerDisplayInfo) is used for the following:

- **FXS interfaces** - to define the caller ID (per port) that is sent to IP.
- **FXO interfaces** - to define the caller ID (per port) that is sent to IP if caller ID isn’t detected on the Tel side, or when EnableCallerID = 0.

- **FXS and FXO interfaces -** to determine the presentation of the caller ID (allowed or restricted).
- **To maintain backward compatibility -** when the strings 'Private' or 'Anonymous' are set in the Caller ID/Name field, the caller ID is restricted and the value in the Presentation field is ignored.

The value of the 'Presentation' field that is defined in the 'Caller Display Information' table can be overridden by configuring the 'Presentation' parameter in the 'Tel to IP Source Number Manipulation' table. Therefore, this table can be used to set the presentation for specific calls according to Source / Destination prefixes.

The caller ID can be restricted/allowed (per port) using keypad features KeyCLIR and KeyCLIRDeact (FXS only).

AssertedIdMode defines the header that is used (in the generated INVITE request) to deliver the caller ID (P-Asserted-Identity or P-Preferred-Identity). Use the parameter UseTelURIForAssertedID to determine the format of the URI in these headers (sip: or tel:).

The parameter EnableRPIheader enables Remote-Party-ID (RPI) headers for calling and called numbers for Tel-to-IP calls.

30.10 Three-Way Conferencing

The device supports three-way conference calls. Multiple, concurrent three-way conference calls are also supported. The device supports the following conference modes:

- **Conference Managed by External, AudioCodes Conferencing (Media) Server:** The conference-initiating INVITE sent by the device uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. This mode is configured by setting the 3WayConferenceMode parameter to 0 (default.)
- **Conference Managed by External, Third-party Conferencing Server:** Two optional modes of operation:
  - The conference-initiating INVITE sent by the device uses only the ConferenceID as the Request-URI. The Conferencing server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is included (by the device) in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the Conferencing server using this conference URI. This mode is configured by setting the 3WayConferenceMode parameter to 1.
  - The conference-initiating INVITE sent by the device uses only the ConferenceID as the Request-URI. The Conferencing server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. The Conference URI is included in the URI of the REFER with a Replaces header sent by the device to the Conferencing server. The Conferencing server then sends an INVITE with a Replaces header to the remote participants. This mode is configured by setting the 3WayConferenceMode parameter to 3.
- **Local, On-board Conferencing:** The conference is established on the device without the need for an external Conferencing server. This feature includes local mixing and transcoding of the 3-Way Call legs on the device, and even allowing multi-codec conference calls. The number of simultaneous, on-board conferences can be limited using the MaxInBoardConferenceCalls parameter. The device supports up to five simultaneous, on-board, three-way conference calls. This mode is configured by setting the 3WayConferenceMode parameter to 2.

Notes:

- Three-way conferencing using an external conference server is supported only by FXS interfaces.
- Instead of using the flash-hook button to establish a three-way conference call, you can dial a user-defined hook-flash code (e.g., "*1"), configured by the HookFlashCode parameter.
- Three-way conferencing is applicable only to FXS and BRI interfaces.
- Three-way conferencing support for the BRI phones connected to the device complies with ETS 300 185.
- The device supports high definition, three-way conferencing using wideband codecs (e.g., G.722 and AMR-WB). This allows conference participants to experience wideband voice quality. Call conferences can also include narrowband and wideband participants.

The following example demonstrates three-way conferencing using the device's local, on-board conferencing feature. In this example, telephone "A" connected to the device establishes a three-way conference call with two remote IP phones, "B" and "C":

1. A establishes a regular call with B.
2. A places B on hold, by pressing the telephone's flash-hook button and the number "1" key.
3. A hears a dial tone and then makes a call to C.
4. C answers the call.
5. A establishes a three-way conference call with B and C, by pressing the flash-hook button and the number "3" key.

To configure local, on-board three-way conferencing:

1. Open the Supplementary Services page.
2. Set 'Enable 3-Way Conference' to **Enable** (Enable3WayConference = 1).
3. Set '3-Way Conference Mode' to **On Board** (3WayConferenceMode = 2).
4. Set 'Flash Keys Sequence Style' to **Sequence 1** or **Sequence 2** (FlashKeysSequenceStyle = 1 or 2).

30.11 Emergency E911 Phone Number Services

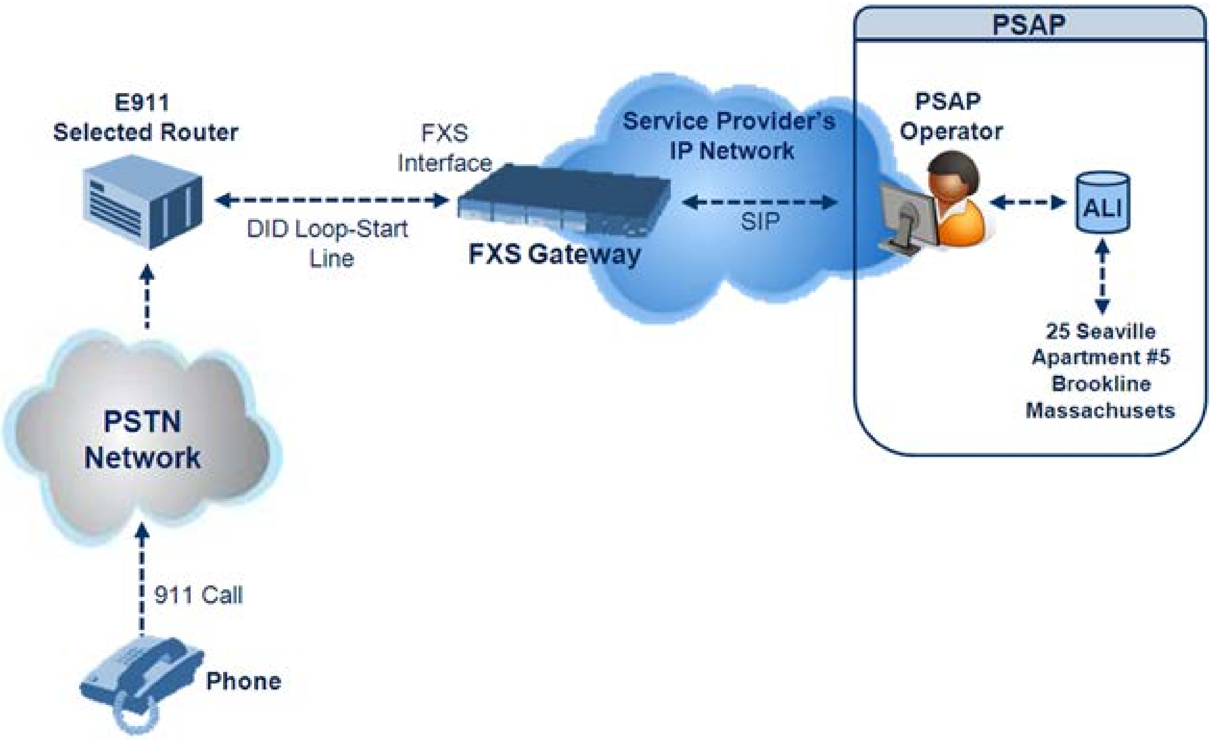
This section describes the device's support for emergency phone number services. The device supports the North American emergency telephone number system known as Enhanced 911 (E911), according to the TR-TSY-000350 and Bellcore's GR-350-Jun2003 standards. The E911 emergency system automatically associates a physical address with the calling party's telephone number, and routes the call to the most appropriate (closest) Public Safety Answering Point (PSAP), allowing the PSAP to quickly dispatch emergency response (e.g., police) to the caller's location.

Typically, the dialed emergency number is routed to the appropriate PSAP by the telephone company's switch, known as a 911 Selective Router (or E911 tandem switch). If the PSAP receives calls from the telephone company on old-style digital trunks, they are specially formatted Multi-Frequency (MF) trunks that pass only the calling party's number (known as Automatic Number Identification - ANI). Once the PSAP receives the call, it searches for the physical address that is associated with the calling party's telephone number (in the Automatic Location Identification database - ALI).

30.11.1 FXS Device Emulating PSAP using DID Loop-Start Lines

The device's FXS interface can be configured to emulate PSAP (using DID loop start lines), according to the Telcordia GR-350-CORE specification.

Figure 30-3: FXS Device Emulating PSAP using DID Loop-Start Lines



The call flow of an E911 call to the PSAP is as follows:

1. The E911 tandem switch seizes the line.
2. The FXS device detects the line seize, and then generates a wink signal (nominal 250 msec). The wink can be delayed by configuring the parameter DelayBeforeDIDWink to 200 (for 200 msec or a higher value).
3. The switch detects the wink and then sends the MF Spill digits with ANI and (optional) Pseudo-ANI (P ANI).
4. The FXS device collects the MF digits, and then sends a SIP INVITE message to the

PSAP with all collected MF digits in the SIP From header as one string.

5. The FXS device generates a mid-call wink signal (two subsequent polarity reversals) toward the E911 tandem switch upon either detection of an RFC 2833 "hookflash" telephony event, or if a SIP INFO message with a "hooflash" body is received from the PSAP (see the example below). The duration of this "flashhook" wink signal is configured using the parameter FlashHookPeriod (usually 500 msec). Usually the wink signal is followed by DTMF digits sent by PSAP to perform call transfer. Another way to perform the call transfer is to use SIP REFER messages, as described below.
6. The FXS device supports call transfer initiated by the PSAP. If it receives a SIP REFER message with the Refer-To URI host part containing an IP address that is equal to the device's IP address, the FXS device generates a 500-msec wink signal (double polarity reversals), and then (after a user-defined interval configured by the parameter WaitForDialTime), plays DTMF digits according to the transfer number received in the SIP Refer-To header URI userpart.
7. When the call is answered by the PSAP operator, the PSAP sends a SIP 200 OK to the FXS device, and the FXS device then generates a polarity reversal signal to the E911 switch.
8. After the call is disconnected by the PSAP, the PSAP sends a SIP BYE to the FXS device, and the FXS device reverses the polarity of the line toward the tandem switch.

The following parameters need to be configured:

- EnableDIDWink = 1
- EnableReversalPolarity = 1
- PolarityReversalType = 1
- FlashHookPeriod = 500 (for 500 msec "hookflash" mid-call Wink)
- WinkTime = 250 (for 250 msec signalling Wink generated by the FXS device after it detects the line seizure)
- EnableTransfer = 1 (for call transfer)
- LineTransferMode = 1 (for call transfer)
- WaitforDialTime = 1000 (for call transfer)
- SwapTEl2IPCalled&CallingNumbers = 1
- DTMFDetectorEnable = 0
- MFR1DetectorEnable = 1
- DelayBeforeDIDWink = 200 (for 200 msec) - can be configured in the range from 0 (default) to 1000.

Note: Modification of the WinkTime parameter requires a device reset.

The outgoing SIP INVITE message contains the following headers:

```
INVITE sip:Line@DomainName
From: <sip:*81977820#@sipgw>;tag=1c143
To: <sip:Line@DomainName>
```

Where:

- *Line* = as configured in the Endpoint Phone Number Table
- *SipGtw* = configured by the SIPGatewayName parameter
- *From* header/user part = calling party number as received from the MF spill

The ANI and the pseudo-ANI numbers are sent to the PSAP either in the From and/or P-AssertedID SIP header.

Typically, the MF spills are sent from the E911 tandem switch to the PSAP, as shown in the table below:

Table 30-1: Dialed MF Digits Sent to PSAP

Digits of Calling Number	Dialed MF Digits
8 digits "nnnnnnnn" (ANI)	"KPnnnnnnnnST"
12 digits "nnnnnnnnnnnn" (ANI)	"KPnnnnnnnnnnnnSTP"
12 digits ANI and 10 digits PANI	"KPnnnnnnnnnnnnSTKPmmmmmmmmmmST"
two digits "nn"	"KPnnSTP"

The MF KP, ST, and STP digits are mapped as follows:

- * for KP
- # for ST
- B for STP

For example, if ANI and PANI are received, the SIP INVITE contains the following From header:

```
From: <sip:*nnnnnnnnnnnn#*mmmmmmmmmm#@10.2.3.4>;tag=1c14
```

Note: It is possible to remove the * and # characters, using the device's number manipulation rules.

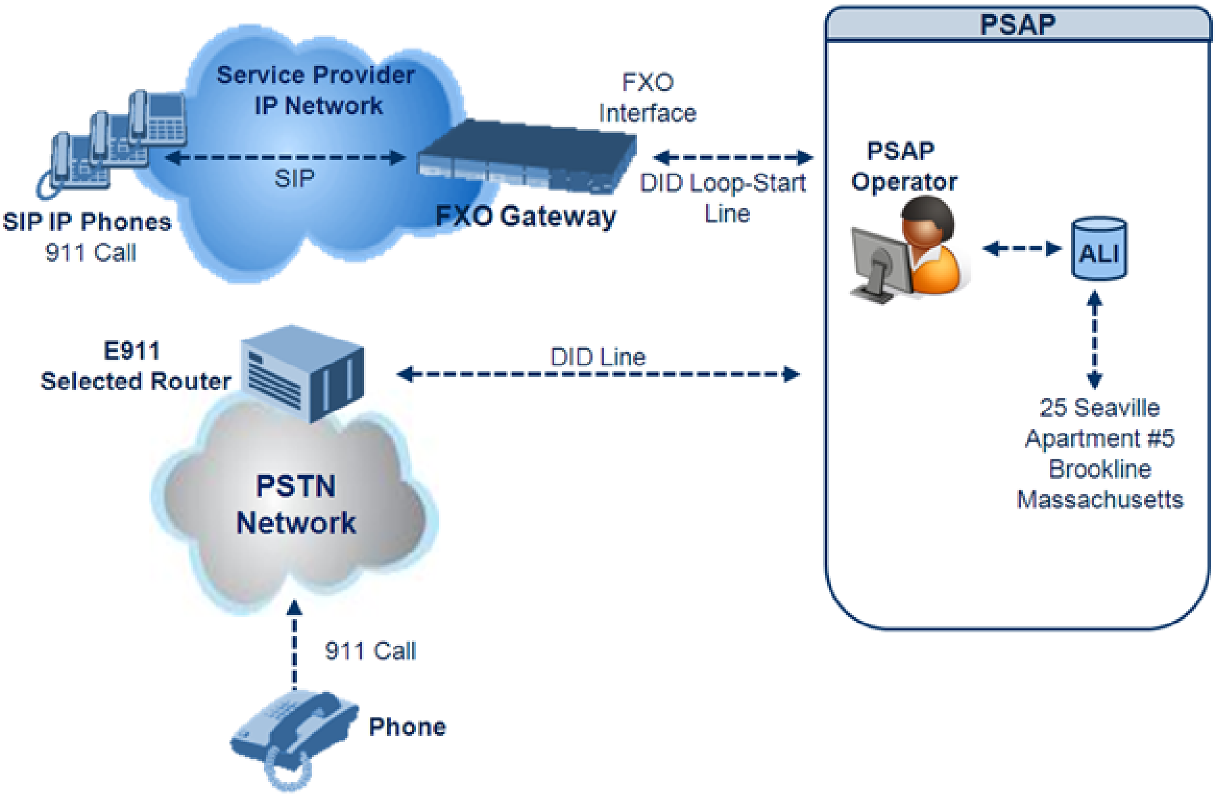
If the device receives the SIP INFO message below, it then generates a "hookflash" mid-call Wink signal:

```
INFO sip:4505656002@192.168.13.40:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.13.2:5060
From: port1vega1 <sip:06@192.168.13.2:5060>
To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-
1040067870294
Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2
CSeq:2 INFO
Content-Type: application/broadsoft
Content-Length: 17
event flashhook
```

30.11.2 FXO Device Interworking SIP E911 Calls from Service Provider's IP Network to PSAP DID Lines

The device's FXO interface can interwork SIP emergency E911 calls from the Service Provider's IP network to the analog PSAP DID lines. The standards that define this interface include TR-TSY-000350 or Bellcore’s GR-350-Jun2003. This protocol defines signaling between the E911 tandem switch (E911 Selective Router) and the PSAP, using analog loop-start lines. The FXO device can be implemented instead of an E911 switch, by connecting directly to the PSAP DID loop-start lines.

Figure 30-4: FXO Device Interfacing between E911 Switch and PSAP



When an IP phone subscriber dials 911, the device receives the SIP INVITE message and makes a call to the PSAP as follows:

1. The FXO device seizes the line.
2. PSAP sends a Wink signal (250 msec) to the device.
3. Upon receipt of the Wink signal, the device dials MF digits after a user-defined time (WaitForDialTime) containing the caller's ID (ANI) obtained from the SIP headers From or P-Asserted-Identity.
4. When the PSAP operator answers the call, the PSAP sends a polarity reversal to the device, and the device then sends a SIP 200 OK to the IP side.
5. After the PSAP operator disconnects the call, the PSAP reverses the polarity of the line, causing the device to send a SIP BYE to the IP side.
6. If, during active call state, the device receives a Wink signal (typically of 500 msec) from the PSAP, the device generates a SIP INFO message that includes a "hookflash" body, or sends RFC 2833 hookflash Telephony event (according to the HookFlashOption parameter).
7. Following the "hookflash" Wink signal, the PSAP sends DTMF digits. These digits are detected by the device and forwarded to the IP, using RFC 2833 telephony events (or

inband, depending on the device's configuration). Typically, this Wink signal followed by the DTMF digits initiates a call transfer.

For supporting the E911 service, used the following configuration parameter settings:

- Enable911PSAP = 1 (also forces the EnableDIDWink and EnableReversalPolarity)
- HookFlashOption = 1 (generates the SIP INFO hookflash message) or 4 for RFC 2833 telephony event
- WinkTime = 700 (defines detection window of 50 to 750 msec for detection of both winks - 250 msec wink sent by the PSAP for starting the device's dialing; 500 msec wink during the call)
- IsTwoStageDial = 0
- EnableHold = 0
- EnableTransfer = 0
    - Use RFC 2833 DTMF relay:
        - RxDTMFOption = 3
        - TxDTMFOption = 4
        - RFC2833PayloadType = 101
- TimeToSampleAnalogLineVoltage = 100
- WaitForDialTime = 1000 (default is 1 sec)
- SetDefaultLinePolarityState = 0 (you need to verify that the RJ-11 two-wire cable is connected without crossing, Tip to Tip, Ring to Ring. Typically, the Tip line is positive compared to the Ring line.)

Note: If the two-wire cable is crossed, the SetDefaultLinePolarityState parameter must be set to 1.

The device expects to receive the ANI number in the From and/or P-Asserted-Identity SIP header. If the pseudo-ANI number exists, it should be sent as the display name in these headers.

Table 30-2: Dialed Number by Device Depending on Calling Number

Digits of Calling Number (ANI)	Digits of Displayed Number	Number Dialed MF Digits
8 "nnnnnnnn"	-	MF dialed "KPnnnnnnnnST"
12 "nnnnnnnnnnnn"	None	"KPnnnnnnnnnnnnSTP"
12 "nnnnnnnnnnnn"	10 "mmmmmmmmmm" (pANI)	"KPnnnnnnnnnnnnSTKPmmmmmmmmmmST"
2 "nn"	None	"KPnnSTP"
1 "n"	-	MF dialed "KPnST" For example: "From: <sip:8>@xyz.com>" generates device MF spill of KP 8 ST

Table notes:

- For all other cases, a SIP 484 response is sent.

- KP is for .
- ST is for #.
- STP is for B.

The MF duration of all digits, except for the KP digit is 60 msec. The MF duration of the KP digit is 120 msec. The gap duration is 60 msec between any two MF digits.

Notes:

- Manipulation rules can be configured for the calling (ANI) and called number (but not on the "display" string), for example, to strip 00 from the ANI "00INXXYYYYY".
- The called number, received as userpart of the Request URI ("301" in the example below), can be used to route incoming SIP calls to FXO specific ports, using the TrunkGroup and PSTNPrefix parameters.
- When the PSAP party off-hooks and then immediately on-hooks (i.e., the device detects wink), the device releases the call sending SIP response "403 Forbidden" and the release reason 21 (i.e., call rejected) "Reason: Q.850 ;cause=21" is sent. Using the cause mapping parameter, it is possible to change the 403 to any other SIP reason, for example, to 603.
- Sometimes a wink signal sent immediately after the FXO device seizes the line is not detected. To overcome this problem, configure the parameter TimeToSampleAnalogLineVoltage to 100 (instead of 1000 msec, which is the default value). The wink is then detected only after this timeout + 50 msec (minimum 150 msec).

Below are two examples for a) INVITE messages and b) INFO messages generated by hook-flash.

- **Example A:** INVITE message with ANI = 333333444444 and pseudo-ANI = 0123456789:

```
INVITE sip:301@10.33.37.79;user=phone SIP/2.0
Via: SIP/2.0/UDP 10.33.37.78;branch=z9hG4bKac771627168
Max-Forwards: 70
From: "0123456789"
<sip:333333444444@audiocodes.com>;tag=1c771623824
To: <sip:301@10.33.37.79;user=phone>
Call-ID: 77162335841200014153@10.33.37.78
CSeq: 1 INVITE
Contact: <sip:101@10.33.37.78>
Supported: em,100rel,timer,replaces,path
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO
,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway-FXO/v.6.80A.227.005
Privacy: none
P-Asserted-Identity: "0123456789"
<sip:3333344444@audiocodes.com>
Content-Type: application/sdp
Content-Length: 253
v=0
o=AudiocodesGW 771609035 771608915 IN IP4 10.33.37.78
s=Phone-Call
c=IN IP4 10.33.37.78
t=0 0
m=audio 4000 RTP/AVP 8 0 101
a=rtpmap:8 pcma/8000
a=rtpmap:0 pcmu/8000
```

```
a=rtpmap:101 telephone-event/8000
a=fmtp:101 0-15
a=ptime:20
a=sendrecv
```

- **Example B:** The detection of a Wink signal generates the following SIP INFO message:

```
INFO sip:4505656002@192.168.13.40:5060 SIP/2.0
Via:  SIP/2.0/UDP 192.168.13.2:5060
From: port1vega1 <sip:06@192.168.13.2:5060>
To: <sip:4505656002@192.168.13.40:5060>;tag=132878796-
1040067870294
Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2
CSeq:2 INFO
Content-Type: application/broadsoft
Content-Length: 17
event flashhook
```

30.11.3 Pre-empting Existing Calls for E911 IP-to-Tel Calls

If the device receives an E911 call from the IP network destined to the Tel, and there are unavailable channels (e.g., all busy), the device terminates one of the calls (arbitrary) and then sends the E911 call to that channel. The preemption is done only on a channel pertaining to the same Trunk Group for which the E911 call was initially destined and if the channel select mode (configured by the ChannelSelectMode parameter) is set to a value other than “By Dest Phone Number” (0).

The preemption is done only if the incoming IP-to-Tel call is identified as an emergency call. The device identifies emergency calls by one of the following:

- The destination number of the IP call matches one of the numbers defined by the EmergencyNumbers parameter. For E911, you must defined this parameter with the value "911".
- The Priority header of the incoming SIP INVITE message contains the “emergency” value.

Emergency pre-emption of calls can be enabled for all calls, using the global parameter CallPriorityMode, or for specific calls using the Tel Profile parameter CallPriorityMode.

Notes:

- For Trunk Groups configured with call preemption, all must be configured to MLPP [1] or all configured to Emergency [2]. In other words, you cannot set some trunks to [1] and some to [2].
- The global parameter must be set to the same value as that of the Tel Profile parameter; otherwise, the Tel Profile parameter is not applied.
- If you configure call preemption using the global parameter and a new Tel Profile is subsequently added, the TelProfile_CallPriorityMode parameter automatically acquires the same setting as well.
- This feature is applicable to FXO, CAS and ISDN interfaces.
- For FXO interfaces, the preemption is done only on existing IP-to-Tel calls. In other words, if all the current FXO channels are busy with calls that were answered by the FXO device (i.e., Tel-to-IP calls), new incoming emergency IP-to-Tel calls are rejected.

30.11.4 Enhanced 9-1-1 Support for Lync Server 2010

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.

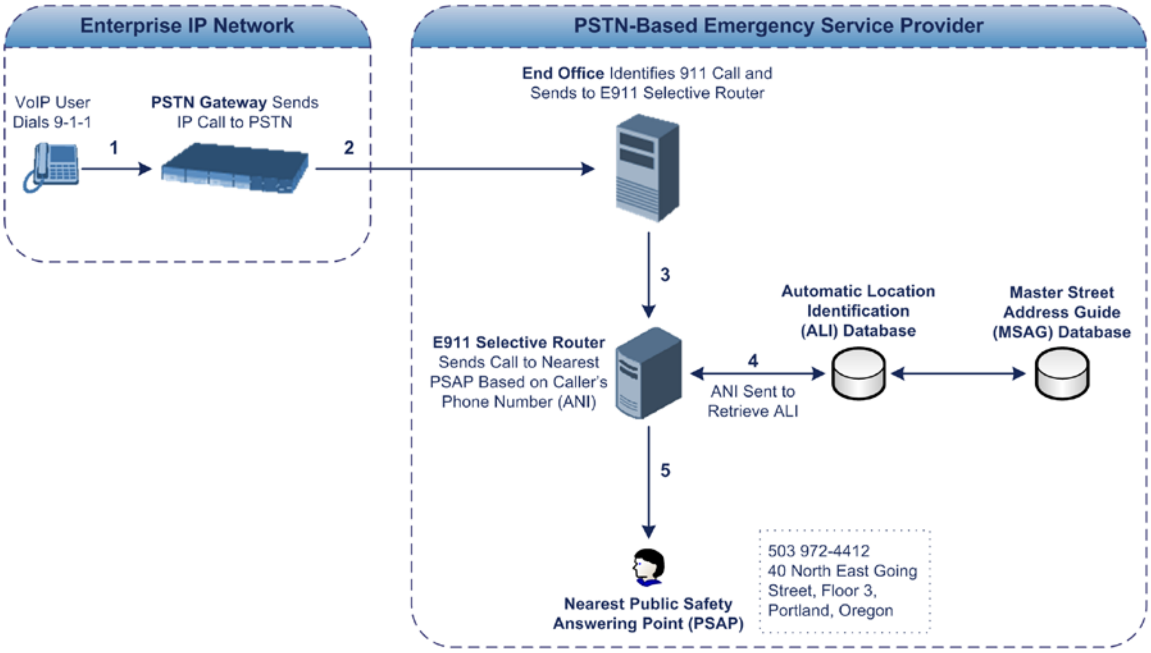
This section describes the E9-1-1 solution provided by Microsoft Lync Server 2010 (hereafter referred to as *Lync Server 2010*), and the deployed AudioCodes ELIN Gateway which provides the ISDN (or CAMA) connectivity to the PSTN-based E9-1-1 emergency providers. This section also describes the configuration of AudioCodes ELIN Gateway for interoperating between the Lync Server 2010 environment and the E9-1-1 emergency provider.

30.11.4.1 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:

Figure 30-5: Call Flow of E9-1-1 to PSTN-Based Emergency Services Provider


Enterprise IP Network
PSTN-Based Emergency Service Provider
VoIP User Dials 9-1-1
PSTN Gateway Sends IP Call to PSTN
1
2
End Office Identifies 911 Call and Sends to E911 Selective Router
3
E911 Selective Router Sends Call to Nearest PSAP Based on Caller's Phone Number (ANI)
4
ANI Sent to Retrieve ALI
Automatic Location Identification (ALI) Database
Master Street Address Guide (MSAG) Database
5
Nearest Public Safety Answering Point (PSAP)
503 972-4412
40 North East Going Street, Floor 3, Portland, Oregon

1. The VoIP user dials 9-1-1.
2. The call is eventually sent to the PSTN through a PSTN Gateway.

3. The PSTN 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.

30.11.4.2 Microsoft Lync Server 2010 and E9-1-1

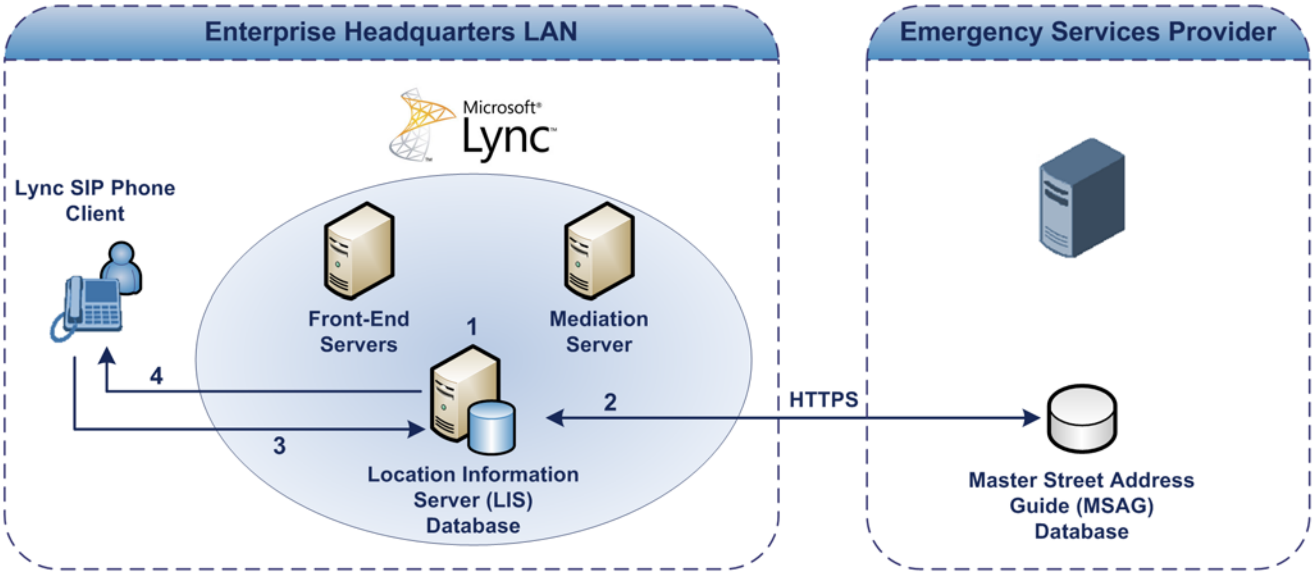
Microsoft Lync Server 2010 enables Enterprise voice users to access its unified communications platform from virtually anywhere and through many different devices. This, together with a deployed MLTS, poses a challenge for E9-1-1 due to the difficulty in accurately locating the E9-1-1 caller. However, Lync Server 2010 offers an innovative solution to solving Enterprises E9-1-1 location problems.

30.11.4.2.1 Gathering Location Information of Lync 2010 Clients for 911 Calls

When a Microsoft® Lync™ 2010 client (hereafter referred to as *Lync 2010 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 2010 client registration process or when the operating system detects a network connection change, each Lync 2010 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 2010 client dials 9-1-1, this location information is then included as part of the emergency call and used by the Emergency Services provider to route the call to the correct PSAP.

The gathering of location information in the Lync Server 2010 network is illustrated in the figure below:

Figure 30-6: Microsoft Lync Server 2010 Client Acquiring Location Information


Enterprise Headquarters LAN
Emergency Services Provider
Microsoft Lync
Lync SIP Phone Client
Front-End Servers
Mediation Server
1
4
2
HTTPS
3
Location Information Server (LIS) Database
Master Street Address Guide (MSAG) Database

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 452.
2. The Administrator validates addresses with the Emergency Services 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 2010 client initiates a location request to the LIS under the following circumstances:
   - Immediately after startup and registering the user with Lync Server 2010
   - Approximately every four hours after initial registration
   - Whenever a network connection change is detected (such as roaming to a new WAP)

   The Lync 2010 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 2010 client that moves inside the corporate network such as a soft phone on a laptop that connects wirelessly to the corporate network, Lync Server 2010 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 2010 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.

30.11.4.2.2 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 30-3: 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 Gateway, the Administrator must add the ELIN number to the <CompanyName> column (shown in the table above in **bold** typeface). As the ELIN Gateway 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 (xxxxxxxxx).

When the ELIN Gateway 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 2010 location policy, Location Required is set to **Yes** or **Disclaimer**, the user will be prompted to enter a location manually.

30.11.4.2.3 Passing Location Information to the PSTN Emergency Provider

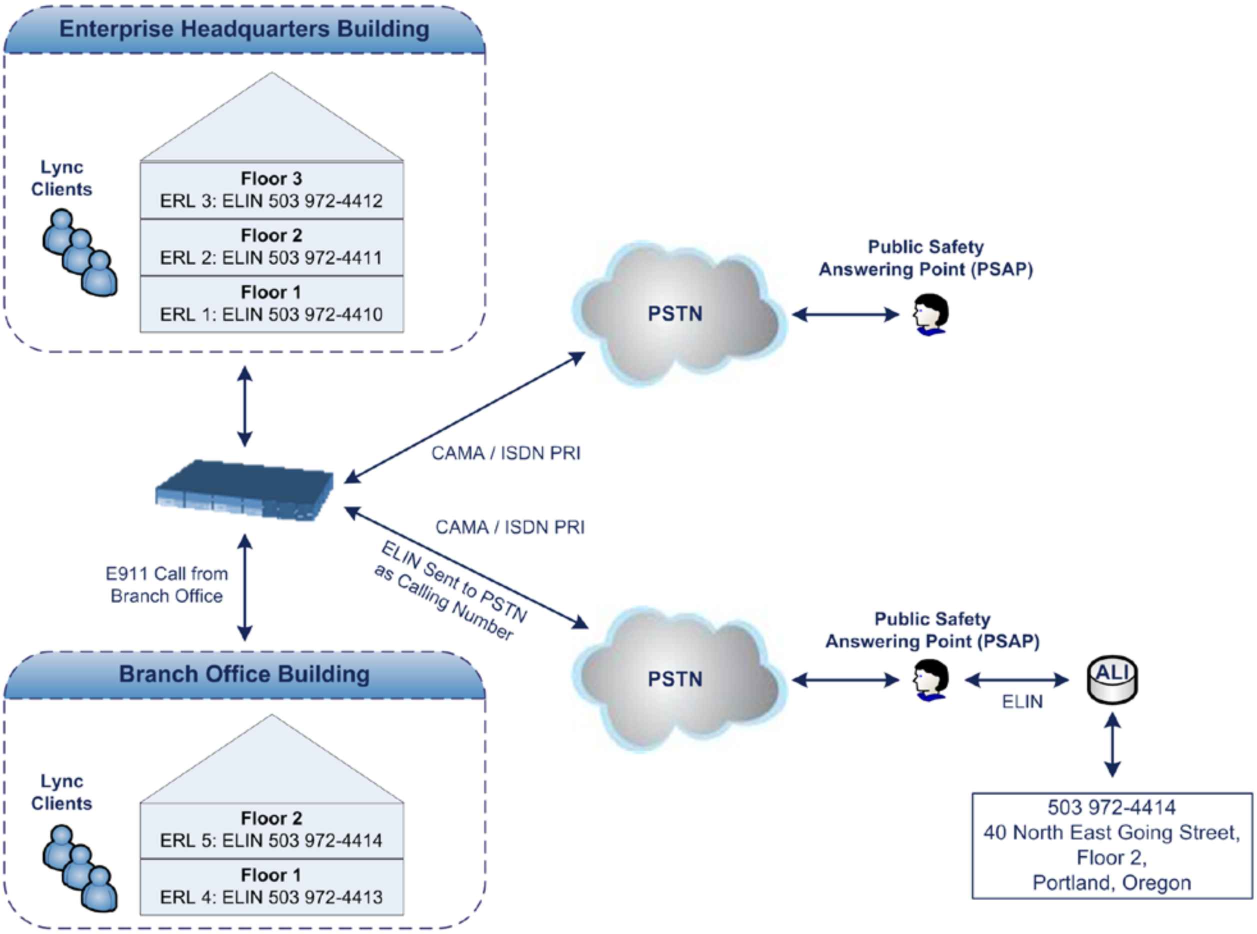
When a Lync 2010 client, enabled for E9-1-1 emergency services, dials 9-1-1, the location data and callback information stored on the client is sent with the call through the Mediation Server to a PSTN-based Emergency Services provider. The Emergency Services provider then routes the call to the nearest and most appropriate PSAP based on the location information contained within the call.

Lync Server 2010 passes the location information of the Lync 2010 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 PSTN network using ISDN PRI due to protocol limitations. To overcome this, Enterprises using PSTN Gateways 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 2010 sends a SIP INVITE message with the PIDF-LO to the PSTN Gateway, it can parse the content and translate the calling number to an appropriate ELIN. The PSTN Gateway then sends the call to the PSTN with the ELIN number as the calling number. This ELIN number is sent to the Emergency Services 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 30-7: Implementing ERLs and ELINs for E9-1-1 in Lync Server 2010



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 30-4: 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

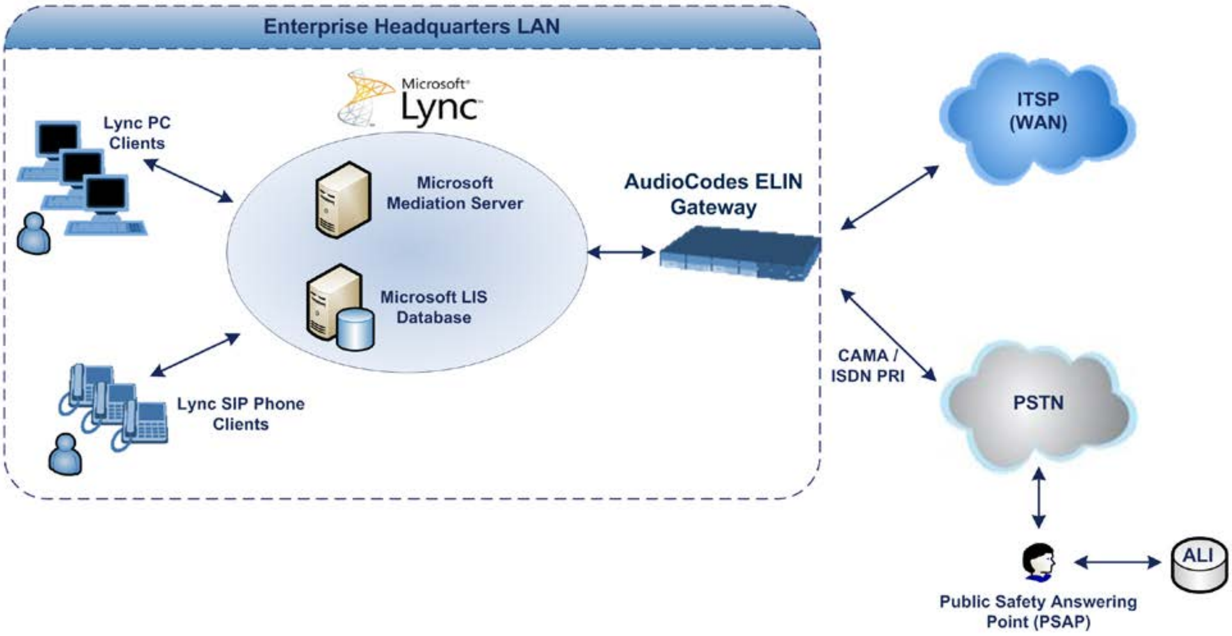
In the table above, a unique IP subnet is associated per ERL. This is useful if you implement different subnets between floors. Therefore, IP phones, for example, on a specific floor are in the same subnet and therefore, use the same ELIN when dialing 9-1-1.

30.11.4.3 AudioCodes ELIN Gateway for Lync Server 2010 E9-1-1 Calls to PSTN

The 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 PSTN network using ISDN PRI due to protocol limitations. To solve this issue, Lync Server 2010 requires a PSTN Gateway (*ELIN Gateway*) to send the E9-1-1 call to the PSTN. When Lync Server 2010 sends the PIDF-LO to the PSTN Gateway, 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 Services provider's ALI database.

The figure below illustrates an AudioCodes ELIN Gateway deployed in the Lync Server 2010 environment for handling E9-1-1 calls between the Enterprise and the PSTN.

Figure 30-8: AudioCodes ELIN Gateway for E9-1-1 in Lync Server 2010 Environment



30.11.4.3.1 Detecting and Handling E9-1-1 Calls

The ELIN Gateway identifies E9-1-1 calls and translates their incoming E9-1-1 calling numbers into ELIN numbers, sent toward the PSAP. The ELIN Gateway handles the received E9-1-1 calls as follows:

1. The ELIN Gateway 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 ELIN Gateway 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 ELIN Gateway 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 (xxxxxxxxx), as shown below:

   ```
   <ca:NAM>1111-222-333; 1234567890 </ca:NAM>
   ```

3. The ELIN Gateway 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 2010 Clients for Dropped E9-1-1 Calls" on page 457. 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 this 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 459), 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 depend on the AudioCodes ELIN Gateway deployed in the Lync Server 2010 environment:

- **Mediant 1000 Series and Mediant 2000:** 100 entries
- **Mediant 3000:** 300 entries

4. The ELIN Gateway uses the ELIN number as the E9-1-1 calling number and sends it in the ISDN Setup message (as an ANI / Calling Party Number) to the PSTN.

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--
```

30.11.4.3.2 Pre-empting Existing Calls for E9-1-1 Calls

If the ELIN Gateway receives an E9-1-1 call from the IP network and there are unavailable channels (for example, all busy), the ELIN Gateway 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 channel pertaining to the same Trunk Group for which the E9-1-1 call was initially destined. For example, if an E9-1-1 call is destined for Trunk Group #2 and all the channels belonging to this group are busy, the ELIN Gateway terminates one of the calls in this group to free a channel for accepting the E9-1-1 call.

This feature is initiated only if the received SIP INVITE message contains a *Priority* header set to "emergency", as shown below:

```
PRIORITY: emergency
```

30.11.4.3.3 PSAP Callback to Lync 2010 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 ELIN Gateway to translate the ELIN number back into the E9-1-1 caller's extension number.

In the ELIN table of the ELIN Gateway, 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 ELIN Gateway 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 ELIN Gateway is as follows:

1. When the ELIN Gateway receives any call from the PSTN, it searches the ELIN table for an ELIN that corresponds to the received Called Party Number in the incoming PSTN call.
2. If a match is found in the ELIN table, it routes the call to the Mediation Sever 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 ELIN Gateway updates the Time in the ELIN table. (The Count is not affected).

The PSAP callback can be done only within a user-defined timeout (see "Configuring the E9-1-1 Callback Timeout" on page 459) 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 ELIN Gateway 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, then the PSAP callback is routed to this new E9-1-1 caller.

In scenarios where the same ELIN number is being used by multiple E9-1-1 callers, upon receipt of a PSAP callback, the ELIN Gateway 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 ELIN Gateway sends it to the E9-1-1 caller with phone number "4258359555".

Table 30-5: Choosing Caller of ELIN

ELIN	Time	Call From
4257275678	11:00	4258359333
4257275678	11:01	4258359444
4257275678	11:03	**4258359555**

30.11.4.3.4 Selecting ELIN for Multiple Calls within Same ERL

The ELIN Gateway 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 ELIN Gateway sends the ELIN number as the E9-1-1 calling number to the PSTN-based emergency provider. If the XML message contains more than one ELIN number, the ELIN Gateway 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 ELIN Gateway 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 ELIN Gateway 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 ELIN Gateway 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 4257**2**75678 was terminated at **11:01** and E9-1-1 caller using ELIN 425727567**0** was terminated at **11:03**, then the ELIN Gateway selects ELIN 425727567**8**.

In this scenario, multiple E9-1-1 calls will be sent with the same ELIN.

30.11.4.3.5 Location Based Emergency Routing

The device supports location-based emergency routing (E-911) in Lync Server 2010. This ensures that E-911 calls from remote branches are routed to emergency providers that are relevant to the geographical area in which the remote branch callers are physically located.

To support this, the device enables routing and SIP header / number manipulation of such emergency calls based on the geographical location of the caller. The device manipulates the received destination number (i.e., 911) from the remote branch callers, into a destination number of an emergency provider that is relevant to the geographical area in which the remote branch office is located.

30.11.4.4 Configuring AudioCodes ELIN Gateway

This section describes E9-1-1 configuration of the AudioCodes ELIN Gateway deployed in the Lync Server 2010 environment.

30.11.4.4.1 Enabling the E9-1-1 Feature

By default, the E9-1-1 feature in the ELIN Gateway for Lync Server 2010 is disabled. To enable it, the following *ini* file parameter setting must be done:

```
E911Gateway = 1
```

30.11.4.4.2 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 interval, by using the following *ini* file parameter:

```
E911CallbackTimeout = <time value>   ; where <time value > can be
any value from 0 through 60
```

30.11.4.4.3 Configuring the SIP Release Cause Code for Failed E9-1-1 Calls

When a Lync 2010 client makes an emergency call, the call is routed through the Microsoft Mediation Server to the ELIN Gateway, which sends it on to the PSTN. In some scenarios, the call may not be established due to either the PSTN (for example, the destination is busy or not found) or ELIN Gateway (for example, lack of resources or an internal error). In such a scenario, the Mediation Server requires that the ELIN Gateway "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 Gateway), instead of retrying the call or returning the release call to the user.

To support this requirement, the ELIN Gateway can be configured to send the 503 "Service Unavailable" release cause code instead of SIP 4xx if an emergency call cannot be established. To enable this support, the following *ini* file parameter setting must be done:

Note: This can also be configured using the *ini* file parameter, EmergencySpecialReleaseCause.

➢ **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, select **Enable**.

30.11.4.4.4 Configuring Location-Based Emergency Routing

The device identifies the geographical location of emergency callers by their ELIN numbers, which is present in the PIDF-LO XML body of received SIP INVITE messages. Therefore, you need to configure the device to route emergency calls to a destination (i.e., emergency center such as police) that is appropriate to the caller's ELIN number. As the destination of incoming calls is the emergency number (e.g., 999), the device needs to manipulate the destination number to a number that represents the caller's local emergency center (e.g., +4420999 for London police).

To add manipulation rules for location-based emergency routing, you need to use the Destination Phone Number Manipulation Table for IP-to-Tel Calls table. In this table, you need to use the ELIN number (e.g., 5000) as the source prefix, with the "ELIN" string value added in front of it (e.g., ELIN5000) which is used internally by the device to identify the number as an ELIN number (and **not** used for any other routing processes etc.). For each corresponding ELIN source number prefix entry, you need to configure the manipulation action required on the destination number so that the call is routed to the appropriate destination.

Following is an example of how to configure location-based emergency routing:

- **Assumptions:**
  - Company with offices in different cities -- London and Manchester.
  - Each city has its local police department.
  - In an emergency, users need to dial 999.
  - Company employs Microsoft Lync for communication between employers, and between employers and the external telephone network (PSTN). In other words, all employers are seemingly (virtual) in the same location in respect to the IP network.
  - ELIN numbers are used to identify the geographical location of emergency calls dialed by users:
    - London ELIN is 5000.
    - Manchester ELIN is 3000.
- **Configuration Objectives:**
  - Emergency calls received from London office users are routed by the device to the London police department, which is +4420999.
  - Emergency calls received from Manchester office users are routed by the device to the Manchester police department, which is +44161999.

  The international code, +44 for England is used for IP routing considerations, but can be omitted depending on your specific network environment.

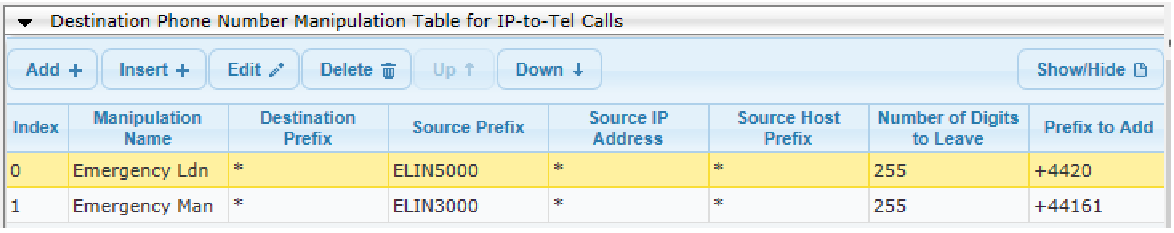
The above example scenario is configured as follows:

1. Enable location-based emergency routing, by loading an ini file to the device with the following parameter setting:

```
E911Gateway = 2
```

2. In the Destination Phone Number Manipulation Table for IP-to-Tel Calls (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Dest Number IP->Tel**), add the following two rules for manipulating the destination number of incoming emergency calls, based on ELIN numbers:

Figure 30-9: Destination Number Manipulation Rules for Location-Based Emergency Routing



Destination Phone Number Manipulation Table for IP-to-Tel Calls
Add +
Insert +
Edit
Delete
Up
Down
Show/Hide

Index	Manipulation Name	Destination Prefix	Source Prefix	Source IP Address	Source Host Prefix	Number of Digits to Leave	Prefix to Add
0	Emergency Ldn	*	ELIN5000	*	*	255	+4420
1	Emergency Man	*	ELIN3000	*	*	255	+44161

Index 0 manipulates the destination number for London emergency callers; Index 1 manipulates the destination number for Manchester emergency callers.

30.11.4.4.5 Viewing the ELIN Table

You can view the ELIN table of the ELIN Gateway:

- Using the following CLI command:

```
# show voip gw 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
```

30.12 Multilevel Precedence and Preemption

The device supports Multilevel Precedence and Preemption (MLPP) service. MLPP is a call priority scheme, which does the following:

- Assigns a precedence level (priority level) to specific phone calls or messages.
- Allows higher priority calls (*precedence call*) and messages to preempt lower priority calls and messages (i.e., terminates existing lower priority calls) that are recognized within a user-defined domain (*MLPP domain ID*). The domain specifies the collection of devices and resources that are associated with an MLPP subscriber. When an MLPP subscriber that belongs to a particular domain places a precedence call to another MLPP subscriber that belongs to the same domain, MLPP service can preempt the existing call that the called MLPP subscriber is on for a higher-precedence call. MLPP service availability does not apply across different domains.

MLPP is typically used in the military where, for example, high-ranking personnel can preempt active calls during network stress scenarios such as a national emergency or degraded network situations.

MLPP can be enabled for all calls, using the global parameter, CallPriorityMode, or for specific calls using the Tel Profile parameter, CallPriorityMode.

Notes:

- MLPP is supported on ISDN PRI and BRI interfaces.
- The device provides MLPP interworking between SIP and ISDN (both directions).
- For Trunk Groups configured with call preemption, all must be configured to MLPP [1] or all configured to Emergency [2]. In other words, you cannot set some trunks to [1] and some to [2].
- The global parameter must be set to the same value as that of the Tel Profile parameter; otherwise, the Tel Profile parameter is not applied.
- If you configure call preemption using the global parameter and a new Tel Profile is subsequently added, the TelProfile_CallPriorityMode parameter automatically acquires the same setting as well.

The Resource Priority value in the Resource-Priority SIP header can be any one of those listed in the table below. A default MLPP call Precedence Level (configured by the SIPDefaultCallPriority parameter) is used if the incoming SIP INVITE or PRI Setup message contains an invalid priority or Precedence Level value respectively. For each MLPP call priority level, the Multiple Differentiated Services Code Points (DSCP) can be set to a value from 0 to 63.

Table 30-6: MLPP Call Priority Levels (Precedence) and DSCP Configuration Parameters

MLPP Precedence Level	Precedence Level in Resource-Priority SIP Header	DSCP Configuration Parameter
0 (lowest)	routine	MLPPRoutineRTPDSCP
2	priority	MLPPPriorityRTPDSCP
4	immediate	MLPPImmediateRTPDSCP
6	flash	MLPPFlashRTPDSCP
8	flash-override	MLPPFlashOverRTPDSCP
9 (highest)	flash-override-override	MLPPFlashOverOverRTPDSCP

The device automatically interworks the network identity digits (NI) in the ISDN Q.931 Precedence Information Element (IE) to the network domain subfield of the INVITE's Resource-Priority header, and vice versa. The SIP Resource-Priority header contains two fields, namespace and priority. The namespace is subdivided into two subfields, network-domain and precedence-domain. Below is an example of a Resource-Priority header whose network-domain subfield is "uc", r-priority field is "priority" (2), and precedence-domain subfield is "000000":

```
Resource-Priority: uc-000000.2
```

The MLPP Q.931 Setup message contains the Precedence IE. The NI digits are presented by four nibbles found in octets 5 and 6. The device checks the NI digits according to the translation table of the Department of Defense (DoD) Unified Capabilities (UC) Requirements (UCR 2008, Changes 3) document, as shown below:

Table 30-7: NI Digits in ISDN Precedence

Level IE	Network Domain in SIP Resource-Priority Header
0000	uc
0001	cuc
0002	dod
0003	nato

Notes:

- If the received ISDN message contains NI digits that are not listed in the translation table, the device sets the network-domain to "uc" in the outgoing SIP message.
- If the received SIP message contains a network-domain value that is not listed in the translation table, the device sets the NI digits to "0000" in the outgoing ISDN message.
- If the received ISDN message does not contain a Precedence IE, you can configure the namespace value - dsn (default), dod, drsn, uc, or cuc - in the SIP Resource-Priority header of the outgoing INVITE message. This is done using the MLPPDefaultNamespace parameter. You can also configure up to 32 user-defined namespaces, using the table ini file parameter, ResourcePriorityNetworkDomains. Once defined, you need to set the MLPPDefaultNamespace parameter value to the desired table row index.

By default, the device maps the received Resource-Priority field of the SIP Resource-Priority header to the outgoing ISDN PRI Precedence Level (priority level) field as follows:

- If the network-domain field in the Resource-Priority header is "uc", then the device sets the Precedence Level field in the ISDN PRI Precedence Level IE according to Table 5.3.2.12-4 (Mapping of RPH r-priority Field to PRI Precedence Level Value):

Table 30-8: Mapping of SIP Resource-Priority Header to PRI Precedence Level for MLPP

MLPP Precedence Level	PRI Precedence Level	SIP Resource-Priority Header Field
Routine	4	0
Priority	3	2
Immediate	2	4
Flash	1	6
Flash Override	0	8

- If the network-domain field in the Resource-Priority header is any value other than "uc", then the device sets the Precedence Level field to "0 1 0 0" (i.e., "routine").

This can be modified using the EnableIp2TelInterworkingtable field of the ini file parameter, ResourcePriorityNetworkDomains.

Notes:

- If required, you can exclude the "resource-priority” tag from the SIP Require header in INVITE messages for Tel-to-IP calls when MLPP priority call handling is used. This is configured using the RPRequired parameter.
- For a complete list of the MLPP parameters, see "MLPP and Emergency Call Parameters" on page 896.

30.12.1 MLPP Preemption Events in SIP Reason Header

The device sends the SIP Reason header (as defined in RFC 4411) to indicate the reason and type of a preemption event. The device sends a SIP BYE or CANCEL request, or SIP 480, 486, 488 response (as appropriate) with a Reason header whose Reason-params can includes one of the following preemption cause classes:

- Reason: preemption ;cause=1 ;text=”UA Preemption”
- Reason: preemption ;cause=2 ;text=”Reserved Resources Preempted”
- Reason: preemption ;cause=3 ;text=”Generic Preemption”
- Reason: preemption ;cause=4 ;text=”Non-IP Preemption”

  This Reason cause code indicates that the session preemption has occurred in a non-IP portion of the infrastructure. The device sends this code in the following scenarios:

  - The device performs a network preemption of a busy call (when a high priority call is received), the device sends a SIP BYE or CANCEL request with this Reason cause code.
  - The device performs a preemption of a B-channel for a Tel-to-IP outbound call request from the softswitch for which it has not received an answer response (e.g., Connect), and the following sequence of events occurs:

    a. The device sends a Q.931 DISCONNECT over the ISDN MLPP PRI to the partner switch to preempt the remote end instrument.

    b. The device sends a 488 (Not Acceptable Here) response with this Reason cause code.

- Reason: preemption; cause=5; text=”Network Preemption”

  This Reason cause code indicates preempted events in the network. Within the Defense Switched Network (DSN) network, the following SIP request messages and response codes for specific call scenarios have been identified for signaling this preemption cause:

  - SIP:BYE - If an active call is being preempted by another call
  - CANCEL - If an outgoing call is being preempted by another call
  - 480 (Temporarily Unavailable), 486 (User Busy), 488 (Not Acceptable Here) - Due to incoming calls being preempted by another call.

  The device receives SIP requests with preemption reason cause=5 in the following cases:

  - The softswitch performs a network preemption of an active call - the following sequence of events occurs:

    a. The softswitch sends the device a SIP BYE request with this Reason cause code.

    b. The device initiates the release procedures for the B-channel associated with the call request and maps the preemption cause to PRI Cause = #8 ‘Preemption’. This value indicates that the call is being preempted. For PRI, it also indicates that the B-channel is not reserved for reuse.

    c. The device sends a SIP 200 OK in response to the received BYE, before the SIP end instrument can proceed with the higher precedence call.

  - The softswitch performs a network preemption of an outbound call request for the device that has not received a SIP 2xx response - the following sequence of events occur:

    a. The softswitch sends the device a SIP 488 (Not Acceptable Here) response code with this Reason cause code. The device initiates the release procedures for the B-channel associated with the call request and maps the preemption cause to PRI Cause = #8 ‘Preemption’.

    b. The device deactivates any user signaling (e.g., ringback tone) and when the call is terminated, it sends a SIP ACK message to the softswitch.

30.12.2 Precedence Ring Tone

You can assign a ring tone that is defined in the CPT file to be played when a precedence call is received from the IP side. This is configured by the PrecedenceRingingType parameter. You can configure the duration for which the device plays a preemption tone to the Tel and IP sides if a call is preempted, using the PreemptionToneDuration parameter.

Emergency Telecommunications Services (ETS) calls (e.g., E911) can be configured with a higher priority than any MLPP call (default), using the E911MLPPBehavior parameter.

30.13 Denial of Collect Calls

You can configure the device to reject (disconnect) incoming Tel-to-IP collect calls and to signal this denial to the PSTN. This capability is required, for example, in the Brazilian telecommunication system to deny collect calls. When this feature is enabled upon rejecting the incoming call, the device sends a sequence of signals to the PSTN. This consists of an off-hook, an on-hook after one second, and then an off-hook after two seconds. In other words, this is in effect, a double-answer sequence.

This feature can be enabled for all calls, using the EnableFXODoubleAnswer "global" parameter, or it can be enabled for specific calls, by enabling this feature in a Tel Profile.

Notes:

- This feature is applicable only to FXO interfaces.
- If automatic dialing is also configured for an FXO port enabled with Denial of Collect Calls, the FXO line does not answer the incoming call (ringing) until a SIP 200 OK is received from the remote destination. When a 200 OK is received, a double answer is sent from the FXO line.
- Ensure that the PSTN side is configured to identify this double-answer signal.

30.14 Configuring Multi-Line Extensions and Supplementary Services

The Supplementary Services table lets you configure up to 100 supplementary services for endpoints connected to the device. These endpoints include analog FXS phones and Integrated Services Digital Network (ISDN) Basic Rate Interface (BRI) phones. This table lets you configure multiple phone line extension numbers per FXS/BRI port. Thus, this offers support for point-to-multipoint configuration of several phone numbers per FXS/BRI channel.

For each line extension, you can also configure the following:

- User ID and password for registering the endpoint to a third-party softswitch for authentication and/or billing. For viewing registration status, see "Viewing Registration Status" on page 685.
- Caller ID name that is displayed to the dialed destination, if enabled.
- Enable the receipt of caller ID from incoming calls.

The Supplementary Services table can also be used to to route IP-to-Tel calls (including voice and fax) to specific endpoints based on the called line extension number. To enable this functionality, in the Hunt Group Settings, set the 'Channel Select Mode' field to **Select Trunk by Supplementary Services Table** for the Trunk Group to which the FXS/BRI port belongs (see "Configuring Hunt Group Settings" on page 367).

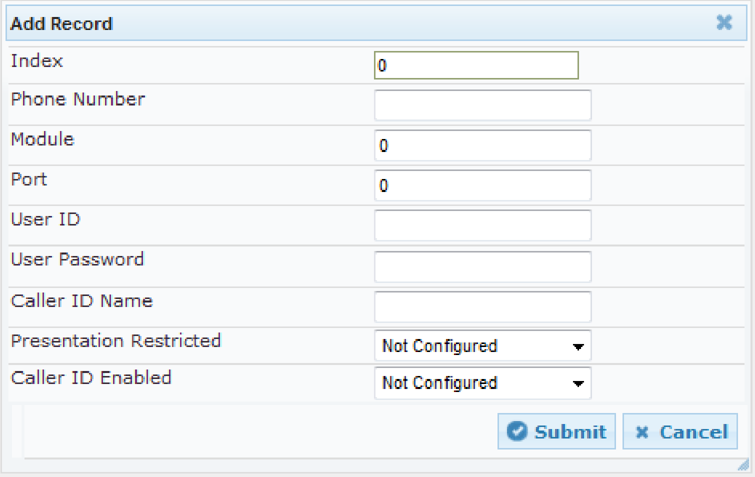
> **Note:** To allow the end-user to hear a dial tone when picking up the BRI phone, it is recommended to set the Progress Indicator in the Setup Ack bit (0x10000=65536). Therefore, the recommended value is 0x10000 + 0 x1000 = 65536 + 4096 = 69632 (i.e., set the ISDNInCallsBehavior parameter to 69632).

The following procedure describes how to configure the Supplementary Services table in the Web interface. You can also configure this table using the table ini file parameter, ISDNSuppServ or CLI command, configure voip > gw digitalgw isdn-supp-serv.

➢ **To configure endpoint supplementary services:**

1. Open the Supplementary Services Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supp Services Table**).
2. Click **Add**; the following dialog box appears:

Figure 30-10: Supplementary Services Table - Add Record



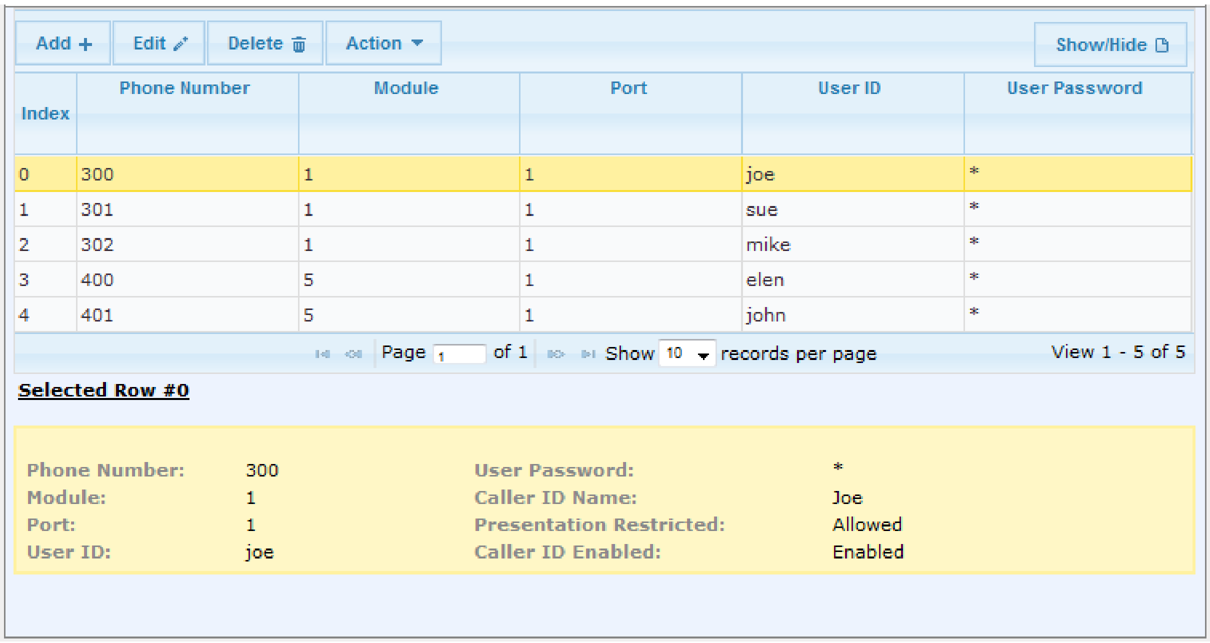
Add Record

Index	0
Phone Number	
Module	0
Port	0
User ID	
User Password	
Caller ID Name	
Presentation Restricted	Not Configured
Caller ID Enabled	Not Configured

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

The figure below displays an example of multiple-line extensions configured in the Supplementary Services table:

Figure 30-11: Supplementary Services Table Page



Index	Phone Number	Module	Port	User ID	User Password
0	300	1	1	joe	*
1	301	1	1	sue	*
2	302	1	1	mike	*
3	400	5	1	elen	*
4	401	5	1	john	*

Selected Row #0

Phone Number:	300	User Password:	*
Module:	1	Caller ID Name:	Joe
Port:	1	Presentation Restricted:	Allowed
User ID:	joe	Caller ID Enabled:	Enabled

You can register and un-register an endpoint configured in the table. The registration method is according to the 'Registration Mode' parameter located in the Trunk Group Settings page (see "Configuring Hunt Group Settings" on page 367).

➢ To register or un-register an endpoint:

1. Select the required table row in which the endpoint is configured.
2. From the 'Action' drop-down list, select **Register**. To unregister the endpoint, select **Un-Register**.

Table 30-9: Supplementary Services Table Parameter Description

Parameter	Description
Index **[ISDNSuppServ_Index]**	Defines an index number for the new table record.
Phone Number CLI: phone-number **[ISDNSuppServ_PhoneNumber]**	Defines a telephone extension number for the endpoint.
Module CLI: module **[ISDNSuppServ_Module]**	Defines the device's module number to which the endpoint is connected.
Port CLI: port **[ISDNSuppServ_Port]**	Defines the port number on the module to which the endpoint is connected.
User ID CLI: user-id **[ISDNSuppServ_UserId]**	Defines the User ID for registering the endpoint to a third-party softswitch for authentication and/or billing.

Parameter	Description
User Password CLI: user-password **[ISDNSuppServ_UserPassword]**	Defines the user password for registering the endpoint to a third-party softswitch for authentication and/or billing. **Note:** For security, the password is displayed as an asterisk (*).
Caller ID Number CLI: caller-id-number **[ISDNSuppServ_CallerID]**	Defines the caller ID name of the endpoint (sent to the IP side). The valid value is a string of up to 18 characters.
Presentation Restricted CLI: presentation-restricted **[ISDNSuppServ_IsPresentationRestricted]**	Determines whether the endpoint sends its Caller ID information to the IP when a call is made. ▪ **[0]** Allowed = The device sends the string defined in the 'Caller ID' field when this endpoint makes a Tel-to-IP call. ▪ **[1]** Restricted = The string defined in the 'Caller ID' field is not sent.
Caller ID Enabled CLI: caller-id-enable **[ISDNSuppServ_IsCallerIDEnabled]**	Enables the receipt of Caller ID. ▪ **[0]** Disabled = The device does not send Caller ID information to the endpoint. ▪ **[1]** Enabled = The device sends Caller ID information to the endpoint.

30.15 Detecting Collect Calls

The device detects collect calls (reverse charge calls) using any of the following information elements (IE) in the received Q.931 ISDN Setup message for Tel-to-IP calls:

- Reverse Charging Indication IE
- Facility IE

When the device detects a collect call, it adds a proprietary header (*X-Siemens-Call-Type: collect call*) to the outgoing SIP INVITE message.

This support does not require any configuration and is applicable to the Euro ISDN protocol variant.

30.16 Advice of Charge Services for Euro ISDN

Advice of charge (AOC) is a pre-billing function that tasks the rating engine with calculating the cost of using a service and relaying that information back to the customer (caller). This allows users to obtain call charging information periodically - during the call (AOC-D) and at the end of the call (AOC-E).

The AOC messages are sent in the EURO ISDN Facility Information Element (IE) message. The device interworks these ISDN messages with SIP by converting the AOC messages into SIP INFO (during call) and BYE messages (end of call), using AudioCodes proprietary SIP AOC header, and vice versa. The device supports both currency (monetary units) and pulse (non-monetary units) AOC messages.

This feature can typically be implemented in the hotel industry, where external calls made by guests can be billed accurately. In such a setup, the device is connected on one side to a PBX through an ISDN line (Euro ISDN), and on the other side to a SIP trunk provided by an ITSP. When a call is made by a guest, the device first sends an AOC-D Facility message to the PBX indicating the connection charge unit, and then sends subsequent AOC-D messages every user-defined interval to indicate the charge unit during the call.

When the call ends, the device sends an AOC-E Facility message to the PBX indicating the total number of charged units.

The device supports various methods for AOC:

- Tel-to-IP direction: The device converts the AOC messages received in the EURO ISDN Facility IE messages into SIP INFO and BYE messages using AudioCodes proprietary SIP AOC header.
- IP-to-Tel direction provides optional methods:
  - The device generates the metering tones according to user-defined pulses and intervals, configured in the Charge Code table (see "Configuring Charge Codes" on page 470). These include:
    - 'Pulses On Answer' - number of charging units in the first generated AOC-D Facility message.
    - 'Pulse Interval' - time between every sent AOC-D Facility message.
    - 'End Time' - time at which the charge code ends.
  - TELES proprietary method. For more information, see the PayPhoneMeteringMode parameter in Metering Tone Parameters on page 954.
  - Cirpack proprietary methods. For more information, see the PayPhoneMeteringMode parameter.

➢ To configure AOC:

1. Make sure that the PSTN protocol for the trunk line is Euro ISDN and set to network side.
2. Make sure that the date and time of the device is correct. For accuracy, it is recommended to use an NTP server to obtain the date and time.
3. For AOC in the Tel-to-IP direction:
   a. Enable the AOC service for sending AOC to IP: on the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**), set 'AoC Support' to **Enable**.
4. For AOC in the IP-to-Tel direction:
   a. Configure the AOC method: on the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**), configure the 'Generate Metering Tones' parameter (PayPhoneMeteringMode).
   b. If the AOC feature uses the Charge Codes table, configure Charge Codes (see Configuring Charge Codes), and then assign the Charge Code index to the relevant Tel-to-IP routing rule in the Outbound IP Routing table (see "Configuring Outbound IP Routing" on page 397).

Note: This feature is applicable to Euro ISDN (PRI and BRI).

30.17 Configuring Charge Codes

The Charge Codes table lets you configure metering tones that the device generates to the Tel side on its FXS interfaces, and for Advice of Charge (AOC) services for Euro ISDN trunks (see Advice of Charge Services for Euro ISDN on page 468). To enable the generation of metering tones, see Configuring Metering Tones on page 474.

You can configure up to 25 different Charge Codes, where each table row represents a Charge Code. Each Charge Code can include up to four different time periods in a day (24 hours). The device selects the time period by comparing the device's current time to the end time of each time period of the selected Charge Code. The device generates the number of pulses on answer once the call is connected, and from that point on, it generates a pulse for each pulse interval. If a call starts at a certain time period and crosses to the next, the information of the next time period is used.

To assign a Charge Code to an outgoing Tel-to-IP call, use the Outbound IP Routing table.

Note: The Charge Codes table is applicable only to FXS and Euro ISDN PRI / BRI interfaces.

The following procedure describes how to configure Charge Codes in the Web interface. You can also configure Charge Codes using the table ini file parameter, ChargeCode or CLI command, configure voip > gw analoggw charge-code.

➢ **To configure a Charge Code:**

1. Open the Charge Codes Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Charge Codes**).

Figure 30-12: Charge Codes Table Page

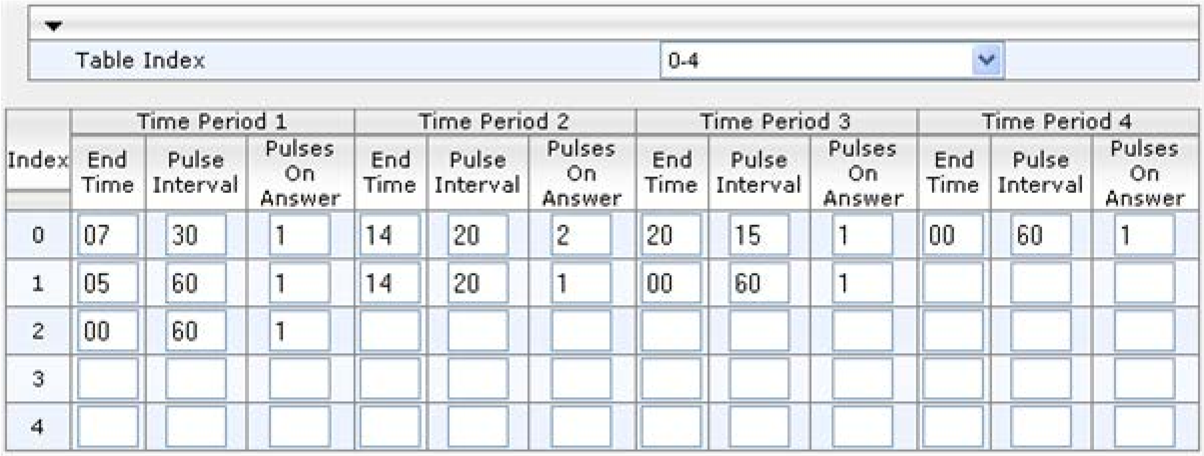


Table Index: 0-4

Index	Time Period 1			Time Period 2			Time Period 3			Time Period 4		
	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer	End Time	Pulse Interval	Pulses On Answer
0	07	30	1	14	20	2	20	15	1	00	60	1
1	05	60	1	14	20	1	00	60	1			
2	00	60	1									
3												
4												

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

Table 30-10: Charge Codes Table Parameter Descriptions

Parameter	Description
End Time CLI: end-time-<1-4> **[ChargeCode_EndTime<1-4>]**	Defines the end of the time period in a 24 hour format, *hh*. For example, "04" denotes 4 A.M. **Notes:** ▪ The first time period always starts at midnight (00).

Parameter	Description
	▪ It is mandatory that the last time period of each rule end at midnight (00). This prevents undefined time frames in a day.
Pulse Interval CLI: pulse-interval-<1-4> **[ChargeCode_PulseInterval<1-4>]**	Defines the time interval between pulses (in tenths of a second).
Pulses On Answer CLI: pulses-on-answer-<1-4> **[ChargeCode_PulsesOnAnswer<1-4>]**	Defines the number of pulses sent on answer.

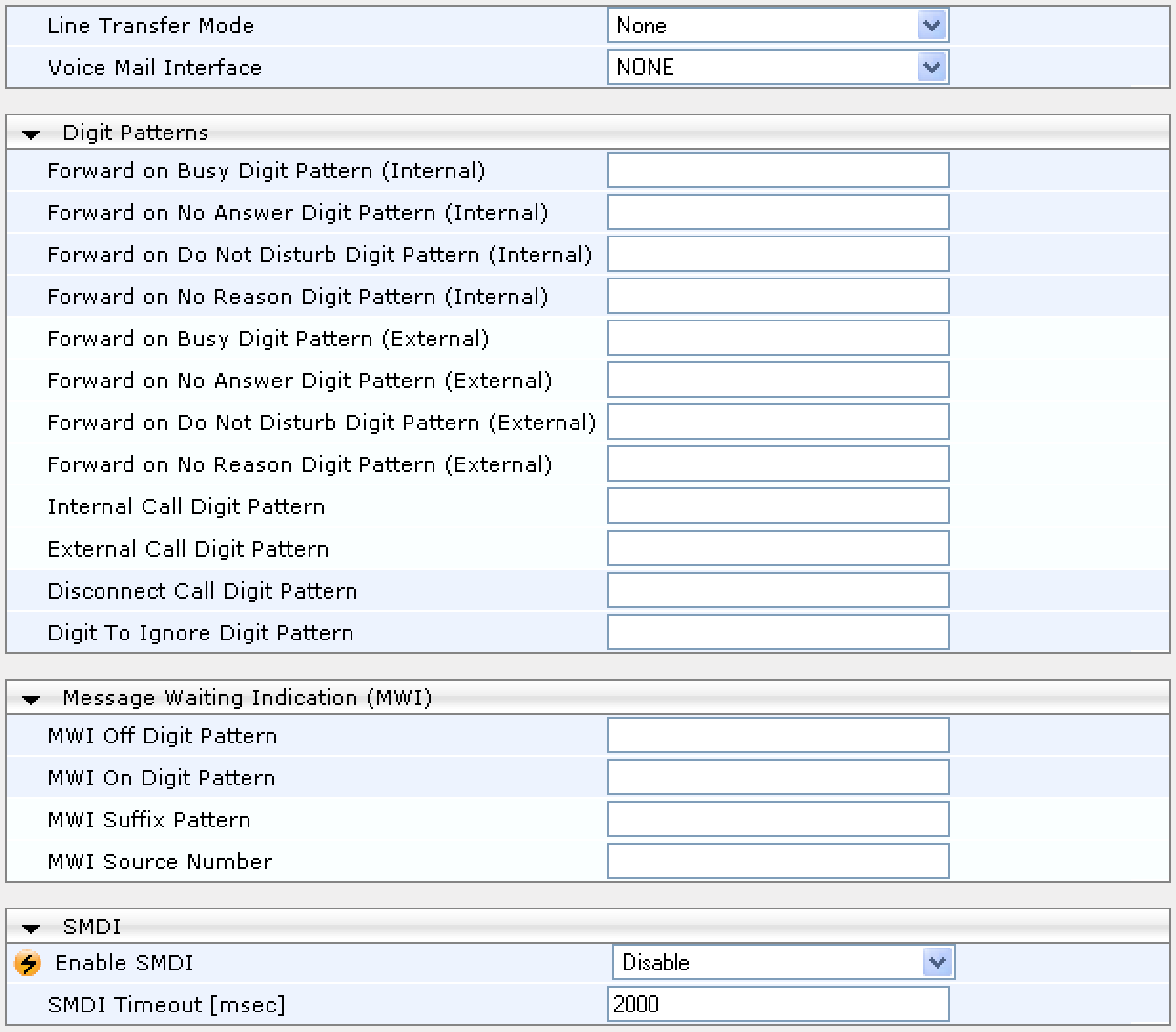
30.18 Configuring Voice Mail

The Voice Mail Settings page allows you to configure the voice mail parameters. For a description of these parameters, see "Configuration Parameters Reference" on page 763.

> **Notes:**
>
> - The Voice Mail Settings page is available only for FXO and CAS interfaces.
> - For more information on configuring voice mail, refer to the *CPE Configuration Guide for Voice Mail User's Manual*.

➢ **To configure the Voice Mail parameters:**

1. Open the Voice Mail Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Advanced Applications** > **Voice Mail Settings**).



Line Transfer Mode	None
Voice Mail Interface	NONE

▼ Digit Patterns

Forward on Busy Digit Pattern (Internal)	
Forward on No Answer Digit Pattern (Internal)	
Forward on Do Not Disturb Digit Pattern (Internal)	
Forward on No Reason Digit Pattern (Internal)	
Forward on Busy Digit Pattern (External)	
Forward on No Answer Digit Pattern (External)	
Forward on Do Not Disturb Digit Pattern (External)	
Forward on No Reason Digit Pattern (External)	
Internal Call Digit Pattern	
External Call Digit Pattern	
Disconnect Call Digit Pattern	
Digit To Ignore Digit Pattern	

▼ Message Waiting Indication (MWI)

MWI Off Digit Pattern	
MWI On Digit Pattern	
MWI Suffix Pattern	
MWI Source Number	

▼ SMDI

Enable SMDI	Disable
SMDI Timeout [msec]	2000

2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

31 Analog Gateway

This section describes configuration of analog settings.

31.1 Configuring Keypad Features

The Keypad Features page enables you to activate and deactivate the following features directly from the connected telephone's keypad:

- Call Forward
- Caller ID Restriction
- Hotline for automatic dialing
- Call Transfer
- Call Waiting
- Rejection of Anonymous Calls

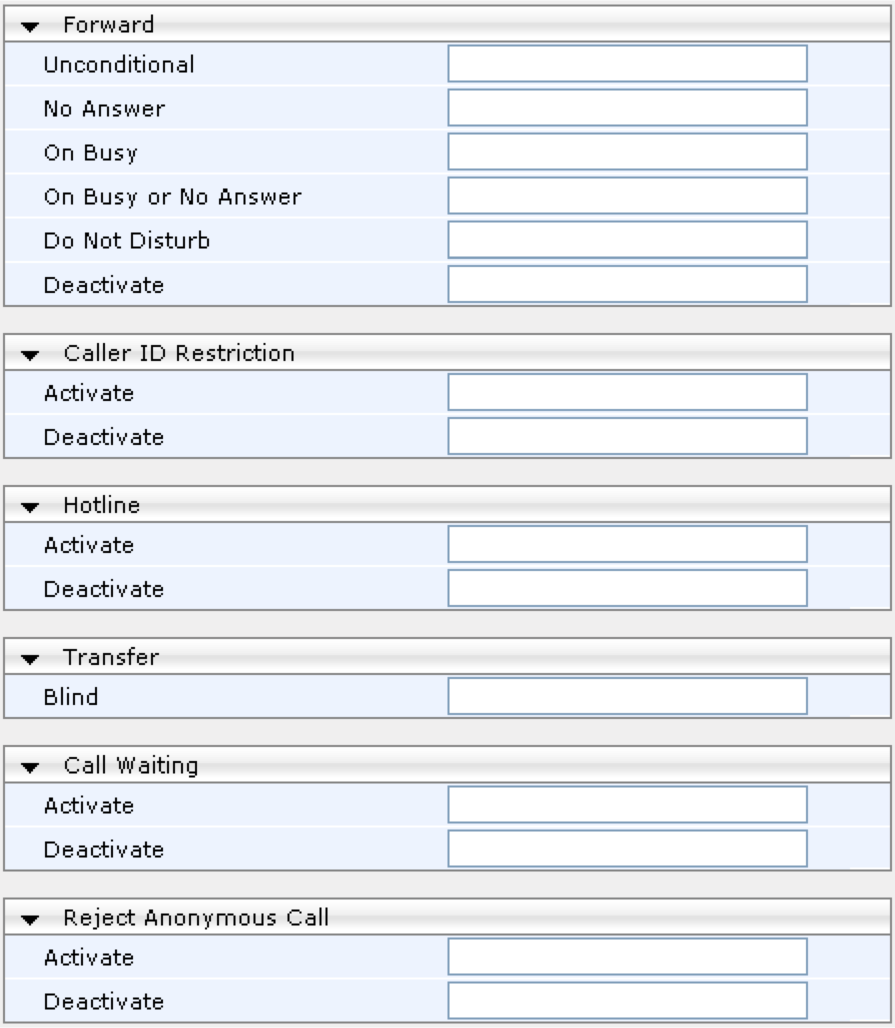
Notes:

- The Keypad Features page is available only for FXS interfaces.
- The method used by the device to collect dialed numbers is identical to the method used during a regular call (i.e., max digits, interdigit timeout, digit map, etc.).
- The activation of each feature remains in effect until it is deactivated (i.e., not deactivated after a call).
- For a description of the keypad parameters, see "Telephone Keypad Sequence Parameters" on page 956.

➢ To configure the keypad features

1. Open the Keypad Features page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Keypad Features**).

Figure 31-1: Keypad Features Page



Forward	
Unconditional	
No Answer	
On Busy	
On Busy or No Answer	
Do Not Disturb	
Deactivate	

Caller ID Restriction	
Activate	
Deactivate	

Hotline	
Activate	
Deactivate	

Transfer	
Blind	

Call Waiting	
Activate	
Deactivate	

Reject Anonymous Call	
Activate	
Deactivate	

2. Configure the keypad features as required.
3. Click **Submit**.

31.2 Configuring Metering Tones

The FXS interfaces can generate 12/16 KHz metering pulses toward the Tel side (e.g., for connection to a pay phone or private meter). Tariff pulse rate is determined according to the device's Charge Codes table. This capability enables users to define different tariffs according to the source/destination numbers and the time-of-day. The tariff rate includes the time interval between the generated pulses and the number of pulses generated on answer.

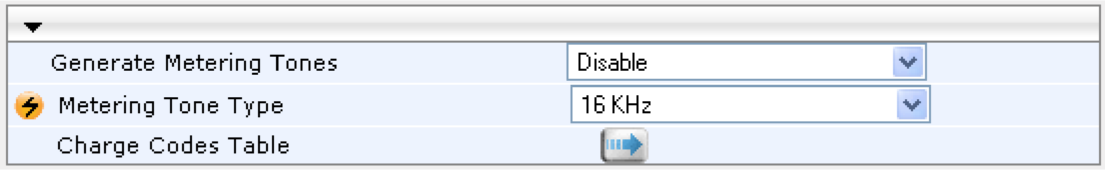
Notes:

- The Metering Tones page is available only for FXS interfaces.
- Charge Code rules can be assigned to routing rules in the Outbound IP Routing table (see "Configuring Outbound IP Routing Table" on page 397). When a new call is established, the Outbound IP Routing table is searched for the destination IP address. Once a route is located, the Charge Code (configured for that route) is used to associate the route with an entry in the Charge Codes table.

➢ To configure Metering tones:

1. Open the Metering Tones page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Metering Tones**).

Figure 31-2: Metering Tones Page



Generate Metering Tones	Disable
Metering Tone Type	16 KHz
Charge Codes Table	

2. Configure the Metering tones parameters as required. For a description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 763.
3. Click **Submit**.
4. To save the changes to the flash memory, see "Saving Configuration" on page 592.

If you set the 'Generate Metering Tones' parameter to **Internal Table**, access the Charge Codes Table page by clicking the **Charge Codes Table** button. For more information on configuring the Charge Codes table, see "Configuring Charge Codes" on page 470.

31.3 Configuring FXO Settings

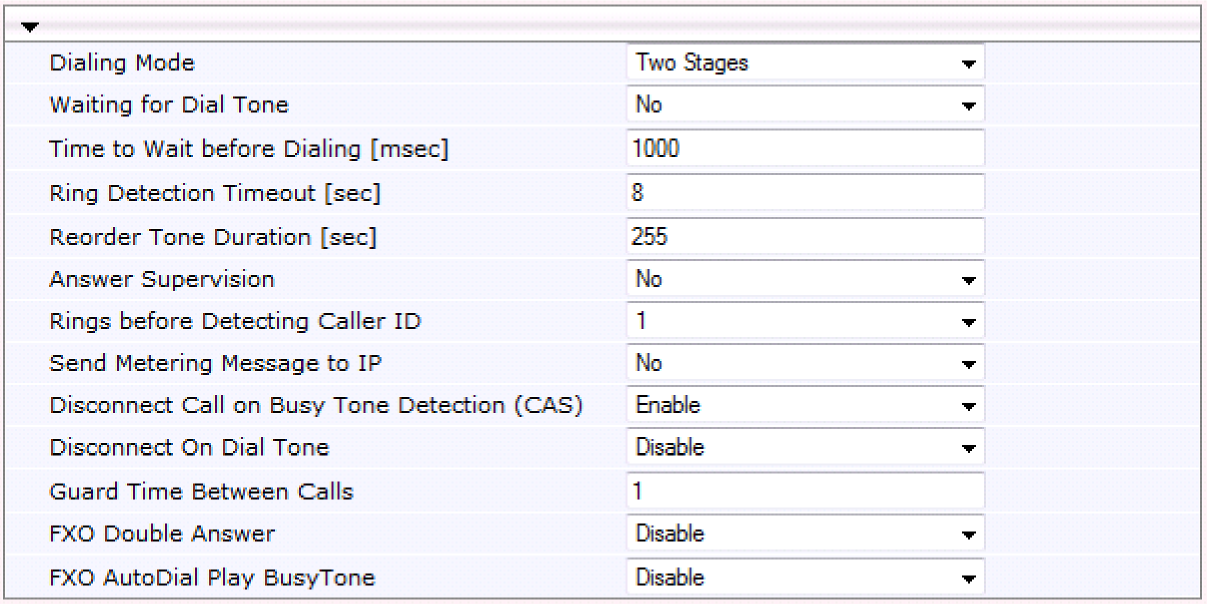
The FXO Settings page allows you to configure the device's specific FXO parameters. For a description of these parameters, see "Configuration Parameters Reference" on page 763.

Note: The FXO Settings page is available only for FXO interfaces.

➢ **To configure the FXO parameters:**

1. Open the FXO Settings page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **FXO Settings**).

Figure 31-3: FXO Settings Page



Dialing Mode	Two Stages
Waiting for Dial Tone	No
Time to Wait before Dialing [msec]	1000
Ring Detection Timeout [sec]	8
Reorder Tone Duration [sec]	255
Answer Supervision	No
Rings before Detecting Caller ID	1
Send Metering Message to IP	No
Disconnect Call on Busy Tone Detection (CAS)	Enable
Disconnect On Dial Tone	Disable
Guard Time Between Calls	1
FXO Double Answer	Disable
FXO AutoDial Play BusyTone	Disable

2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

31.4 Configuring Authentication

The Authentication table lets you configure an authentication username and password per device FXS and FXO port.

Notes:

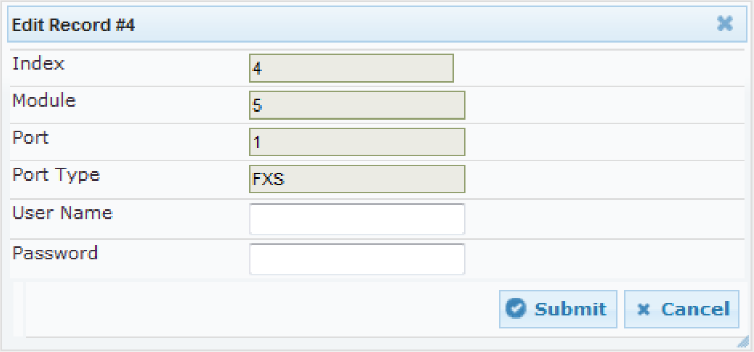
- For configuring whether authentication is done per port or for the entire device, use the parameter AuthenticationMode.
- If authentication is configured for the entire device, the configuration in this table is ignored.
- If the user name or password is not configured in this table, the port's phone number (configured in the Trunk Group table) and global password (configured by the global parameter, Password) are used instead for authentication of the port.
- After you click **Submit**, the password is displayed as an asterisk (*).

The following procedure describes how to configure authentication per port in the Web interface. You can also configure this using the table ini file parameter, Authentication or CLI command, configure voip > gw analoggw authentication.

➢ To configure authentication credentials per port:

1. Set the parameter 'Registration Mode' (AuthenticationMode) to **Per Endpoint.** This can be configured in any of the following pages:
   - Proxy & Registration page (see "Configuring Proxy and Registration Parameters" on page 296).
   - Trunk Group Settings page (see "Configuring Hunt Group Settings" on page 367), where registration method is configured per Trunk Group.
2. Open the Authentication page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Authentication**).
3. Click **Add**; the following dialog box appears:

Figure 31-4: Authentication Page - Edit Record



Edit Record #4

Index	4
Module	5
Port	1
Port Type	FXS
User Name	
Password	

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

Table 31-1: Authentication Table Parameter Descriptions

Parameter	Description
Index	(Read-only) Displays the index number of the table record.

Parameter	Description
[Authentication_Index]	
Port CLI: port **[Authentication_Port]**	(Read-only) Displays the port number.
Module CLI: port-type **[Authentication_Module]**	(Read-only) Displays the module number on which the port is located.
Port Type **[Authentication_PortType]**	(Read-only) Displays the port type (FXS or FXO).
User Name CLI: user-name **[Authentication_UserId]**	Defines the user name used for authenticating the port.
Password CLI: password **[Authentication_UserPassword]**	Defines the password used for authenticating the port.

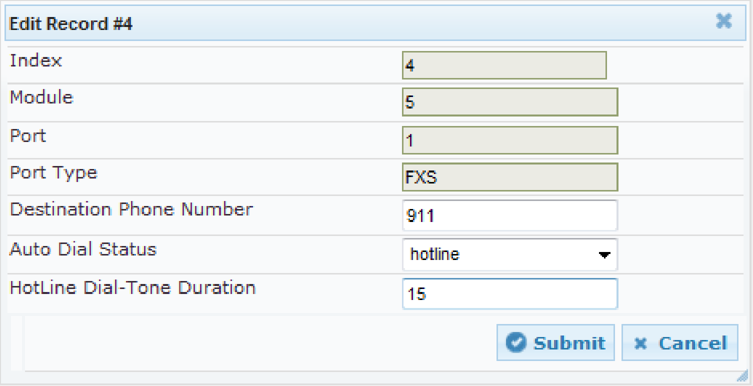
31.5 Configuring Automatic Dialing

The Automatic Dialing table lets you configure telephone numbers that are automatically dialed when FXS or FXO ports go off-hook. The dialing can be done immediately upon off-hook, or after a user-defined interval after off-hook referred to as *Hotline* dialing.

The following procedure describes how to configure automatic dialing upon off-hook in the Web interface. You can also configure this using the table ini file parameter, TargetOfChannel or CLI command, configure voip > gw analoggw automatic-dialing.

➢ **To configure automatic dialing per port:**

1. Open the Automatic Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Automatic Dialing**).
2. Click **Add**; the following dialog box appears:



Edit Record #4

Index	4
Module	5
Port	1
Port Type	FXS
Destination Phone Number	911
Auto Dial Status	hotline
HotLine Dial-Tone Duration	15

The figure above shows a configuration example where hotline-automatic dialing is enabled for an FXS port, whereby if the port is off-hooked for over 15 seconds, the device automatically dials 911.

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

Table 31-2: Automatic Dialing Table Parameter Descriptions

Parameter	Description
Index **[TargetOfChannel_Index]**	(Read-only) Displays the index number on the table netry.
Module **[TargetOfChannel_Module]**	(Read-only) Displays the module number on which the port is located.
Port CLI: port **[TargetOfChannel_Port]**	(Read-only) Displays the port number.
Port Type **[TargetOfChannel_PortType]**	(Read-only) Displays the port type (FXS or FXO).
Destination Phone Number CLI: /dst-number **[TargetOfChannel_Destination]**	Defines the destination telephone number to automatically dial.
Auto Dial Status CLI: auto-dial-status **[TargetOfChannel_Type]**	Enables automatic dialing. ▪ **[0]** Disable = Automatic dialing for the specific port is disabled. ▪ **[1]** Enable = (Default) Automatic dialing is enabled and the phone number configured in the 'Destination Phone Number' field is automatically dialed if the following occurs: ✓ FXS interfaces: The phone is off-hooked ✓ FXO interfaces: A ring signal (from a PBX/PSTN switch) is detected on the FXO line. The device initiates a call to the destination without seizing the FXO line. The line is seized only after the SIP call is answered. ▪ **[2]** Hotline = Automatic dialing is done after an interval configured by the 'Hotline Dial Tone Duration' parameter: ✓ FXS interfaces: When the phone is off-hooked and no digit is dialed within a user-defined time, the configured destination number is automatically dialed. ✓ FXO interfaces: If a ring signal is detected, the device seizes the FXO line, plays a dial tone, and then waits for DTMF digits. If no digits are detected within a user-defined time, the configured destination number is automatically dialed by sending a SIP INVITE message with this number.
Hotline Dial Tone Duration CLI: hotline-dia-ltone-duration **[TargetOfChannel_HotLineToneDuration]**	Defines the duration (in seconds) after which the destination phone number is automatically dialed. This is applicable only if the port has been configured for Hotline (i.e., 'Auto Dial Status' is set to **Hotline**). The valid value is 0 to 60. The default is 16. **Note:** You can configure this Hotline interval for all ports, using the global parameter, HotLineToneDuration.

31.6 Configuring Caller Display Information

The Caller Display Information table lets you configure caller identification strings (Caller ID) per FXS and FXO port. this table also lets you enable the device to send the Caller ID to the IP when a call is made. The called party can use this information for caller identification. The device sends the configured caller ID in the outgoing INVITE message's From header. For information on Caller ID restriction according to destination/source prefixes, see "Configuring Source/Destination Number Manipulation" on page 373.

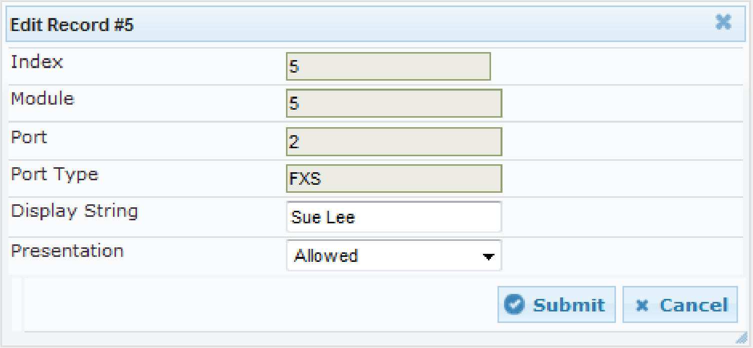
Notes:

- If an FXS port receives 'private' or 'anonymous' strings in the SIP From header, the calling name or number is not sent to the Caller ID display.
- If the device detects Caller ID on an FXO line (EnableCallerID = 1), it uses this Caller ID instead of the Caller ID configured in the Caller Display Information table.

The following procedure describes how to configure caller ID in the Web interface. You can also configure this using the table ini file parameter, CallerDisplayInfo or CLI command, configure voip > gw analoggw caller-display-info.

➢ **To configure Caller display:**

1. Open the Caller Display Information page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Caller Display Information**).
2. Click **Add**; the following dialog box appears:



Edit Record #5

Index: 5
Module: 5
Port: 2
Port Type: FXS
Display String: Sue Lee
Presentation: Allowed

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

Table 31-3: Caller Display Information Table Parameter Descriptions

Parameter	Description
Index **[CallerDisplayInfo_Index]**	(Read-only) Displays the index number of the table record.
Module **[CallerDisplayInfo_Module]**	(Read-only) Displays the module number on which the port is located.
Port **[CallerDisplayInfo_Port]**	(Read-only) Displays the port number.
Port Type **[CallerDisplayInfo_PortType]**	(Read-only) Displays the port type (FXS or FXO).
Display String CLI: display-string **[CallerDisplayInfo_DisplayString]**	Defines the Caller ID string. The valid value is a string of up to 18 characters.

Parameter	Description
	Note: If you set this parameter to "Private" or "Anonymous", Caller ID is restricted and the settings of the 'Presentation' parameter is ignored.
Presentation CLI: presentation **[CallerDisplayInfo_IsCidRestricted]**	Enables the sending of the caller ID string. ▪ **[0]** Allowed = The caller ID string is sent when a Tel-to-IP call is made. ▪ **[1]** Restricted = The caller ID string is not sent. The Caller ID is sent to the remote side using only the SIP P-Asserted-Identity or P-Preferred-Identity headers, according to the AssertedIdMode parameter. **Note:** This parameter is overridden by the 'Presentation' parameter in the Source Number Manipulation table (see "Configuring Source/Destination Number Manipulation" on page 373).

31.7 Configuring Call Forward

The Call Forward table lets you configure call forwarding per FXS or FXO port, for IP-to-Tel calls. This redirects the call, using a SIP 302 response, initially destined to a specific port, to a different port on the device or to an IP destination.

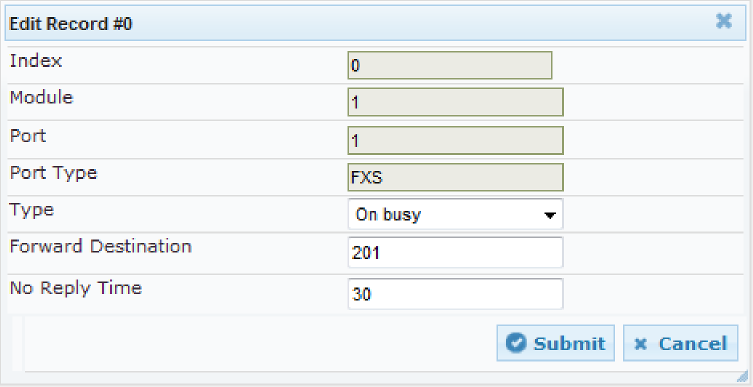
Note: To enable call forwarding, set the 'Enable Call Forward' parameter to **Enable**. This is done in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**).

The following procedure describes how to configure call forwarding per port in the Web interface. You can also configure this using the table ini file parameter, FwdInfo or CLI command, configure voip > gw analoggw call-forward.

➢ To configure call forward per port:

1. Open the Call Forward Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Call Forward**).
2. Click **Add**; the following dialog box appears:

Figure 31-5: Call Forward Table - Edit Record



Edit Record #0

Index	0
Module	1
Port	1
Port Type	FXS
Type	On busy
Forward Destination	201
No Reply Time	30

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

Table 31-4: Call Forward Table Parameter Descriptions

Parameter	Description
Index **[FwdInfo_Index]**	(Read-only) Displays the index number of the table record.
Module **[FwdInfo_Module]**	(Read-only) Displays the module number on which the port is located.
Port **[FwdInfo_Port]**	(Read-only) Displays the port number.
Port Type **[FwdInfo_PortType]**	(Read-only) Displays the port type (FXS or FXO).
Type CLI: type **[FwdInfo_Type]**	Defines the condition upon which the call is forwarded. ▪ **[0]** Deactivate = (Default) Don't forward incoming calls. ▪ **[1]** On Busy = Forward incoming calls when the port is busy. ▪ **[2]** Unconditional = Always forward incoming calls. ▪ **[3]** No Answer = Forward incoming calls that are not answered within the time specified in the 'No Reply Time' field. ▪ **[4]** On Busy or No Answer = Forward incoming calls when the port is busy or when calls are not answered within the time specified in the 'No Reply Time' field. ▪ **[5]** Don't Disturb = Immediately reject incoming calls.
Forward Destination CLI: destination **[FwdInfo_Destination]**	Defines the telephone number or URI (<number>@<IP address>) to where the call is forwarded. **Note:** If this parameter is configured with only a telephone number and a Proxy isn't used, this forwarded-to phone number must be specified in the Outbound IP Routing table (see "Configuring Outbound IP Routing Table" on page 397).
No Reply Time CLI: no-reply-time **[FwdInfo_NoReplyTime]**	If you have set the 'Type' parameter for this port to **No Answer** or **On Busy or No Answer**, then configure the number of seconds the device waits before forwarding the call to the specified phone number.

31.8 Configuring Caller ID Permissions

The Caller ID Permissions table lets you enable per port, Caller ID generation for FXS interfaces and Caller ID detection for FXO interfaces.

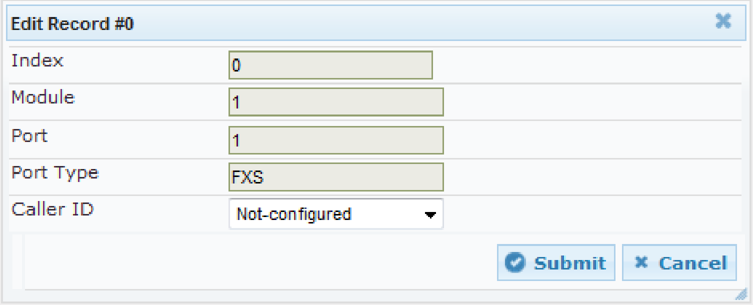
> **Note:** If Caller ID permissions is not configured for a port in this table, its Caller ID generation / detection is determined according to the global parameter, 'Enable Call ID' in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**).

The following procedure describes how to configure caller ID permissions in the Web interface. You can also configure this using the table ini file parameter, EnableCallerID or the CLI command, configure voip > gw analoggw enable-caller-id.

➢ **To configure caller ID permissions per port:**

1. Open the Caller ID Permissions page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Caller ID Permissions**).
2. Click **Add**; the following dialog box appears:

Figure 31-6: Caller ID Permissions Page - Edit Record



Edit Record #0

Index	0
Module	1
Port	1
Port Type	FXS
Caller ID	Not-configured

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

Table 31-5: Caller ID Permissions Table Parameter Descriptions

Parameter	Description
Index **[EnableCallerId_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Module **[EnableCallerId_Module]**	(Read-only) Displays the module number on which the port is located.
Port **[EnableCallerId_Port]**	(Read-only) Displays the port number.
Port Type **[EnableCallerId_PortType]**	(Read-only) Displays the port type (e.g., FXS).
Caller ID CLI: caller-id **[EnableCallerId_IsEnabled]**	Enables Caller ID generation (FXS) or detection (FXO) per port. ▪ **[0]** Disable ▪ **[1]** Enable

31.9 Configuring Call Waiting

The Call Waiting table lets you enable or disable call waiting per FXS port.

Notes:

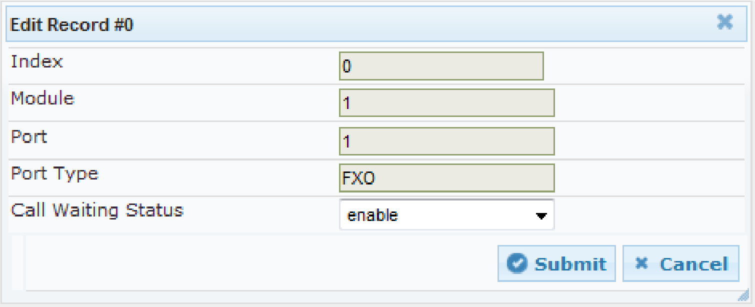
- You can enable or disable call waiting for all the device's ports using the global parameter, 'Enable Call Waiting' in the Supplementary Services page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **Supplementary Services**).
- The CPT file installed on the device must include a 'call waiting Ringback' tone (caller side) and a 'call waiting' tone (called side, FXS interfaces only).
- The EnableHold parameter must be enabled on both the calling and the called sides.
- For additional call waiting configuration, see the following parameters: FirstCallWaitingToneID (in the CPT file), TimeBeforeWaitingIndication, WaitingBeepDuration, TimeBetweenWaitingIndications, and NumberOfWaitingIndications.

The following procedure describes how to configure call waiting per port in the Web interface. You can also configure this using the table ini file parameter, CallWaitingPerPort or CLI command, configure voip > gw analoggw call-waiting.

➢ **To enable call waiting per port:**

1. Open the Call Waiting page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Call Waiting**).
2. Click **Add**; the following dialog box appears:

Figure 31-7: Call Waiting Table - Edit Record



Edit Record #0

Index: 0
Module: 1
Port: 1
Port Type: FXO
Call Waiting Status: enable

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

Table 31-6: Call Waiting Table Parameter Descriptions

Parameter	Description
Index **[CallWaitingPerPort_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Module **[CallWaitingPerPort_Module]**	(Read-only) Displays the module number on which the port is located.
Port **[CallWaitingPerPort_Port]**	(Read-only) Displays the port number.
Port Type	(Read-only) Displays the port type (e.g., FXS).

Parameter	Description
[CallWaitingPerPort_PortType]	
Call Waiting Configuration CLI: enable-call-waiting **[CallWaitingPerPort_IsEnabled]**	Enables call waiting for the port. ▪ **[0]** Disable ▪ **[1]** Enable = Enables call waiting for the port. When the device receives a call on a busy port, it responds with a SIP 182 response (not with a 486 busy). The device plays a call waiting indication signal. When the device detects a hook-flash from the FXS port, the device switches to the waiting call. The device that initiated the waiting call plays a call waiting ringback tone to the calling party after a 182 response is received.

31.10 Rejecting Anonymous Calls

You can configure the device to reject anonymous calls received from the IP and destined for an FXS port. This can be configured using the ini file parameter, RejectAnonymousCallPerPort. If configured, when an FXS interface receives an anonymous call, the device rejects the call and responds with a SIP 433 (Anonymity Disallowed) response. For a description of the parameter see "Caller ID Parameters" on page 880.

31.11 Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number

You can configure a distinctive ringing tone and call waiting tone per calling (source) and/or called (destination) number (or prefix) for IP-to-Tel calls. This feature can be configured per FXS endpoint or for a range of FXS endpoints. Therefore, different tones can be played per FXS endpoint depending on the source and/or destination number of the received call. You can also configure multiple entries with different source and/or destination prefixes and tones for the same FXS port.

Typically, the played ring and/or call waiting tone is indicated in the SIP Alert-info header field of the received INVITE message. If this header is not present in the received INVITE, then this feature is used and the tone played is according to the settings in this table.

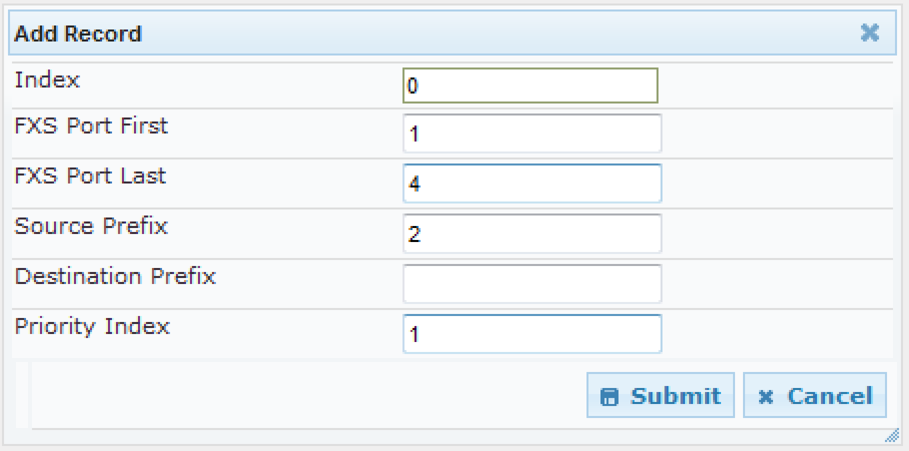
Notes:

- This page is applicable only to FXS interfaces.
- The Tone Index table can also be configured using the table ini file parameter, ToneIndex or CLI command, configure voip > gw analoggw tone-index.

➢ To configure distinctive ringing and call waiting per FXS port:

1. Open the Tone Index Table page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Analog Gateway** > **Tone Index**).
2. Click **Add**; the following dialog box appears:

Figure 31-8: Tone Index Table Page



Add Record

Index	0
FXS Port First	1
FXS Port Last	4
Source Prefix	2
Destination Prefix	
Priority Index	1

The figure above shows a configuration example for using distinctive ringing and call waiting tones of Index #9 ('Priority Index' 1) in the CPT file for FXS endpoints 1 to 4 when a call is received from a source number with prefix 2.

3. Configure the table as required. For a description of the parameters, see the table below.
4. Click **Submit**.

Table 31-7: Tone index Table Parameter Description

Parameter	Description
Index	Defines the table index entry. Up to 50 entries can be defined.
FXS Port First CLI: fxs-port-first **[ToneIndex_FXSPort_First]**	Defines the first port in the FXS port range.
FXS Port Last CLI: fxs-port-last **[ToneIndex_FXSPort_Last]**	Defines the last port in the FXS port range.
Source Prefix CLI: src-prefix **[ToneIndex_SourcePrefix]**	Defines the prefix of the calling number.
Destination Prefix CLI: dst-prefix **[ToneIndex_DestinationPrefix]**	Defines the prefix of the called number.
Priority Index CLI: priority **[ToneIndex_PriorityIndex]**	Defines the index of the distinctive ringing and call waiting tones. The call waiting tone index equals to the Priority Index plus the value of the FirstCallWaitingToneID parameter. For example, if you want to use the call waiting tone in the CPT file at Index #9, you need to enter "1" as the Priority Index value and set the FirstCallWaitingToneID parameter to "8". The summation of these values is 9, i.e., index #9. The default is 0.

31.12 FXS/FXO Coefficient Types

The FXS Coefficient and FXO Coefficient types used by the device can be one of the following:

- US line type of 600 ohm AC impedance and 40 V RMS ringing voltage for REN = 2
- European standard (TBR21)

These Coefficient types are used to increase return loss and trans-hybrid loss performance for two telephony line type interfaces (US or European). This adaptation is performed by modifying the telephony interface characteristics. This means, for example, that changing impedance matching or hybrid balance doesn't require hardware modifications, so that a single device is able to meet requirements for different markets. The digital design of the filters and gain stages also ensures high reliability, no drifts (over temperature or time) and simple variations between different line types.

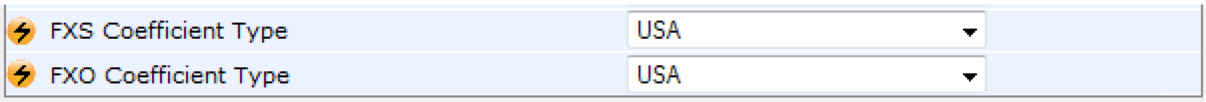
The FXS Coefficient types provide best termination and transmission quality adaptation for two FXS line type interfaces. This parameter affects the following AC and DC interface parameters:

- DC (battery) feed characteristics
- AC impedance matching
- Transmit gain
- Receive gain
- Hybrid balance
- Frequency response in transmit and receive direction
- Hook thresholds
- Ringing generation and detection parameters

➢ **To select the FXO and FXS Coefficient types:**

1. Open the Analog Settings page (**Configuration** tab > **VoIP** menu > **Media** > **Analog Settings**). This page includes the Coefficient type parameters, as shown below:

Figure 31-9: FXS/FXO Coefficient Parameters in Analog Settings Page



FXS Coefficient Type: USA
FXO Coefficient Type: USA

2. From the 'FXS Coefficient Type' drop-down list (FXSCountryCoefficients), select the required FXS Coefficient type.
3. From the 'FXO Coefficient Type' drop-down list (CountryCoefficients), select the required FXO Coefficient type.
4. Click **Submit**.
5. Save your settings to the flash memory ("burn") with a device reset.

31.13 FXO Operating Modes

This section provides a description of the device's FXO operating modes:

- For IP-to-Tel calls (see "FXO Operations for IP-to-Tel Calls" on page 488)
- For Tel-to-IP calls (see "FXO Operations for Tel-to-IP Calls" on page 490)
- Call termination on FXO devices (see "Call Termination on FXO Devices" on page 492)

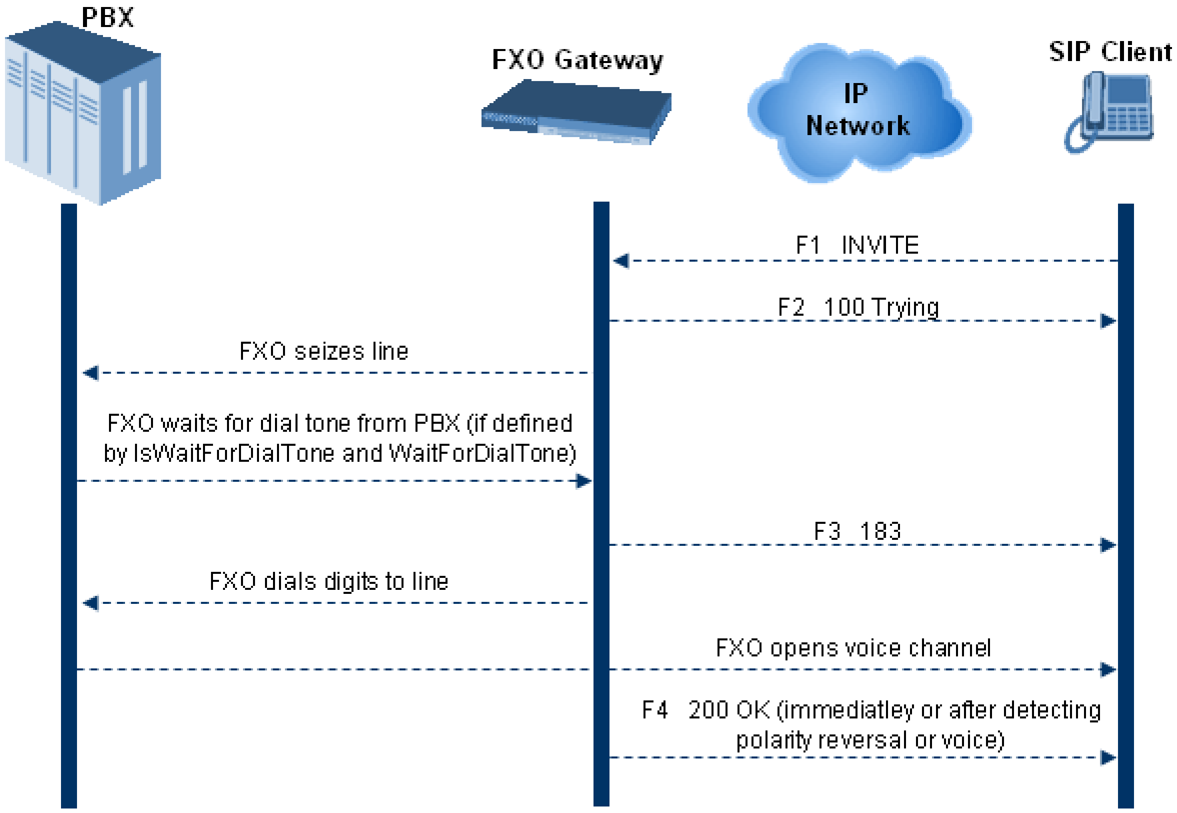
31.13.1 FXO Operations for IP-to-Tel Calls

The FXO device provides the following operating modes for IP-to-Tel calls:

- One-stage dialing (see "One-Stage Dialing" on page 488)
  - Waiting for dial tone (see "Two-Stage Dialing" on page 489)
  - Time to wait before dialing
  - Answer supervision
- Two-stage dialing (see "Two-Stage Dialing" on page 489)
- Dialing time: DID wink (see "DID Wink" on page 489)

31.13.1.1 One-Stage Dialing

One-stage dialing is when the FXO device receives an IP-to-Tel call, off-hooks the PBX line connected to the telephone, and then immediately dials the destination telephone number. In other words, the IP caller doesn't dial the PSTN number upon hearing a dial tone.


PBX
FXO Gateway
IP Network
SIP Client
F1 INVITE
F2 100 Trying
FXO seizes line
FXO waits for dial tone from PBX (if defined by IsWaitForDialTone and WaitForDialTone)
F3 183
FXO dials digits to line
FXO opens voice channel
F4 200 OK (immediatley or after detecting polarity reversal or voice)

One-stage dialing incorporates the following FXO functionality:

- **Waiting for Dial Tone:** Enables the device to dial the digits to the Tel side only after detecting a dial tone from the PBX line. The *ini* file parameter IsWaitForDialTone is used to configure this operation.
- **Time to Wait Before Dialing:** Defines the time (in msec) between seizing the FXO line and starting to dial the digits. The *ini* file parameter WaitForDialTime is used to configure this operation.

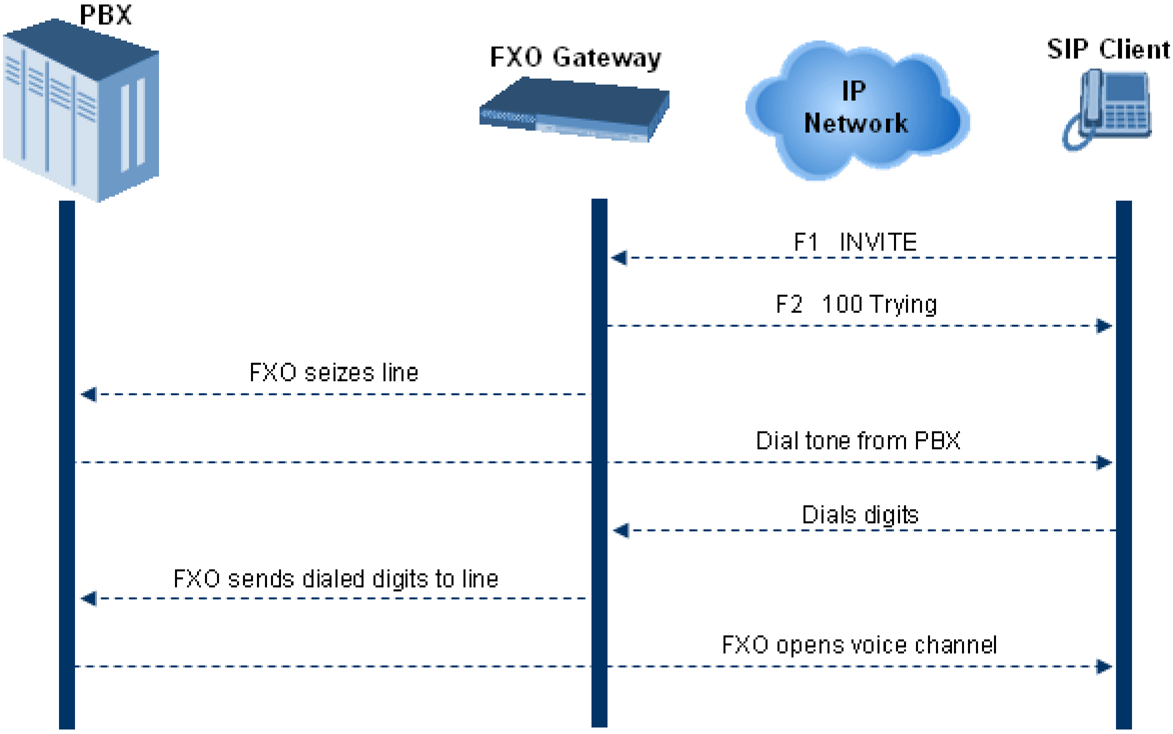
Note: The *ini* file parameter IsWaitForDialTone must be disabled for this mode.

- **Answer Supervision:** The Answer Supervision feature enables the FXO device to determine when a call is connected, by using one of the following methods:
  - **Polarity Reversal:** the device sends a 200 OK in response to an INVITE only when it detects a polarity reversal.
  - **Voice Detection:** the device sends a 200 OK in response to an INVITE only when it detects the start of speech (fax or modem answer tone) from the Tel side. Note that the IPM detectors must be enabled.

31.13.1.2 Two-Stage Dialing

Two-stage dialing is when the IP caller is required to dial twice. The caller initially dials to the FXO device and only after receiving a dial tone from the PBX (via the FXO device), dials the destination telephone number.

Figure 31-10: Call Flow for Two-Stage Dialing



Two-stage dialing implements the Dialing Time feature. Dialing Time allows you to define the time that each digit can be separately dialed. By default, the overall dialing time per digit is 200 msec. The longer the telephone number, the greater the dialing time.

The relevant parameters for configuring Dialing Time include the following:

- DTMFDigitLength (100 msec): time for generating DTMF tones to the PSTN (PBX) side
- DTMFInterDigitInterval (100 msec): time between generated DTMF digits to PSTN (PBX) side

31.13.1.3 DID Wink

The device's FXO ports support Direct Inward Dialing (DID). DID is a service offered by telephone companies that enables callers to dial directly to an extension on a PBX without the assistance of an operator or automated call attendant. This service makes use of DID trunks, which forward only the last three to five digits of a phone number to the PBX. If, for example, a company has a PBX with extensions 555-1000 to 555-1999, and a caller dials 555-1234, the local central office (CO) would forward, for example, only 234 to the PBX. The PBX would then ring extension 234.

DID wink enables the originating end to seize the line by going off-hook. It waits for acknowledgement from the other end before sending digits. This serves as an integrity check that identifies a malfunctioning trunk and allows the network to send a re-order tone to the calling party.

The "start dial" signal is a wink from the PBX to the FXO device. The FXO then sends the last four to five DTMF digits of the called number. The PBX uses these digits to complete the routing directly to an internal station (telephone or equivalent).

- DID Wink can be used for connection to EIA/TIA-464B DID Loop Start lines
- Both FXO (detection) and FXS (generation) are supported

31.13.2 FXO Operations for Tel-to-IP Calls

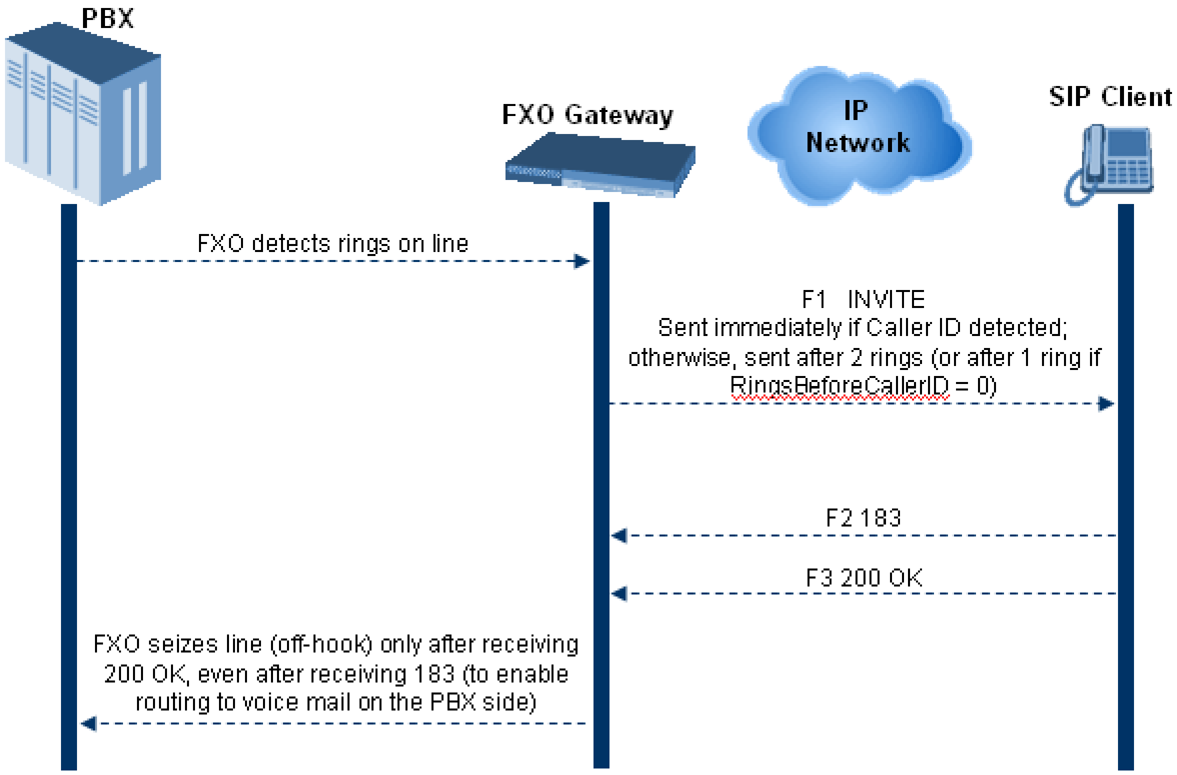
The FXO device provides the following FXO operating modes for Tel-to-IP calls:

- Automatic Dialing (see "Automatic Dialing" on page 490)
- Collecting Digits Mode (see "Collecting Digits Mode" on page 491)
- FXO Supplementary Services (see "FXO Supplementary Services" on page 491)
  - Hold/Transfer Toward the Tel side
  - Hold/Transfer Toward the IP side
  - Blind Transfer to the Tel side

31.13.2.1 Automatic Dialing

Automatic dialing is defined using the Web interface's Automatic Dialing (TargetOfChannel ini file parameter) page, described in see "Configuring Automatic Dialing" on page 478.

The SIP call flow diagram below illustrates Automatic Dialing.

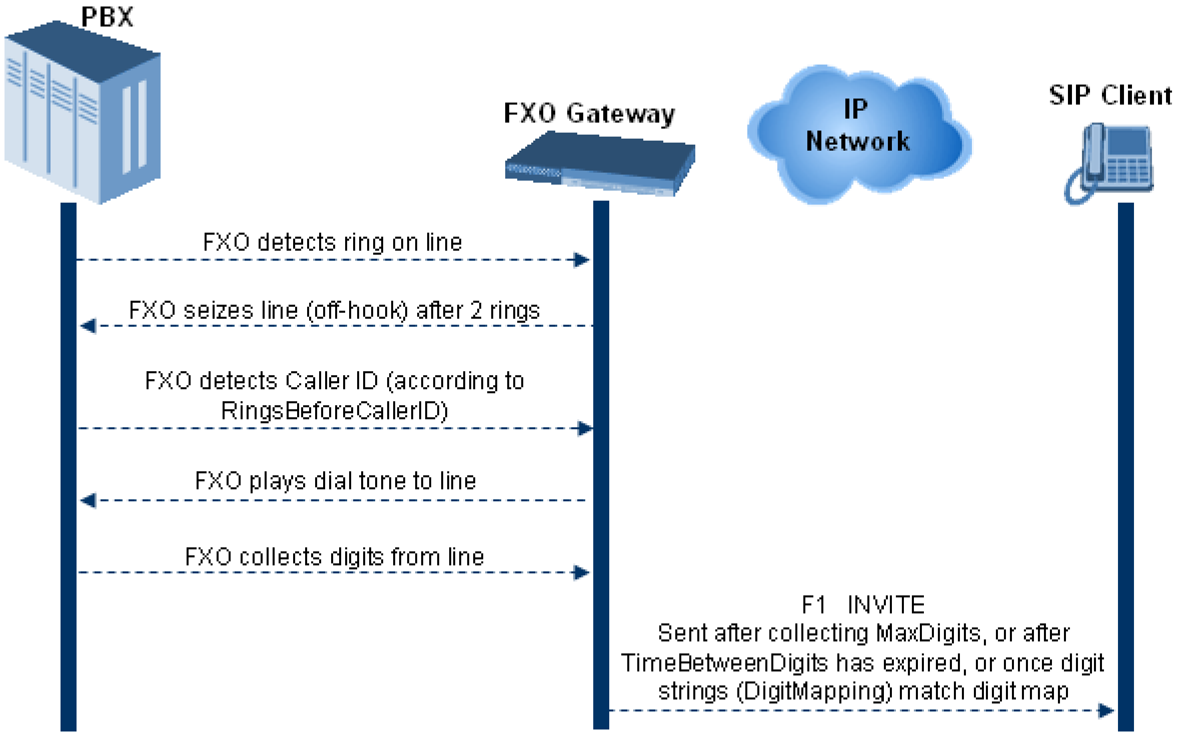

PBX
FXO Gateway
IP Network
SIP Client
FXO detects rings on line
F1 INVITE
Sent immediately if Caller ID detected;
otherwise, sent after 2 rings (or after 1 ring if RingsBeforeCallerID = 0)
F2 183
F3 200 OK
FXO seizes line (off-hook) only after receiving 200 OK, even after receiving 183 (to enable routing to voice mail on the PBX side)

31.13.2.2 Collecting Digits Mode

When automatic dialing is not defined, the device collects the digits.

The SIP call flow diagram below illustrates the Collecting Digits Mode.

Figure 31-11: Call Flow for Collecting Digits Mode


PBX
FXO Gateway
IP Network
SIP Client
FXO detects ring on line
FXO seizes line (off-hook) after 2 rings
FXO detects Caller ID (according to RingsBeforeCallerID)
FXO plays dial tone to line
FXO collects digits from line
F1 INVITE
Sent after collecting MaxDigits, or after TimeBetweenDigits has expired, or once digit strings (DigitMapping) match digit map

31.13.2.3 FXO Supplementary Services

The FXO supplementary services include the following:

- **Hold / Transfer toward the Tel side:** The *ini* file parameter LineTransferMode must be set to 0 (default). If the FXO receives a hook-flash from the IP side (using out-of-band or RFC 2833), the device sends the hook-flash to the Tel side by performing one of the following:
  - Performing a hook flash (i.e., on-hook and off-hook)
  - Sending a hook-flash code (defined by the *ini* file parameter HookFlashCode)

  The PBX may generate a dial tone that is sent to the IP, and the IP side may dial digits of a new destination.
- **Blind Transfer to the Tel side:** A blind transfer is one in which the transferring phone connects the caller to a destination line before ringback begins. The *ini* file parameter LineTransferMode must be set to 1.

  The blind transfer call process is as follows:
  - FXO receives a REFER request from the IP side
  - FXO sends a hook-flash to the PBX, dials the digits (that are received in the Refer-To header), and then drops the line (on-hook). Note that the time between flash to dial is according to the WaitForDialTime parameter.
  - PBX performs the transfer internally
- **Hold / Transfer toward the IP side:** The FXO device doesn't initiate hold / transfer as a response to input from the Tel side. If the FXO receives a REFER request (with or

without replaces), it generates a new INVITE according to the Refer-To header.

31.13.3 Call Termination on FXO Devices

This section describes the device's call termination capabilities for its FXO interfaces:

- Calls terminated by a PBX (see "Call Termination by PBX" on page 492)
- Calls terminated before call establishment (see "Call Termination before Call Establishment" on page 493)
- Ring detection timeout (see "Ring Detection Timeout" on page 493)

31.13.3.1 Calls Termination by PBX

The FXO device supports various methods for identifying when a call has been terminated by the PBX.

The PBX doesn't disconnect calls, but instead signals to the device that the call has been disconnected using one of the following methods:

- **Detection of polarity reversal/current disconnect:** The call is immediately disconnected after polarity reversal or current disconnect is detected on the Tel side (assuming the PBX/CO generates this signal). This is the recommended method.

  Relevant parameters: EnableReversalPolarity, EnableCurrentDisconnect, CurrentDisconnectDuration, CurrentDisconnectDefaultThreshold, and TimeToSampleAnalogLineVoltage.
- **Detection of Reorder, Busy, Dial, and Special Information Tone (SIT) tones:** The call is immediately disconnected after a Reorder, Busy, Dial, or SIT tone is detected on the Tel side (assuming the PBX / CO generates this tone). This method requires the correct tone frequencies and cadence to be defined in the Call Progress Tones file. If these frequencies are unknown, define them in the CPT file. The tone produced by the PBX / CO must be recorded and its frequencies analyzed. This method is slightly less reliable than the previous one.

  Relevant parameters: DisconnectOnBusyTone and DisconnectOnDialTone.
- **Detection of silence:** The call is disconnected after silence is detected on both call directions for a specific (configurable) amount of time. The call is not disconnected immediately. Thus, use this method only as a backup option.

  Relevant parameters: EnableSilenceDisconnect and FarEndDisconnectSilencePeriod.
- **Special DTMF code:** A digit pattern that when received from the Tel side, indicates to the device to disconnect the call.

  Relevant *ini* file parameter: TelDisconnectCode.
- **Interruption of RTP stream:** Relevant parameters: BrokenConnectionEventTimeout and DisconnectOnBrokenConnection.

Note: This method operates correctly only if silence suppression is not used.

- **Protocol-based termination of the call from the IP side**

Note: The implemented disconnect method must be supported by the CO or PBX.

31.13.3.2 Call Termination before Call Establishment

The device supports the following call termination methods before a call is established:

- **Call termination upon receipt of SIP error response (in Automatic Dialing mode):** By default, when the FXO device operates in Automatic Dialing mode, there is no method to inform the PBX if a Tel-to-IP call has failed (SIP error response - 4xx, 5xx or 6xx - is received). The reason is that the FXO device does not seize the line until a SIP 200 OK response is received. Use the FXOAutoDialPlayBusyTone parameter to allow the device to play a busy / reorder tone to the PSTN line if a SIP error response is received. The FXO device seizes the line (off-hook) for the duration defined by the TimeForReorderTone parameter. After playing the tone, the line is released (on-hook).
- **Call termination after caller (PBX) on-hooks phone (Ring Detection Timeout feature):** This method operates in one of the following manners:
    - **Automatic Dialing is enabled:** if the remote IP party doesn't answer the call and the ringing signal (from the PBX) stops for a user-defined time (configured by the parameter FXOBetweenRingTime), the FXO device releases the IP call.
    - **No automatic dialing and Caller ID is enabled:** the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP.

31.13.3.3 Ring Detection Timeout

The operation of Ring Detection Timeout depends on the following:

- **Automatic dialing is disabled and Caller ID is enabled:** if the second ring signal is not received for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device doesn’t initiate a call to the IP.
- **Automatic dialing is enabled:** if the remote party doesn't answer the call and the ringing signal stops for a user-defined time (using the parameter FXOBetweenRingTime), the FXO device releases the IP call.

Ring Detection Timeout supports full ring cycle of ring on and ring off (from ring start to ring start).

31.14 Remote PBX Extension between FXO and FXS Devices

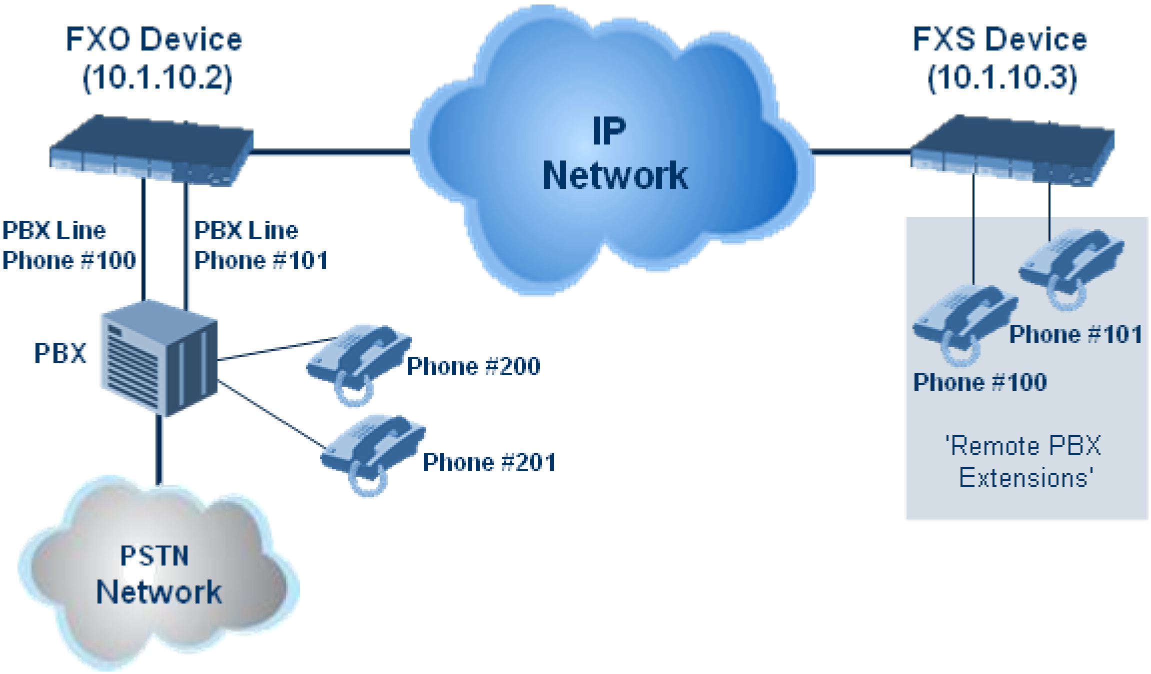
Remote PBX extension offers a company the capability of extending the "power" of its local PBX by allowing remote phones (remote offices) to connect to the company's PBX over the IP network (instead of via PSTN). This is as if the remote office is located in the head office (where the PBX is installed). PBX extensions are connected through FXO ports to the IP network, instead of being connected to individual telephone stations. At the remote office, FXS units connect analog phones to the same IP network. To produce full transparency, each FXO port is mapped to an FXS port (i.e., one-to-one mapping). This allows individual extensions to be extended to remote locations. To call a remote office worker, a PBX user or a PSTN caller simply dials the PBX extension that is mapped to the remote FXS port.

This section provides an example on how to implement a remote telephone extension through the IP network, using FXO andFXS interfaces. In this configuration, the FXO device routes calls received from the PBX to the ‘Remote PBX Extension’ connected to the FXS device. The routing is transparent as if the telephone connected to the FXS device is directly connected to the PBX.

The following is required:

- FXO interfaces with ports connected directly to the PBX lines (shown in the figure below)

- FXS interfaces for the 'remote PBX extension'
- Analog phones (POTS)
- PBX (one or more PBX loop start lines)
- LAN network



31.14.1 Dialing from Remote Extension (Phone at FXS)

The following procedure describes how to dial from the 'remote PBX extension' (i.e., phone connected to the FXS interface).

➢ To make a call from the FXS interface:

1. Off-hook the phone and wait for the dial tone from the PBX. This is as if the phone is connected directly to the PBX. The FXS and FXO interfaces establish a voice path connection from the phone to the PBX immediately after the phone is off-hooked.
2. Dial the destination number (e.g., phone number 201). The DTMF digits are sent over IP directly to the PBX. All the audible tones are generated from the PBX (such as ringback, busy, or fast busy tones). One-to-one mapping occurs between the FXS ports and PBX lines.
3. The call disconnects when the phone connected to the FXS goes on-hook.

31.14.2 Dialing from PBX Line or PSTN

The following procedure describes how to dial from a PBX line (i.e., from a telephone directly connected to the PBX) or from the PSTN to the 'remote PBX extension' (i.e., telephone connected to the FXS interface).

➢ To dial from a telephone directly connected to the PBX or from the PSTN:

- Dial the PBX subscriber number (e.g., phone number 101) in the same way as if the user's phone was connected directly to the PBX. As soon as the PBX rings the FXO device, the ring signal is 'sent' to the phone connected to the FXS device. Once the phone connected to the FXS device is off-hooked, the FXO device seizes the PBX line and the voice path is established between the phone and PBX.

There is one-to-one mapping between PBX lines and FXS device ports. Each PBX line is routed to the same phone (connected to the FXS device). The call disconnects when the phone connected to the FXS device is on-hooked.

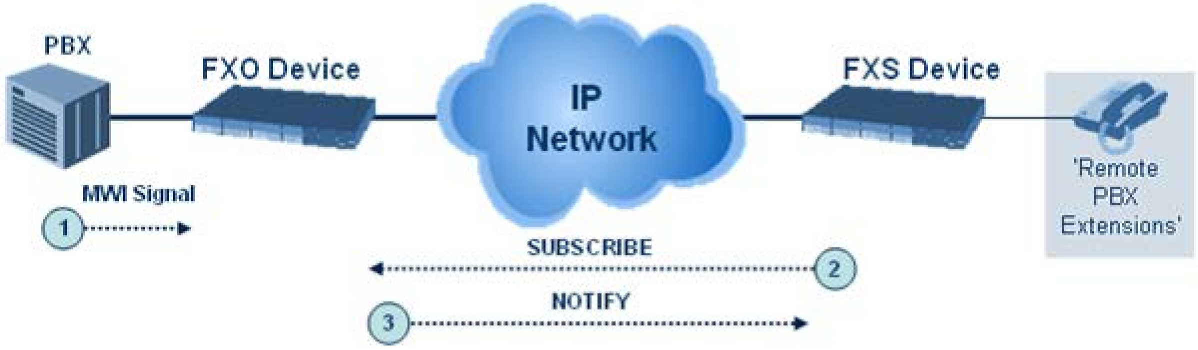
31.14.3 Message Waiting Indication for Remote Extensions

The device supports the relaying of Message Waiting Indications (MWI) for remote extensions (and voice mail applications). Instead of subscribing to an MWI server to receive notifications of pending messages, the FXO device receives subscriptions from the remote FXS device and notifies the appropriate extension when messages (and the number of messages) are pending.

The FXO device detects an MWI message from the Tel (PBX) side using any one of the following methods:

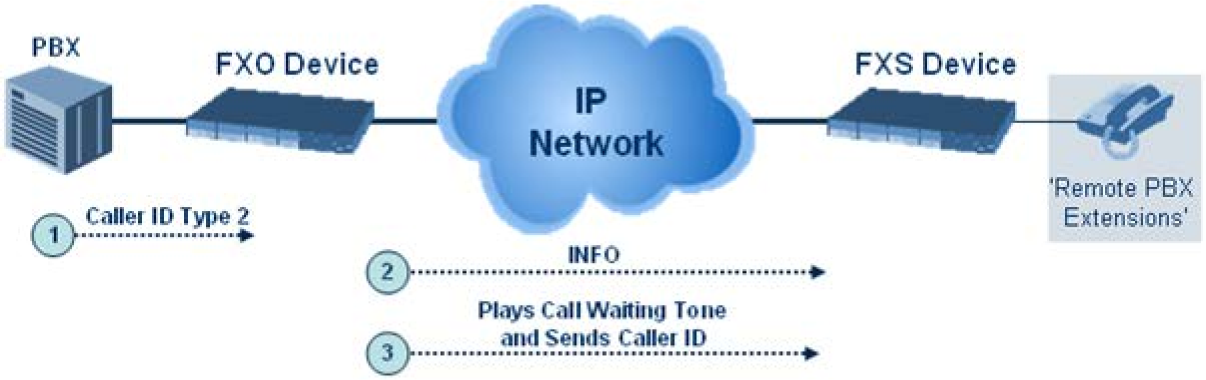
- 100 VDC (sent by the PBX to activate the phone's lamp)
- Stutter dial tone from the PBX
- MWI display signal (according to the parameter CallerIDType)

Upon detection of an MWI message, the FXO device sends a SIP NOTIFY message to the IP side. When receiving this NOTIFY message, the remote FXS device generates an MWI signal toward its Tel side.



31.14.4 Call Waiting for Remote Extensions

When the FXO device detects a Call Waiting indication (FSK data of the Caller Id - CallerIDType2) from the PBX, it sends a proprietary INFO message, which includes the caller identification to the FXS device. Once the FXS device receives this INFO message, it plays a call waiting tone and sends the caller ID to the relevant port for display. The remote extension connected to the FXS device can toggle between calls using the Hook Flash button.



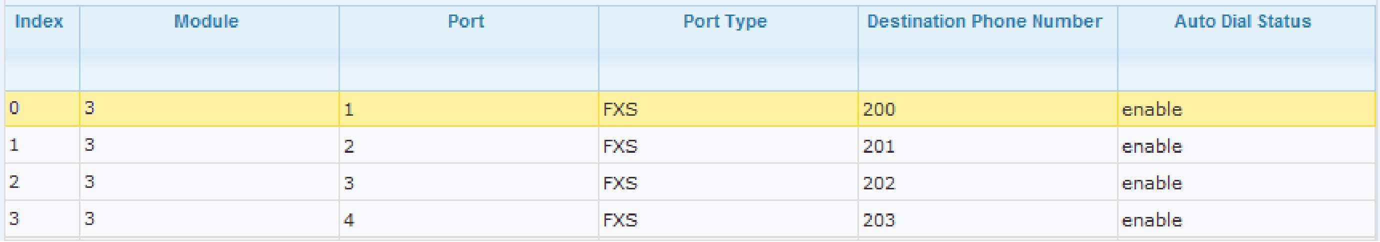
31.14.5 FXS Gateway Configuration

The following procedure describes how to configure the FXS interface (at the 'remote PBX extension').

➤ **To configure the FXS interface:**

1. In the Trunk Group table (see Configuring Trunk Group Table), assign the phone numbers 100 to 104 to the device's endpoints.

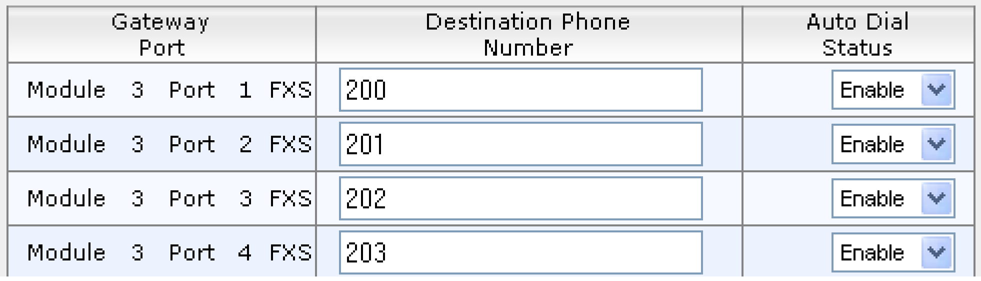
Figure 31-12: Assigning Phone Numbers to FXS Endpoints



Index	Module	Port	Port Type	Destination Phone Number	Auto Dial Status
0	3	1	FXS	200	enable
1	3	2	FXS	201	enable
2	3	3	FXS	202	enable
3	3	4	FXS	203	enable

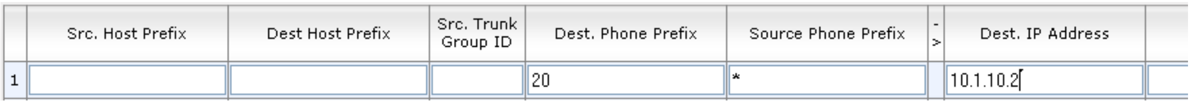
2. In the Automatic Dialing page (see "Configuring Automatic Dialing" on page 478), enter the phone numbers of the FXO device in the 'Destination Phone Number' fields. When a phone connected to Port #1 off-hooks, the FXS device automatically dials the number '200'.

Figure 31-13: Automatic Dialing for FXS Ports



Gateway Port	Destination Phone Number	Auto Dial Status
Module 3 Port 1 FXS	200	Enable
Module 3 Port 2 FXS	201	Enable
Module 3 Port 3 FXS	202	Enable
Module 3 Port 4 FXS	203	Enable

3. In the Outbound IP Routing table (see "Configuring Outbound IP Routing Table" on page 397), enter 20 for the destination phone prefix, and 10.1.10.2 for the IP address of the FXO device.



	Src. Host Prefix	Dest Host Prefix	Src. Trunk Group ID	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address	
1				20	*		10.1.10.2	

Note: For the transfer to function in remote PBX extensions, Hold must be disabled at the FXS device (i.e., Enable Hold = 0) and hook-flash must be transferred from the FXS to the FXO (HookFlashOption = 4).

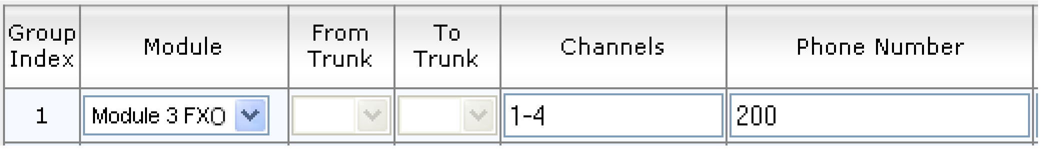
31.14.6 FXO Gateway Configuration

The following procedure describes how to configure the FXO interface (to which the PBX is directly connected).

➢ **To configure the FXO interface:**

1. In the Trunk Group table (see Configuring Trunk Group Table on page 365), assign the phone numbers 200 to 204 to the device's FXO endpoints.

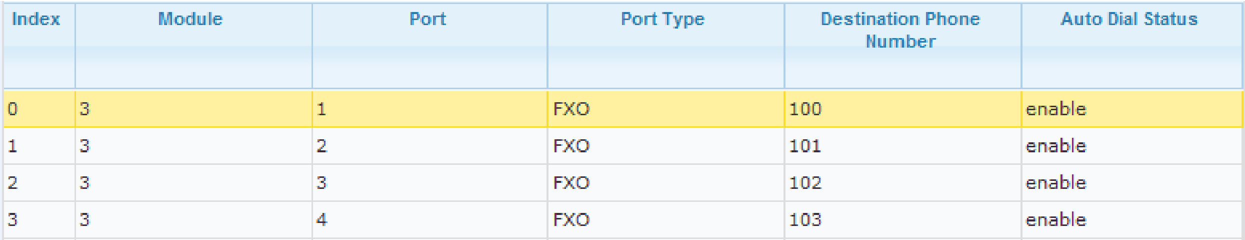
Figure 31-14: Assigning Phone Numbers to FXO Ports



Group Index	Module	From Trunk	To Trunk	Channels	Phone Number
1	Module 3 FXO			1-4	200

2. In the Automatic Dialing page, enter the phone numbers of the FXS device in the 'Destination Phone Number' fields. When a ringing signal is detected at Port #1, the FXO device automatically dials the number '100'.

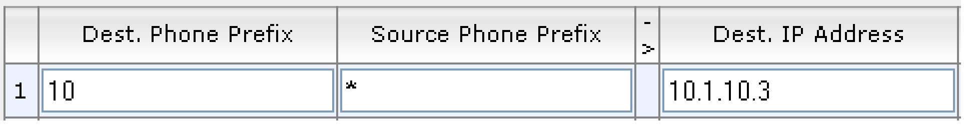
Figure 31-15: FXO Automatic Dialing Configuration



Index	Module	Port	Port Type	Destination Phone Number	Auto Dial Status
0	3	1	FXO	100	enable
1	3	2	FXO	101	enable
2	3	3	FXO	102	enable
3	3	4	FXO	103	enable

3. In the Outbound IP Routing table, enter 10 in the 'Destination Phone Prefix' field, and the IP address of the FXS device (10.1.10.3) in the field 'IP Address'.

Figure 31-16: FXO Tel-to-IP Routing Configuration



	Dest. Phone Prefix	Source Phone Prefix	->	Dest. IP Address
1	10	*		10.1.10.3

4. In the FXO Settings page (see "Configuring FXO Parameters" on page 476), set the parameter 'Dialing Mode' to **Two Stages** (IsTwoStageDial = 1).

This page is intentionally left blank.

Part VI

Session Border Controller Application

32 SBC Overview

This section provides a detailed description of the device's SBC application.

Notes:

- For guidelines on how to deploy your E-SBC device, refer to the E-*SBC Design Guide* document.
- The SBC feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 621.
- 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 1021.

The SBC application supports the following main features:

- NAT traversal: The device supports NAT traversal, allowing, for example, communication with ITSPs with globally unique IP addresses, for LAN-to-WAN VoIP signaling (and bearer), using two independent legs. This also enables communication for "far-end" users located behind a NAT on the WAN. The device supports this by:
  - Continually registering far-end users in its dynamic database.
  - Maintaining remote NAT binding state by frequent registrations, thereby, off-loading far-end registrations from the LAN IP PBX.
  - Using Symmetric RTP (RFC 4961) to overcome bearer NAT traversal.
- VoIP firewall and security for signaling and media:
  - SIP signaling:
    - Deep and stateful inspection of all SIP signaling packets.
    - SIP dialog initiations may be rejected based on values of incoming SIP INVITE message and other Layer-3 characteristics.
    - Packets not belonging to an authorized SIP dialog are discarded.
  - RTP:
    - Opening pinholes (ports) in the device's firewall based on Offer-Answer SDP negotiations.
    - Deep packet inspection of all RTP packets.
    - Late rogue detection - if a SIP session was gracefully terminated and someone tries to "ride on it" with rogue traffic from the already terminated RTP and SIP context, the VoIP Firewall prevents this from occurring.
    - Disconnects call (after user-defined time) if RTP connection is broken.
    - Black/White lists for both Layer-3 firewall and SIP classification.
- Topology hiding: The device intrinsically supports topology hiding, limiting the amount of topology information displayed to external parties. For example, IP addresses of ITSPs' equipment (e.g. proxies, gateways, and application servers) can be hidden from outside parties. The device's topology hiding is provided by implementing back-to-back user agent (B2BUA) leg routing:
  - Strips all incoming SIP Via header fields and creates a new Via value for the outgoing message.
  - Each leg has its own Route/Record Route set.
  - Modifies SIP To, From, and Request-URI host names (must be configured using the Message Manipulations table).
  - Generates a new SIP Call-ID header value (different between legs).

    - Changes the SIP Contact header to the device's own address.
    - Layer-3 topology hiding by modifying source IP address in the SIP IP header.
- SIP normalization: The device supports SIP normalization, whereby the SBC application can overcome interoperability problems between SIP user agents. This is achieved by the following:
    - Manipulation of SIP URI user and host parts.
    - Connection to ITSP SIP trunks on behalf of an IP-PBX - the device can register and utilize user and password to authenticate for the IP-PBX.
- Survivability:
    - Routing calls to alternative routes such as the PSTN.
    - Routing calls between user agents in the local network using a dynamic database (built according to registrations of SIP user agents).
- Routing:
    - IP-to-IP routing translations of SIP, UDP, TCP, TLS (when extensive transcoding is not required).
    - Load balancing and redundancy of SIP servers.
    - Routing according to Request-URI\Specific IP address\Proxy\FQDN.
    - Alternative routing.
    - Routing between different Layer-3 networks (e.g., LAN and WAN).
- Load balancing\redundancy of SIP servers.
- ITSP accounts.
- SIP URI user and host name manipulations.
- Coder transcoding.

32.1 SIP Network Definitions

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

32.2 SIP Dialog Initiation Process

The device's SIP dialog initiation process concerns all incoming SIP dialog initiation requests. This includes SIP methods such as INVITE, SUBSCRIBE, OPTIONS, REFER, INFO, UNSOLICITED NOTIFY, MESSAGE, and REGISTER.

The SIP dialog initiation process consists of the following stages:

1. **Determining source and destination URL:** The SIP protocol has more than one URL in a dialog-establishing request that may represent the source and destination URLs. When handling an incoming request, the device uses specific SIP headers for obtaining the source and destination URLs. Once these URLs are determined, their user and host parts are used as input for the classification process, message manipulation, and call routing.
    - **All SIP requests (e.g., INVITE) except REGISTER dialogs:**
        - ♦ Source URL: The source URL is obtained from the SIP header according to the following logic:
            - ✓ The source URL is obtained from the From header.
            - ✓ If the From header contains the value 'Anonymous', the source URL is obtained from the P-Preferred-Identity header.
            - ✓ If the P-Preferred-Identity header does not exist, the source URL is obtained from the P-Asserted-Identity header.

    - ♦ Destination URL: The destination URL is obtained from the Request-URI.
- **REGISTER dialogs:**
    - ♦ Source URL: The source URL is obtained from the To header.
    - ♦ Destination URL: The destination URL is obtained from the Request-URI.

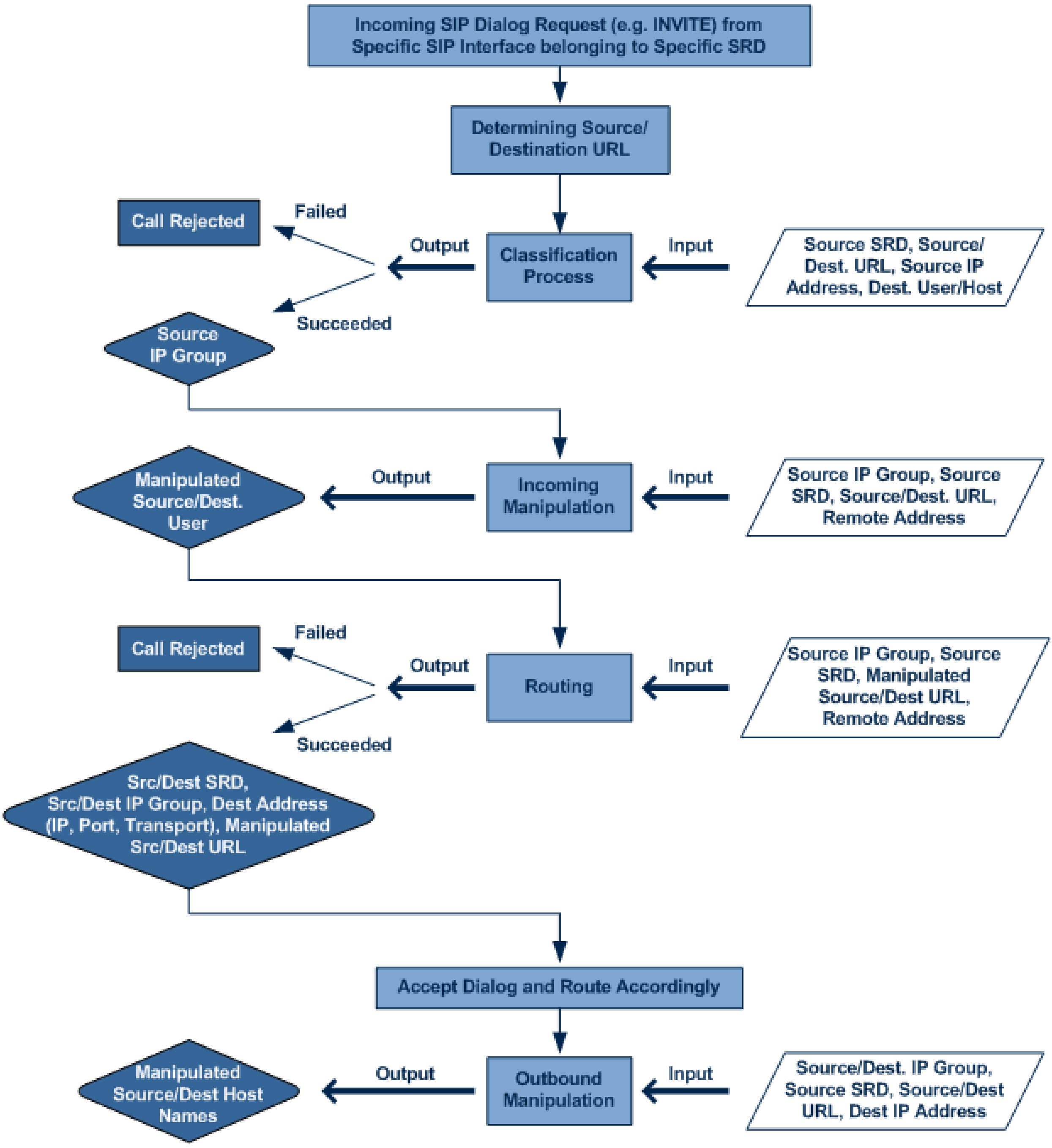
> **Note:** You can determine the SIP header from where the device obtains the source URL in the incoming SIP request. This is done in the IP Group table using the 'Source URI Input' parameter.

2. **Classifying incoming SIP dialog-initiating requests to a source IP Group:** The classification identifies the incoming SIP dialog request as belonging to a specific IP Group (from where the SIP dialog request originated). For more information, see "Configuring Classification Rules" on page 545.
3. **SBC IP-to-IP routing:** The device routes the call to a destination that can be configured to one of the following:
    - Registered user Contact listed in the device's database (only for User-type IP Groups).
    - IP Group - the destination is the address configured for the Proxy Set associated with the IP Group (allows redundancy/load balancing).
    - Specified destination address (can be based on IP address, host name, port, transport type, and/or SRD). Routing to a host name can be resolved using NAPTR/SRV/A-Record.
    - Request-URI of incoming SIP dialog initiating requests.
    - ENUM query.
    - Hunt Group - used for call survivability.
    - IP address (in dotted-decimal notation or FQDN - NAPTR/SRV/A-Record resolutions) according to a specified Dial Plan index listed in the loaded Dial Plan file.
    - LDAP server or LDAP query result.

    For more information, see "Configuring SBC IP-to-IP Routing Rules" on page 552.
4. **Manipulating SIP URI user part (source and destination) of inbound and/or outbound SIP dialog requests:** You can configure rules for manipulating the SIP URI user part (source and destination) on the inbound and/or outbound leg. For more information, see "SBC Manipulations" on page 563.
5. **SIP message manipulations:** You can configure SIP message manipulation rules that can add, remove, and/or modify SIP headers and parameters. For more information, see "Configuring SIP Message Manipulation" on page 299.

The flowchart below illustrates the SBC process:

Figure 32-1: Routing Process


Incoming SIP Dialog Request (e.g. INVITE) from Specific SIP Interface belonging to Specific SRD
Determining Source/ Destination URL
Call Rejected
Failed
Output
Classification Process
Input
Source SRD, Source/ Dest. URL, Source IP Address, Dest. User/Host
Succeeded
Source IP Group
Manipulated Source/Dest. User
Output
Incoming Manipulation
Input
Source IP Group, Source SRD, Source/Dest. URL, Remote Address
Call Rejected
Failed
Output
Routing
Input
Source IP Group, Source SRD, Manipulated Source/Dest URL, Remote Address
Succeeded
Src/Dest SRD, Src/Dest IP Group, Dest Address (IP, Port, Transport), Manipulated Src/Dest URL
Accept Dialog and Route Accordingly
Manipulated Source/Dest Host Names
Output
Outbound Manipulation
Input
Source/Dest. IP Group, Source SRD, Source/Dest URL, Dest IP Address

32.3 User Registration

To allow registrations to traverse the SBC, the device must be configured with at least one User-type IP Group. These IP Groups represent a group of user agents that share the following characteristics:

- Perform registrations and share the same serving proxy\registrar
- Possess identical SIP and media behavior
- Reside on the same Layer-3 network and are associated with the same SRD

Typically, the device is configured as the user agent's outbound proxy and the device is configured (using the IP-to-IP Routing table) to route requests received from this IP Group to the serving proxy and vice versa. Survivability can be achieved using the alternative routing feature.

32.3.1 Initial Registration Request Processing

The device's handling of registration requests (REGISTER messages) are as follows:

- The device obtains the source URL from the SIP To header and the destination URL from the Request-URI.
- The device's classification process for REGISTER requests is the same as for other SIP messages. However, the REGISTER request must be received from **User-type** IP Groups only. If classification fails or the IP Group is not a User-type, the device rejects the registration request.
- The device's routing of REGISTER requests is done using the IP-to-IP Routing table. If the destination is a User-type IP Group, the device does not forward the registration; instead, it accepts (replies with a SIP 200 OK response) or rejects (SIP 4xx) the request, according to the user's IP Group configuration.
- If registration succeeds (replied with 200 OK by the IP PBX), the device adds a record to its Users Registration database that identifies the specific contact of the specific user (AOR). This record is used by the device to route subsequent requests to the specific user (in normal or in survivability modes).
- Alternative routing can be configured for REGISTER requests, in the IP-to-IP Routing table.
- The record in the device's database includes the SIP Contact header. Every REGISTER request is added to the database before manipulation, allowing correct user identification in the Classification process for the next received request.
- Call Admission Control (CAC) can be configured for incoming and outgoing REGISTER requests. For example, limiting REGISTER requests from a certain IP Group/SRD. Note that this is only for concurrent register dialogs and not concurrent registrations in the device's Users Registration database.
- The device can retain or change the original value of the SIP Expires header received from the user or proxy in the outgoing REGISTER message. This feature also applies when the device is in survivability mode (i.e., REGISTER requests cannot be forwarded to the proxy and is terminated by the device). This is configured by the SBCUserRegistrationTime, SBCProxyRegistrationTime, SBCSurvivabilityRegistrationTime, and SBCRandomizeExpires parameters.
- By default, the Contact header in the outgoing REGISTER is populated with a unique contact generated by the device and associated with the specific registration. Alternatively, the original user can be retained in the Contact header and used in the outgoing REGISTER request (using the SBCKeepContactUserinRegister parameter).

32.3.2 SBC Users Registration Database

The device manages a dynamic Users Registration database that is updated according to registration requests that traverse it. Each database entry for a user represents a binding between an AOR (obtained from the SIP To header) and one or more contact (obtained from the SIP Contact headers). Database bindings are added upon successful registration responses.

Database bindings are removed in the following cases:

- Successful de-registration responses (REGISTER with Expires header that equals zero).
- Registration failure responses.

- Timeout of the Expires header value (in scenarios where the user agent did not send a refresh registration request).

Note: The device's Users Registration database poses the following restrictions:

- The same contact cannot belong to more than one AOR.
- Contacts with identical URIs and different ports and transport types are not supported (same key is created).
- Multiple contacts in a single REGISTER is not supported.
- One database is shared between all User-type IP Groups.

32.3.3 Routing using Users Registration Database

The device uses the Users Registration database when routing calls of registered users. The device tries to locate a match for the IP-to-IP Routing rule between the incoming Request-URI and the following, listed in chronological order:

1. Unique Contact: the contact generated by the device and sent in the initial registration request to the serving proxy.
2. Registered AOR in the Users Registration database: the AOR of the incoming REGISTER request.
3. Registered Contact in the Users Registration database: the Contact of the incoming REGISTER request.

If registrations are destined to the database (using the above rules), the device does not attempt to find a database match, but instead replies with a SIP 200 OK (used for Survivability). Once a match is found, the request is routed either to the contact received in the initial registration or (if the device identifies that the user agent is behind a NAT) to the source IP address of the initial registration.

32.3.4 Registration Refreshes

Registration refreshes are incoming REGISTER requests that are associated with a registered user in the Users Registration database. These refreshes are routed to the serving proxy only if the serving proxy Expires time is about to expire; otherwise, the device responds with a 200 OK without routing the REGISTER. Each such refreshes also refresh the internal timer set on the device for this specific registration.

The device automatically notifies SIP Proxy / Registrar servers of users that are registered in the device's Users Registration database whose registration timeout has expired. When a user's registration timer expires, the device removes the user record from the database and sends an un-register notification (REGISTER message with the Expires header set to 0) to the Proxy/Registrar. This occurs only if a REGISTER message is sent to an IP Group destination type (in the IP-to-IP Routing table).

The device can be configured to add extra time (grace period) to the expiration timer of registered users in the database. If you configure this grace period, the device keeps the user in the database (and does not send an un-register to the Registrar server), allowing the user to send a "late" re-registration to the device. The device removes the user from the database only when this additional time expires. This featured is configured using the 'User Registration Grace Time' parameter (SBCUserRegistrationGraceTime).

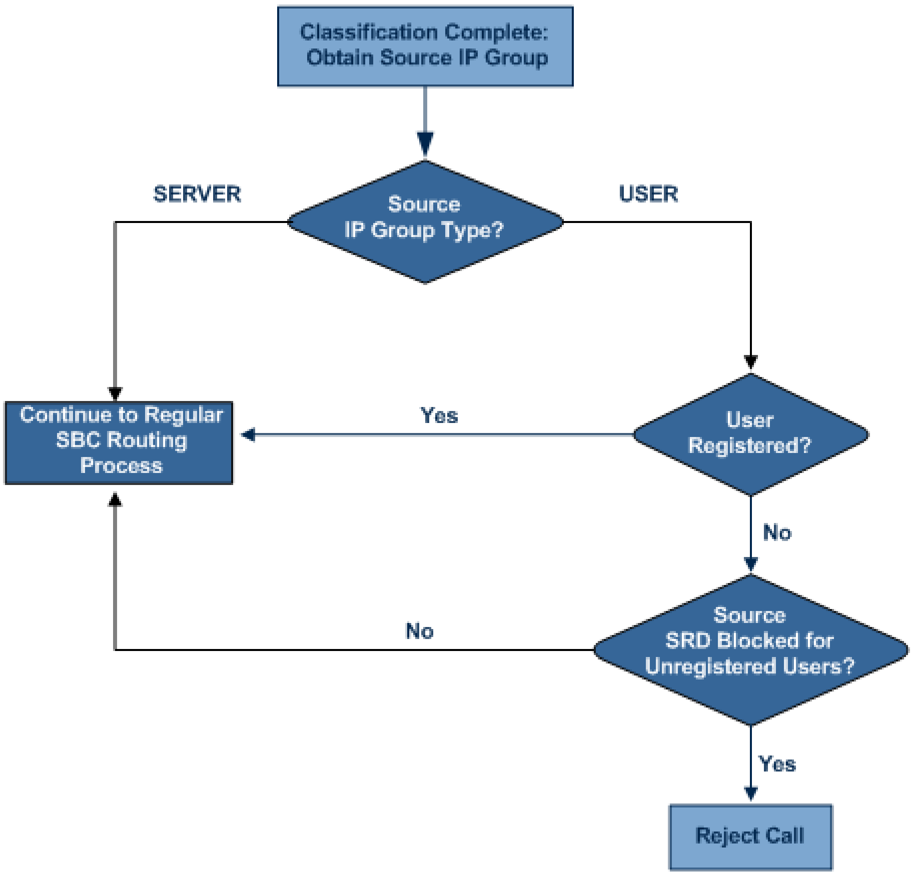
The device keeps registered users in its Users Registration database even if connectivity with the SIP proxy server is lost (i.e., proxy does not respond to users' registration refresh requests). The device removes users from the database only when their registration expiry time is reached (with the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

32.3.5 Registration Restriction Control

The device provides flexibility in controlling user registration:

- **Limiting Number of Registrations:** You can limit the number of users that can register with the device per IP Group and/or SRD. By default, no limitation exists for registered users. This is configured in the SRD and IP Group tables.
- **Blocking Incoming Calls from Unregistered Users:** You can block incoming calls (INVITE requests) from unregistered users (pertaining to User-type IP Groups). By default, calls from unregistered users are not blocked. This is configured in the SRD table. The flowchart below depicts the process for blocking unregistered users. When the call is rejected, the device sends a SIP 500 (Server Internal Error) response to the remote end.

Figure 32-2: Blocking Incoming Calls from Unregistered Users


Classification Complete: Obtain Source IP Group
Source IP Group Type?
SERVER
USER
Continue to Regular SBC Routing Process
Yes
User Registered?
No
No
Source SRD Blocked for Unregistered Users?
Yes
Reject Call

32.4 SBC Media Handling

Media behavior includes anything related to the establishment, management and termination of media sessions within the SIP protocol. Media sessions are created using the SIP "offer"/"answer" mechanism. If successful, the result is a bi-directional media (RTP) flow (e.g. audio, fax, modem, DTMF). Each offer/answer may create more than one media session of different types (e.g. audio and fax). In a SIP dialog, multiple offer/answer transactions may occur, each may change the media sessions characteristics (e.g. IP address, port, coders, media types, and RTP mode). The media capabilities exchanged in an offer/answer transaction include the following:

- Media types (Audio, Secure Audio, Video, Fax, Text...)

- IP addresses and ports of the media flow
- Media flow mode (send receive, receive only, send only, inactive)
- Media coders (coders and their characteristics used in each media flow)
- Other (standard or proprietary) media and session characteristics

Even though the device usually does not change the negotiated media capabilities (mainly performed by the remote user agents), it does examine the media exchange to control negotiated media types (if necessary) and to know how to open the RTP media channels (IP addresses, coder type, payload type etc.). The device forwards multiple video streams and text, as is.

The device interworks (normalization) the media (RTP-to-RTP, SRTP-to-RTP, and SRTP-to-SRTP) between its SBC legs. It "re-builds" specific fields in the RTP header when forwarding media packets. The main fields include the sequence number, SSRC, and timestamp.

The device is aware and sometimes active in the offer\answer process due to the following:

- NAT traversal: the device changes the SDP address to be its own address, thereby, resolving NAT problems.
- Firewall and security:
  - RTP pin holes - only RTP packets related to a successful offer\answer negotiation traverse the device: When the device initializes, there are no RTP pin holes opened, this means that each RTP\RTCP packets destined to the device are discarded. Once an offer\answer transaction ends successfully, an RTP pin hole is opened and RTP\RTCP flows between the two remote user agents. Once a pin hole is opened, the payload type and RTP header version is validated for each packet. RTP pin holes close if one of the associated SIP dialogs is closed (may also be due to broken connection).
  - Late rogue detection - once a dialog is disconnected, the related pin holes also disconnect.
  - Deep Packet inspection of the RTP that flows through the opened pin holes.
- Adding of media functionality to SIP user agents:
  - Transcoding (for a description on the transcoding modes, see "Transcoding Modes" on page 512)
  - Broken connection

According to the above functionalities, the call can be configured to operate in one of the following modes:

- **Media Anchoring without Transcoding (Transparent):** RTP traverses the device with minimal RTP packet changes (no DSP resources needed). This is typically used to solve NAT, firewall, and security issues. In this mode, all the "audio" coders in the received offer are included in the SBC outgoing offer. The Coder Table configuration has no effect on the coders in the outgoing offer. For more information, see "Media Anchoring without Transcoding (Transparent)" on page 508.
- Media Anchoring with Transcoding: RTP traverses the device and each leg uses a different coder or coder parameters (DSP resources are required). For more information, see Media Anchoring with Transcoding on page 509.
- **No Media Anchoring:** The RTP packet flow does not traverse the device. Instead, the two SIP UA's establish a direct RTP/SRTP flow between one another (see "No Media Anchoring" on page 511).

32.4.1 Media Anchoring without Transcoding (Transparent)

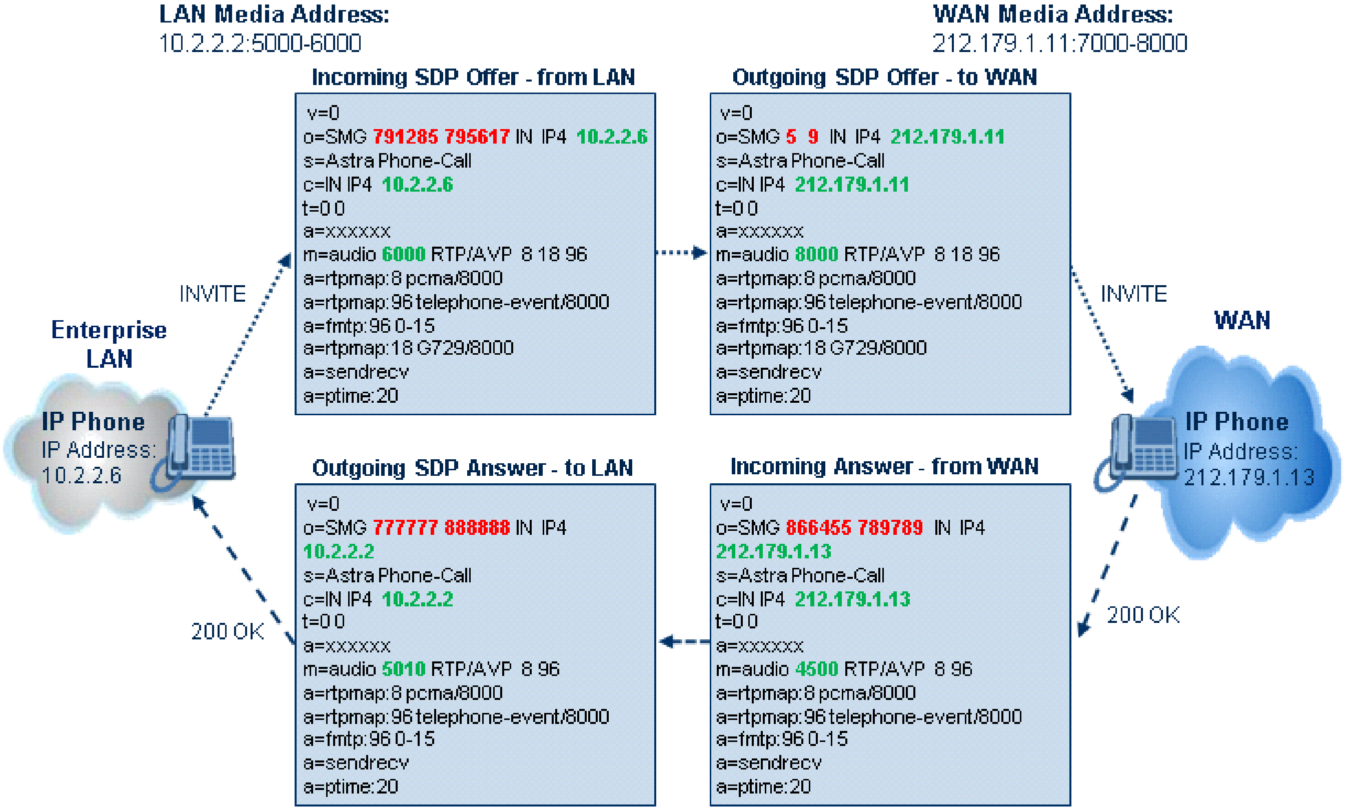
To direct the RTP to flow through the device (for NAT traversal, firewall and security), all IP address fields in the SDP are modified:

- Origin: IP address, session and version id

- Session connection attribute ('c=' field)
- Media connection attribute ('c=' field)
- Media port number
- RTCP media attribute IP address and port

Each SBC leg allocates and uses the device's local ports (e.g., for RTP\RTCP\fax). The local ports are allocated from a Media Realm associated with each leg. The legs are associated with a Media Realm as follows: If the leg's IP Group is configured with a Media Realm, then this is the associated Media Realm; otherwise, the leg's SRD Media Realm is the associated one. The figure below illustrates an example of SDP handling for a call between a LAN IP Phone 10.2.2.6 and a remote IP Phone 212.179.1.13 on the WAN.

Figure 32-3: SDP Offer/Answer Example



Incoming SDP Offer - from LAN

```
v=0
o=SMG 791285 795617 IN IP4 10.2.2.6
s=Astra Phone-Call
c=IN IP4 10.2.2.6
t=0 0
a=xxxxxx
m=audio 6000 RTP/AVP 8 18 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=rtpmap:18 G729/8000
a=sendrecv
a=ptime:20
```

Outgoing SDP Offer - to WAN

```
v=0
o=SMG 5 9 IN IP4 212.179.1.11
s=Astra Phone-Call
c=IN IP4 212.179.1.11
t=0 0
a=xxxxxx
m=audio 8000 RTP/AVP 8 18 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=rtpmap:18 G729/8000
a=sendrecv
a=ptime:20
```

Outgoing SDP Answer - to LAN

```
v=0
o=SMG 777777 888888 IN IP4
10.2.2.2
s=Astra Phone-Call
c=IN IP4 10.2.2.2
t=0 0
a=xxxxxx
m=audio 5010 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=sendrecv
a=ptime:20
```

Incoming Answer - from WAN

```
v=0
o=SMG 866455 789789 IN IP4
212.179.1.13
s=Astra Phone-Call
c=IN IP4 212.179.1.13
t=0 0
a=xxxxxx
m=audio 4500 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=sendrecv
a=ptime:20
```

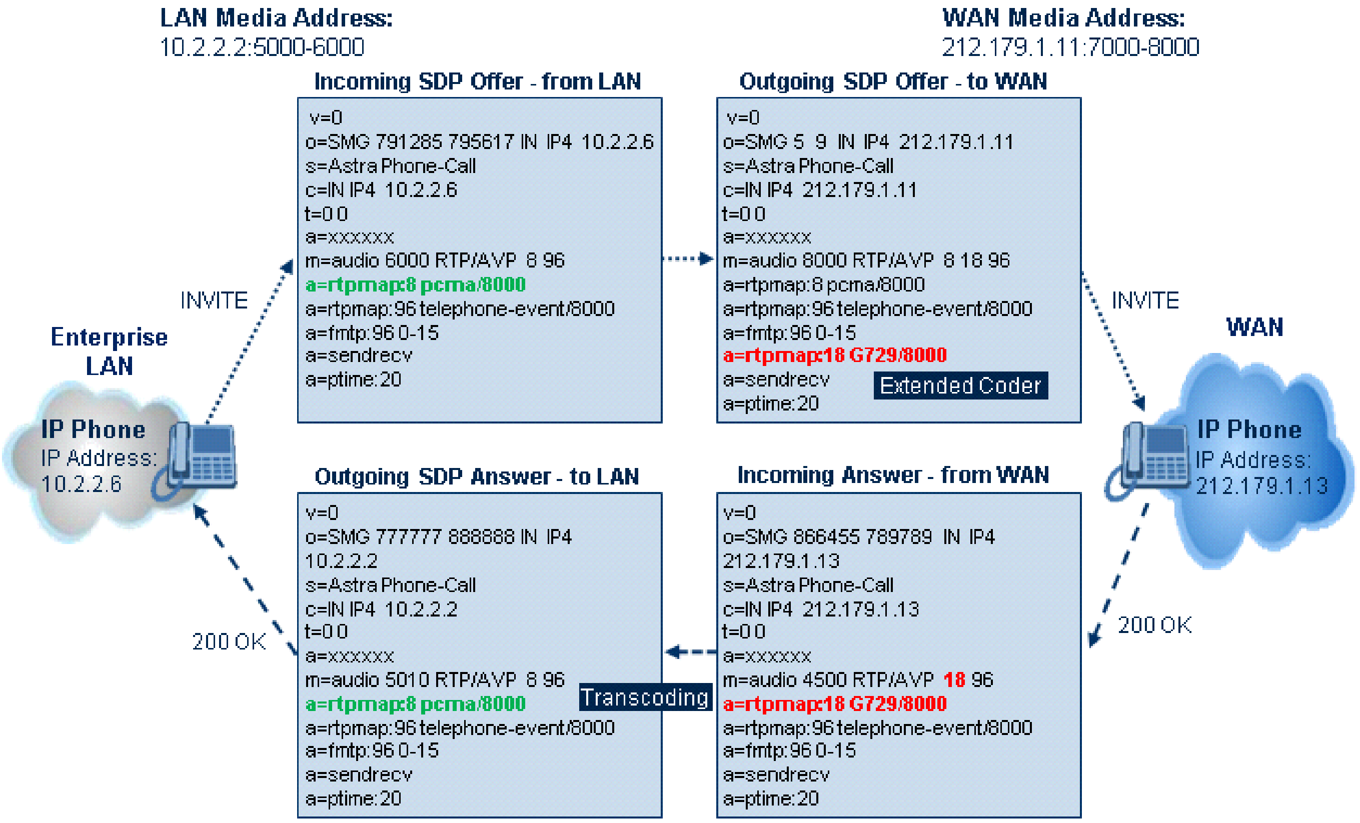
32.4.2 Media Anchoring with Transcoding

The device performs transcoding when there are no common coders between the two user agents (i.e., the SDP answer from one user agent doesn't include any coder included in the offer previously sent by the other user agent). For transcoding, the device can be configured to add media capabilities to user agents pertaining to a specific IP Group, and then perform transcoding in cases where the selected coder in the answer SDP is not one that appears in the original offer. The capabilities that can be added are one or more of the device's supported coders and are configured by using the parameter SBCExtensionCodersGroupID (points to a coders list) in the IP Profile table (which is assigned to the IP Group). Therefore, to allow user agents of different IP Groups to communicate with each other (regardless of their capabilities), an extended coders table with at least one coder that is supported by each IP Groups' user agents needs to be assigned to each IP Group. Therefore, each offer destined to specific IP Groups include this coder.

In the scenario depicted in the figure below, the IP phone on the LAN side initiates a call to the IP phone on the WAN. The initial SDP offer (from the LAN leg) includes codec G.711

as its supported codec. Since this is sent to a Destination IP Group that is configured with an extended coder list, on the WAN leg the device adds another supported codec G.729 to the SDP, which is now offered to the WAN IP phone. The WAN IP phone chooses the extended codec (G.729) in its SDP answer to the device's WAN leg. Since this codec was not included in the original incoming offer, the device performs transcoding (between G.729 and G.711) between its two legs, allowing the streaming of media to occur.

Figure 32-4: Transcoding using Extended Coders (Example)



Incoming SDP Offer - from LAN

```
v=0
o=SMG 791285 795617 IN IP4 10.2.2.6
s=Astra Phone-Call
c=IN IP4 10.2.2.6
t=0 0
a=xxxxxx
m=audio 6000 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=sendrecv
a=ptime:20
```

Outgoing SDP Offer - to WAN

```
v=0
o=SMG 5 9 IN IP4 212.179.1.11
s=Astra Phone-Call
c=IN IP4 212.179.1.11
t=0 0
a=xxxxxx
m=audio 8000 RTP/AVP 8 18 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=rtpmap:18 G729/8000
a=sendrecv
a=ptime:20
```

Outgoing SDP Answer - to LAN

```
v=0
o=SMG 777777 888888 IN IP4
10.2.2.2
s=Astra Phone-Call
c=IN IP4 10.2.2.2
t=0 0
a=xxxxxx
m=audio 5010 RTP/AVP 8 96
a=rtpmap:8 pcma/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=sendrecv
a=ptime:20
```

Incoming Answer - from WAN

```
v=0
o=SMG 866455 789789 IN IP4
212.179.1.13
s=Astra Phone-Call
c=IN IP4 212.179.1.13
t=0 0
a=xxxxxx
m=audio 4500 RTP/AVP 18 96
a=rtpmap:18 G729/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=sendrecv
a=ptime:20
```

For an SDP offer to provide an extended coder list to a remote user agent, the following prerequisites must be fulfilled:

- An extended coders list has been configured for the user agent's IP Group (i.e., Destination IP Group)
- The incoming offer contains at least one supported coder (otherwise, transcoding can't be performed)
- Both legs have available DSP's
- T.38 doesn't appear in the offer

If the above prerequisites are not met, the SDP offer is sent without the extended coders list. The coders from the extended list are added after the ones from the original offer (decreases transcoding probability). Coders common between the extended coders list and those in the original SDP offer are not added. Transcoding may be performed even in scenarios when the same coder has been chosen - this occurs if the coders use different coder parameters (e.g. rate and packetization time).

The device also supports early media, whereby the first offer\answer transaction is finalized and the media flow starts before the SIP call is connected (before the INVITE 200 OK response). The offer and 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 ReINVITE SIP transactions. The media handling is similar to the original offer/answer handling. If the offer is rejected by the remote party, then no media changes occur (e.g. INVITE without SDP, then 200 OK and ACK, offer\answer within an offer/answer, and Hold ReINVITE with IP address of 0.0.0.0 - IP address is unchanged).

32.4.3 No Media Anchoring

The No Media Anchoring (commonly referred to as Anti-Tromboning) feature enables the use of SBC signaling capabilities without handling the media (RTP/SRTP) flow between remote SIP user agents (UA). The media flow does not traverse the device. Instead, the two SIP UAs establish a direct media flow (i.e., direct call) between one another. Signaling continues to traverse the device with minimal intermediation and involvement to enable certain SBC abilities such as routing.

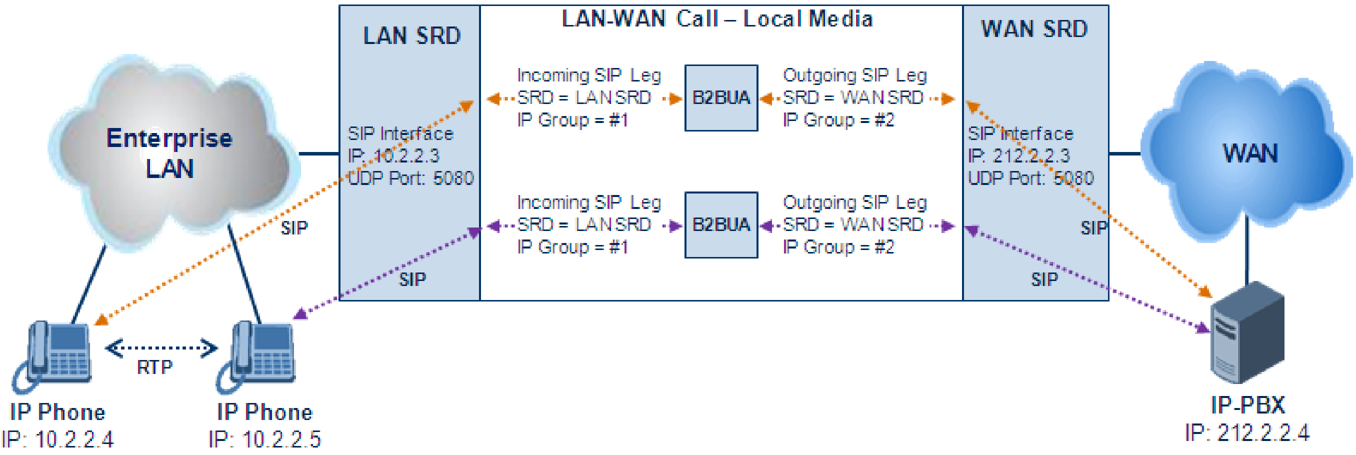
By default, media packets traverse the device to solve NAT problems, enforce media security policy, perform media transcoding between the two legs, and media monitoring. In certain deployments, specific calls do not require media anchoring, for example, when there is no need for NAT, security, or transcoding.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

- Internal LAN calls: When the SBC routes a call between two UAs within the same LAN, the SBC can forward the SDP directly between caller and callee, and direct the media to flow between the UAs without traversing the SBC.
- Internal LAN calls via WAN: In this setup, the SBC dynamically identifies the call as between UAs located in the same network (i.e., LAN) and thereby, directs the media to flow between these UAs without traversing the SBC.

The No Media Anchoring feature is typically implemented in the following scenarios:

- The device is located within the LAN.
- Calls between two SIP UAs in the same LAN and signaling is sent to a SIP proxy server (or hosted IP PBX) located in the WAN.
- The device does not need to perform NAT traversal (for media) and all the users are in the same domain.

Figure 32-5: SBC SIP Signaling without RTP Media Flow



The benefits of implementing the No Media Anchoring feature include the following:

- Saves network bandwidth

- Reduces CPU usage (no media handling)
- Avoids interference in SDP negotiation and header manipulation on RTP/SRTP

The device handles the No Media Anchoring process as follows:

1. Identifies a No Media Anchoring call according to configuration and the call's properties (such as source, destination, IP Group, and SRD).
2. Handles the identified No Media Anchoring call.

The No Media Anchoring feature is enabled for all calls (regardless of SRD), using the global parameter, SBCDirectMedia. You can also enable No Media Anchoring per SRD (in the SRD table), whereby calls belonging to this same SRD (source and destination) are handled as No Media Anchoring (direct media) calls. This occurs even if the global parameter is disabled.

Notes:

- No Media Anchoring can be used when the SBC does not do NAT traversal (for media) where all the users are in the same domain.
- No Media Anchoring calls cannot operate with the following features:
  - √ Manipulation of SDP data (offer/answer transaction) such as ports, IP address, coders
  - √ 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 No Media Anchoring calls. Restricted coders are removed from the SDP offer message.
- For No Media Anchoring, opening of voice channels and allocation of IP media ports are not required.
- When two UAs belong to the same SRD which is enabled for No Media Anchoring, and one of the UAs is defined as a foreign user (example, "follow me service") located in the WAN while the other UA is located in the LAN: calls between these two UAs cannot be established until the No Media Anchoring for the SRD is disabled, as the device does not interfere in the SIP signaling. In other words, parameters such as IP addresses are not manipulated for calls between LAN and WAN (although required).
- When the global parameter SBCDirectMedia is disabled, No Media Anchoring can only occur for calls between UAs belonging to the same SRD that is configured for No Media Anchoring in the SRD table.

32.4.4 Transcoding Modes

The device supports the configuration of the voice transcoding mode (media negotiation) between the two SBC legs. The device can be configured to perform transcoding only when necessary. Typically, the SBC passes RTP packets transparently (RTP-to-RTP) between the two user agents. If the device is configured to always perform transcoding, then transcoding is performed on the outgoing SBC leg and the device's SBC application interworks the media by implementing transcoding (since both legs have different media capabilities).

In the SBC application, forced transcoding of voice in an SBC session allows the device to receive capabilities that are not negotiated between the SBC legs. For example, if on the SBC session you want to force Gain Control to use voice transcoding even though both sides of the session have negotiated without SBC intervention (for example, coder extension).

The transcoding mode can be configured globally, using the TranscodingMode parameter or for specific calls, using the IP Profile table.

Note: 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.

32.4.5 Restricting Coders

The SBC Allowed Coders (coders restriction) feature determines the coders that can be used for a specific SBC leg. This provides greater control over bandwidth by enforcing the use of specific coders (*allowed coders groups*) while preventing the use of other coders. This is done by defining a group of allowed coders for the SBC leg, as described below:

1. Configure a Coders Group for allowed coders, using the AllowedCodersGroup parameter.
2. Select this Coders Group using the SBCAllowedCodersGroupID parameter of the IP Profile table.
3. Enable this feature by setting the SBCAllowedCodersMode parameter of the IP Profile table to **Restriction**.

Coders that are not listed (including unknown coders) in the Allowed Coders Group are removed from the SDP offer. Therefore, only coders common between the SDP offer and Allowed Coders Group are used. If the SDP offer does not list any of the Allowed Coders, the call is rejected, unless transcoding is configured using Extension coders (see Coder Transcoding on page 513).

Notes:

- For a list of supported coders, see "Configuring Default Coders" on page 307.
- Allowed Coder Groups are applicable only to audio media.

The Allowed Coders process is as follows:

a. The device receives an incoming SIP message with SDP (offer) and checks the offered coders.

b. The source (first) leg may have Allowed Coders (i.e. list of coders that can be used - enforced).

c. The device checks for common coders between the SDP offered coders and the Allowed Coders Group list.

For example, assume the following:

- The SDP coder offer includes the following coders: G.729, G.711, and G.723.
- The source (first) leg includes the following Allowed Coders: G.711 and G.729.

The device selects the common coders, i.e., G.711 and G.729 (with changed preferred coder priority - highest for G.711). In other words, it removes the coders that are not in the Allowed Coders list and the order of priority is first according to the Allowed Coders list.

32.4.6 Coder Transcoding

The device can add coders, referred to as *Extension coders* to the SDP offer in the outgoing leg.

For example, assume the following:

- The SDP coder offer includes the following coders: G.729, G.711, and G.723.
- The incoming leg includes the following Allowed coders: G.711 and G.729.

1. The device selects the common coders, i.e., G.711 and G.729 (with changed preferred coder priority - highest for G.711). In other words, it removes the coders that are not in the Allowed coders list and the order of priority is first according to the Allowed coders list.
2. Assuming that the outgoing leg also includes Allowed coders and/or Extension coders: The device performs the Allowed coders procedure (common coders) between the updated coder list and the outgoing leg's Allowed coders. The Extension coders procedure is performed before Allowed coders.
3. Adding to the example, assume the following:
    - For the outgoing leg, the device selects the common coders G.711 and G.729 (explained in the example above).
    - Outgoing leg includes the Extended coder G.726.
    - Outgoing leg includes the following Allowed coders: G.723, G.726, and G.729.

    As a result, the device selects the common coders, i.e., G.729 and G.726 (coder priority did not change Extension coder order).

If the Allowed coders policy on the SDP incoming leg returns an empty coders list, the device rejects the call (SIP 488 or ACK and BYE). If both Extension coders and Allowed coders policies on SDP (in this order) returns an empty coders list, the outgoing leg rejects the call (SIP 488, or ACK and BYE).

Below is an example, assuming that Allowed Coders list (ordered) includes G711A-law (PCMA), G729, and G711U-law (PCMU), and Extension Coder is G729.

1. SDP offer - original offer:

```
m=audio 6050 RTP/AVP 0 8 4 96
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
```

2. SDP offer - after manipulation:

```
m=audio 6010 RTP/AVP 8 0 96 18
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
a=rtpmap:18 G729/8000
a=fmtp:18 annexb=no
```

In the SDP, the "m=audio 6010 RTP/AVP 8 0 96 18" line shows that the coder priority has changed - G.711A-law ("8") and then G.711U-law ("0") - and that the Extension coder G.729 ("18") has been added. The G.723 coder ("4) in the original offer was removed as it was not defined in the Allowed Coders list (i.e., a restricted coder).

➢ **To configure Extension coders:**

1. In the Coders Group table (see "Configuring Coder Groups" on page 310), configure a

Coders Group for extension coders.

2. In the IP Profile table, select this Coders Group in the 'Extension Coders Group ID' parameter.
3. In the IP Profile table, enable this feature by setting the 'Allowed Coders Mode' parameter to **Restriction** or **Restriction and Preference**.

32.4.7 Prioritizing Coder List in SDP Offer

In addition to restricting the use of coders with Allowed coders, you can prioritize the coders listed in the SDP offer. This feature is referred to as *Coder Preference*. This is done on both SBC legs:

- **Incoming SDP offer:** The device arranges the coder list according to the order in the Allowed Coders Group table. The coders listed higher up in the table take preference over ones listed lower down in the table. This feature is enabled by setting the 'Allowed Coders Mode' parameter in the IP Profile table to **Preference** or **Restriction and Preference**. If set to **Preference**, in addition to the Allowed coders that are listed first in the SDP offer, the original coders received in the SDP are retained and listed after the Allowed coders. Thus, this mode does not necessarily restrict coder use to Allowed coders, but uses (prefers) the Allowed coders whenever possible.
- **Outgoing SDP offer:** If only Allowed coders are used, the coders are arranged in the SDP offer as described above. However, if Extension coders are also used, the coder list is arranged according to the 'SBC Preferences Mode' parameter in the IP Profile table. This parameter can be configured to add the Extension coders after the Allowed coders (i.e., at the end of the list - default) according to their order in the Coders Group table, or arrange Allowed and Extension coders according to their position in the Coders Group table.

32.4.8 SRTP-RTP and SRTP-SRTP Transcoding

The device supports transcoding between SRTP and RTP. The device can also enforce specific SBC legs to use SRTP and/or RTP. The device’s handling of SRTP/RTP is configured using the IP Profile parameter, SBCMediaSecurityBehaviour, which provides the following options:

- SBC passes the media as is, regardless of whether it’s RTP or SRTP (default).
- SBC legs negotiate only SRTP media lines (m=); RTP media lines are removed from the incoming SDP offer\answer.
- SBC legs negotiate only RTP media lines; SRTP media lines are removed from the incoming offer\answer.
- Each SDP offer\answer is extended (if not already) to two media lines for RTP and SRTP.

If after SDP offer\answer negotiation, one SBC leg uses RTP while the other uses SRTP, then the device performs RTP-SRTP transcoding. To translate between RTP and SRTP, the following prerequisites must be met:

- At least one supported SDP "crypto" attribute.
- The EnableMediaSecurity parameter must be set to 1.

Transcoding where both legs are configured for SRTP is typically required to trans-encrypt and trans-decrypt. This is relevant when the MKI and Symmetric MKI parameters are enabled. In other words, both sides need to both encrypt and decrypt the outgoing and incoming SRTP packets, respectively.

Channel (DSP) resources are not required for RTP and SRTP transcoding.

32.4.9 Multiple RTP Media Streams per Call Session

The device's SBC application supports multiple RTP media streams per SBC call session. Up to five different media types can be included in a session:

- Audio (m=audio)
- Video (m=video)
- Text (m=text)
- Fax (m=image)

Therefore, the device can provide transcoding of various attributes in the SDP offer/answer (e.g., codec, port, and packetization time) per media type. If the device is unable to perform transcoding (for example, does not support the codec), it relays the SBC dialog transparently.

32.4.10 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 this 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.

32.5 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 IP Profile and Coder Group Settings tables. The IP Profile table determines the supported fax coders and 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 314.

> **Note:** The voice-related coder configuration (Allowed and Extended 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.

32.6 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.

32.7 SBC Authentication

The device can authenticate SIP servers and SBC users (clients). The different authentication methods are described in the subsequent subsections.

32.7.1 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 618). 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 270).

32.7.2 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.

32.8 Interworking SIP Signaling

The device supports interworking of SIP signaling messages to ensure interoperability between communicating SIP UAs or entities. This is critical in network environments where the UAs on opposing SBC legs have different SIP signaling support. For example, some UAs may support different versions of a SIP method while others may not event support a specific SIP method. The configuration method for assigning specific SIP message handling modes to UAs, includes configuring an IP Profile with the required interworking mode, and then assigning the IP Profile to the relevant IP Group.

This section describes some of the device's support for handling SIP methods to ensure interoperability.

32.8.1 Interworking SIP 3xx Redirect Responses

The device supports interworking of SIP 3xx redirect responses. By default, the device's handling of SIP 3xx responses is to send the Contact header unchanged. However, some SIP UAs may support different versions of the SIP 3xx standard while others may not even support SIP 3xx.

The handling of SIP 3xx can be configured for all calls, using the global parameter SBC3xxBehavior. For configuring different SIP 3xx handling options for different UAs (i.e., per IP Group), use the IP Profile table parameter, 'SBC Remote 3xx Behavior'.

32.8.1.1 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:

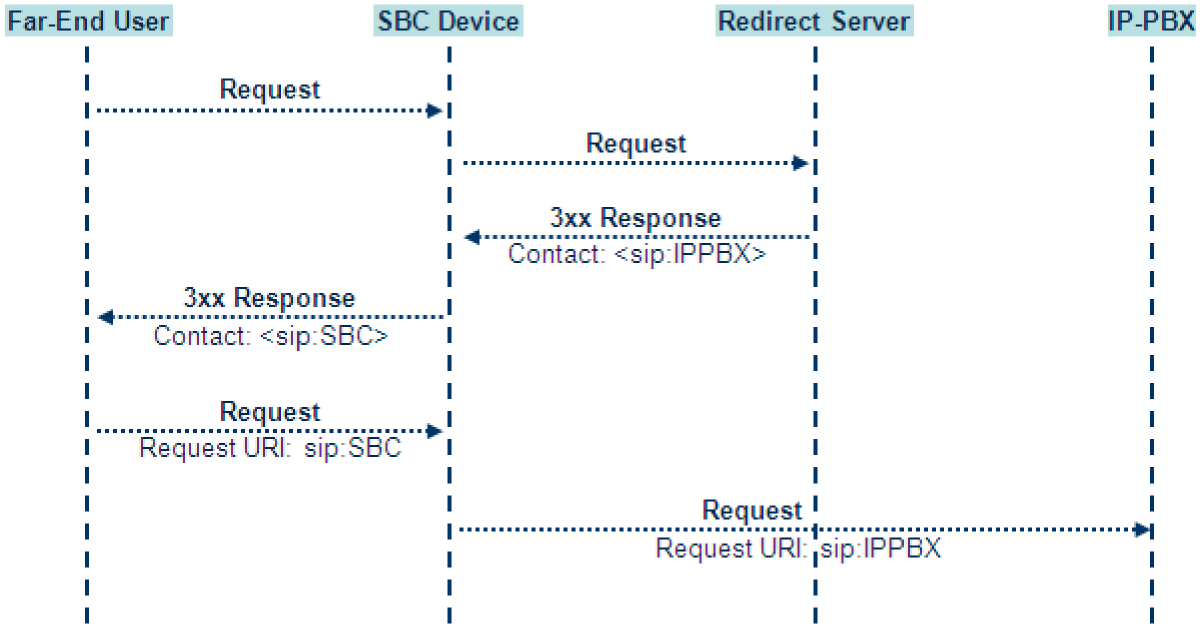
- Changes the host part to the device's IP address – this change causes the remote user agent to send the INVITE to the device.
- Adds a special prefix ("T~&R_") to the Contact user part – to identify the new INVITE as a 3xx resultant INVITE.

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 32-6: 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).

32.8.1.2 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).

32.8.2 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 32-1: Handling of SIP Diversion and History-Info Headers

Parameter Value	SIP Header Present in Received SIP Message		
	Diversion	History-Info	Diversion and History-Info
HistoryInfoMode = Add **DiversionMode = Remove**	Diversion converted to History-Info. Diversion removed.	Not present	Diversion removed.
HistoryInfoMode = Remove **DiversionMode = Add**	Not present.	History-Info converted to Diversion. History-Info removed.	History-Info added to Diversion. History-Info removed.
HistoryInfoMode = Disable **DiversionMode = Add**	Diversion converted to History-Info.	Not present.	Diversion added to History-Info.
HistoryInfoMode = Disable **DiversionMode = Add**	Not present.	History-Info converted to Diversion.	History-Info added to Diversion.
HistoryInfoMode = Add **DiversionMode = Add**	Diversion converted to History-Info.	History-Info converted to Diversion.	Headers are synced and sent.
HistoryInfoMode = Remove **DiversionMode = Remove**	Diversion removed.	History-Info removed.	Both removed.

32.8.3 Interworking SIP REFER Messages

The device supports interworking of SIP REFER messages. SIP UAs may support different versions of the REFER standard while others may not even support REFER.

This feature supports the following:

- Attended, unattended, and semi-attended call transfers
- Sending INVITE, REFER-notifications, BYE, PRACK and Session Timer on behalf of

peer PBXs

- Advanced routing rules for the new, initiated INVITE
- Forwarding early media after REFER while attempting to avoid transcoding (by sending session update)
- Interoperate with environments were different SIP UAs lack basic SIP functionality such as re-INVITE, UPDATE, PRACK, Delayed Offer, re-INVITE without SDP
- Session updates after connect to avoid transcoding

The handling of REFER can be configured for all calls, using the global parameter SBCReferBehavior. For configuring different REFER handling options for different UAs (i.e., IP Groups), use the IP Profile table parameter, 'SBC Remote Refer Behavior'.

- Local handling of REFER: This option is used for UAs that do not support REFER. Upon receipt of a REFER request, instead of forwarding it to the IP Group, the device handles it locally. It generates a new INVITE to the alternative destination according to the rules in the IP-to-IP Routing table (where the 'Call Trigger' field is set to **REFER**). It is also possible to specify the IP Group that sent the REFER request, as matching criteria for the re-routing rule in this table ('ReRoute IP Group ID' field).
- Transparent handling: The device forwards the REFER with the Refer-To header unchanged.
- Re-routing through SBC: The device changes the Refer-To header so that the re-routed INVITE is sent through the SBC application.
- IP Group Name: The device sets the host part in the REFER message to the name configured for the IP Group in the IP Group table.

32.8.4 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.

32.8.5 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'.

32.8.6 Interworking SIP Early Media

The device supports various interworking modes for SIP early media between SIP UAs (i.e., IP Groups):

- **Early Media Enabling:** The device supports the interworking of early media between SIP UAs that support early media and those that do not support receipt of early media. Early media can arrive in provisional responses to an INVITE request. The device forwards the request of early media for IP Groups that support this capability; otherwise, the device terminates it. Provisional responses whose SDP are suppressed are changed to a SIP 180 response. This feature is also supported for delayed offers. This is configured using the IP Profile parameter, 'SBC Remote Early Media Support'. The device refers to this parameter also for features that require early media such as playing ringback tone.
- **Early Media Response Type:** The device supports the interworking of different SIP provisional response types between UAs for forwarding the early media to the caller. This can support all early media response types (default), SIP 180 only, or SIP 183 only, and is configured by the IP Profile parameter, 'SBC Remote Early Media Response Type'.
- **Multiple 18x:** The device supports the interworking of different support for multiple 18x responses (including 180 Ringing, 181 Call is Being Forwarded, 182 Call Queued, and 183 Session Progress) that are forwarded to the caller. The UA can be configured as supporting only receipt of the first 18x response (i.e., the device forwards only this response to the caller), or receipt of multiple 18x responses (default). This is configured by the IP Profile parameter, 'SBC Remote Multiple 18x Support'.
- **Early Media RTP:** The device supports the interworking with remote clients that send 18x responses with early media and whose subsequent RTP is delayed, and with remote clients that do not support this and require RTP to immediately follow the 18x response. Some clients do not support 18x with early media, while others require 18x with early media (i.e., they cannot play ringback tone locally). These various interworking capabilities are configured by the IP Profile parameters, 'SBC Remote Early Media RTP', 'SBC Remote Supports RFC 3960', and 'SBC Remote Can Play Ringback'. See the flowcharts below for the device's handling of such scenarios:

Figure 32-7: SBC Early Media RTP 18x without SDP

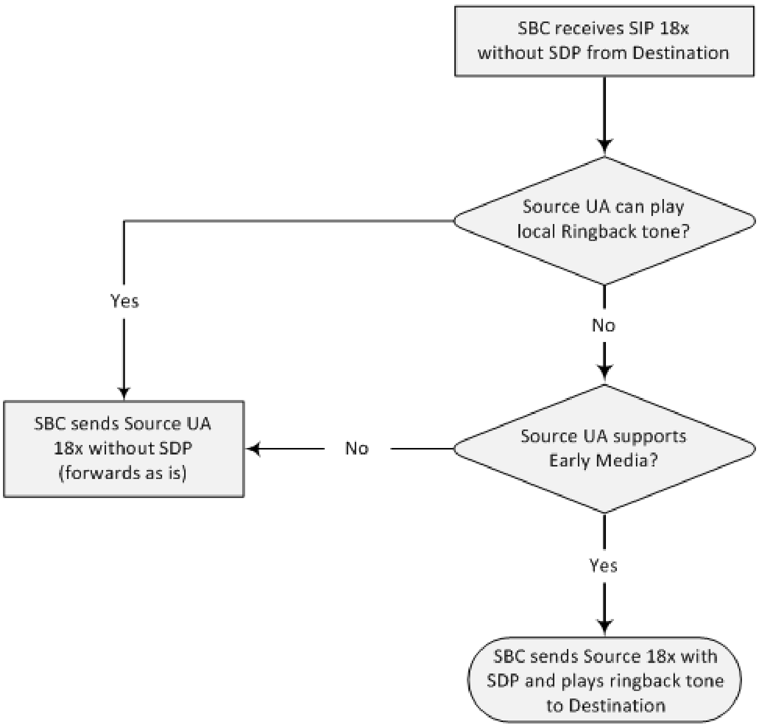
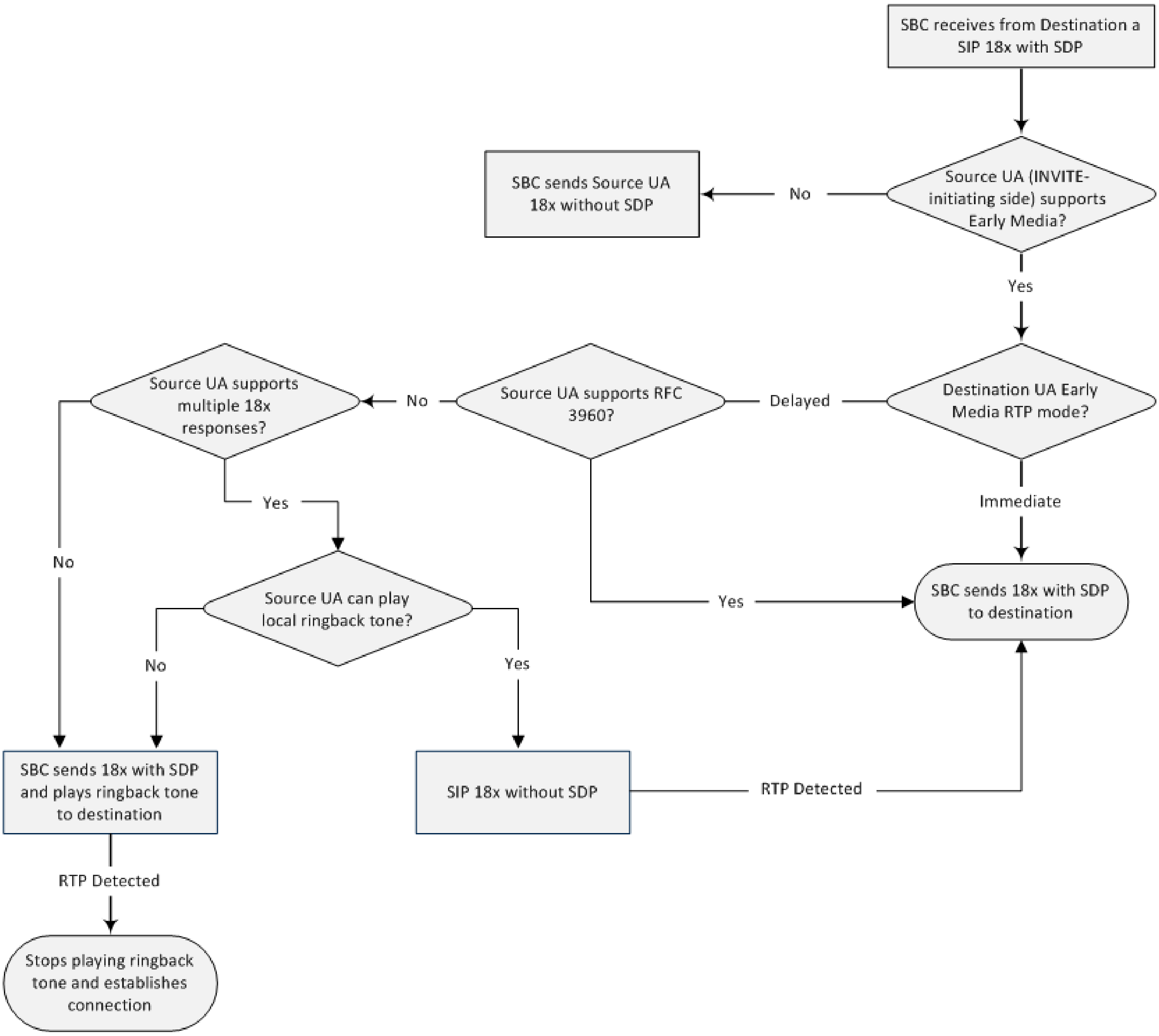

SBC receives SIP 18x without SDP from Destination
Source UA can play local Ringback tone?
Yes
No
SBC sends Source UA 18x without SDP (forwards as is)
No
Source UA supports Early Media?
Yes
SBC sends Source 18x with SDP and plays ringback tone to Destination

Figure 32-8: Early Media RTP - SIP 18x with SDP


SBC receives from Destination a SIP 18x with SDP
Source UA (INVITE-initiating side) supports Early Media?
No
SBC sends Source UA 18x without SDP
Yes
Destination UA Early Media RTP mode?
Delayed
Source UA supports RFC 3960?
No
Source UA supports multiple 18x responses?
Immediate
Yes
SBC sends 18x with SDP to destination
Yes
Source UA can play local ringback tone?
No
No
Yes
SBC sends 18x with SDP and plays ringback tone to destination
SIP 18x without SDP
RTP Detected
RTP Detected
Stops playing ringback tone and establishes connection

32.8.7 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'.

32.8.8 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'.

32.8.9 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.

32.8.10 Interworking Delayed Offer

The device enables sessions between endpoints (IP Groups) that send INVITEs without SDP (i.e., delayed media) and those that do not support the receipt of INVITEs without SDP. The device creates an SDP and adds it to INVITEs that arrive without SDP. This intervention in the SDP offer/answer process may require transcoding. Delayed offer is also supported when early media is present.

The interworking of delayed offer is configured using the IP Profile parameter 'SBC Remote Delayed Offer Support'.

Note: For this feature to function properly, a valid Extension Coders Group ID needs to be configured for IP Profiles that do not support delayed offer.

32.8.11 Interworking Call Hold

The device supports the interworking of call hold / retrieve requests between IP entities supporting different call hold capabilities:

- Interworking SDP call hold formats. This is configured by the IP Profile parameter, 'SBC Remote Hold Format'.
- Interworking the play of the held tone for IP entities that cannot play held tones locally. This is configured by the IP Profile parameter, 'SBC Play Held Tone'.
- Interworking generation of held tone where the device generates the tone to the held party instead of the call hold initiator. This is configured by the IP Profile parameter, 'SBC Reliable Held Tone Source'.

For configuring IP Profiles, see "Configuring IP Profiles" on page 314.

32.9 Call Survivability

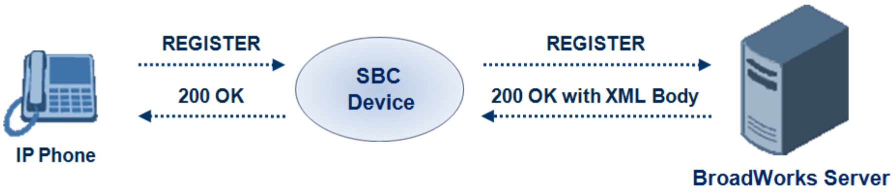
This section describes various call survivability features supported by the SBC device.

32.9.1 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 32-9: 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:

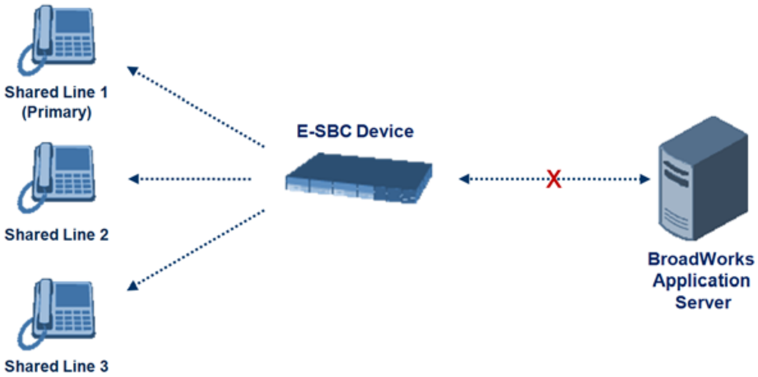
```
<?xml version="1.0" encoding="utf-8"?>
  <BroadsoftDocument version="1.0" content="subscriberData">
    <phoneNumbers>
      <phoneNumber>2403645317</phoneNumber>
      <phoneNumber>4482541321</phoneNumber>
    </phoneNumbers>
    <aliases>
      <alias>sip:bob@broadsoft.com</alias>
      <alias>sip:rhughes@broadsoft.com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

32.9.2 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 531. Note that incoming calls specific to extensions 601 or 602 ring only at these specific extensions.

Figure 32-10: 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 270), 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 552), 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 565), add a manipulation rule for the secondary extensions (e.g., 601 and 602) so that they also register to the device's database under the primary extension contact (e.g., 600):
   - Set the 'Manipulation Purpose' field to **Shared Line**.

- Set the 'Source IP Group' field to the IP Group ID that you created for the users (e.g., 2).
- Set the 'Source Username Prefix' field to represent the secondary extensions (e.g., 601 and 602).
- Set the 'Manipulated URI' field to **Source** to manipulate the source URI.
- Set the 'Remove From Right' field to "1" to remove the last digit of the extensions (e.g., 601 is changed to 60).
- Set the 'Suffix to Add' field to "0" to add 0 to the end of the manipulated number (e.g., 60 is changed to 600).

32.9.3 Call Survivability for Call Centers

The device supports call survivability for call centers. When a communication failure (e.g., in the network) occurs with the remote voice application server responsible for handling the call center application (such as IVR), the device routes the incoming calls received from the customer (i.e., from the TDM gateway) to the call center agents.

In normal operation, the device registers the agents in its users registration database. Calls received from the TDM gateway are forwarded by the device to the application server, which processes the calls and sends them to specific call center agents, through the device. Upon a failure with the application server, the device routes the calls from the TDM Gateway to the agents. The device routes the call to the first available user it founds. If the call is not answered by the user, the device routes it to the next available user. The SBC can handle a sequence of up to five users, after which the session is timed out and the call is dropped.

Figure 32-11: Normal Operation in Call Center Application

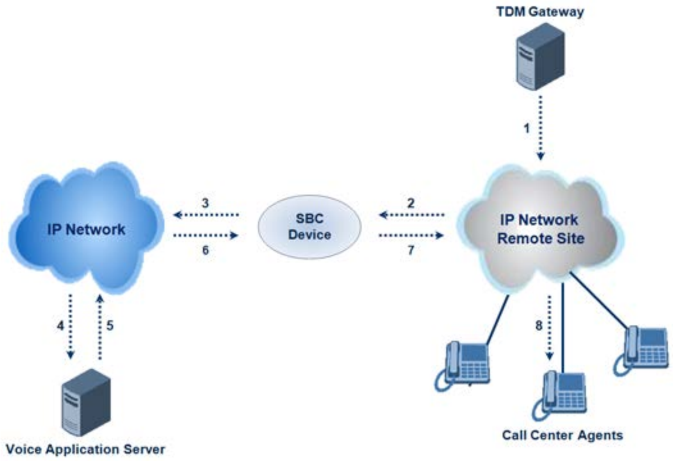
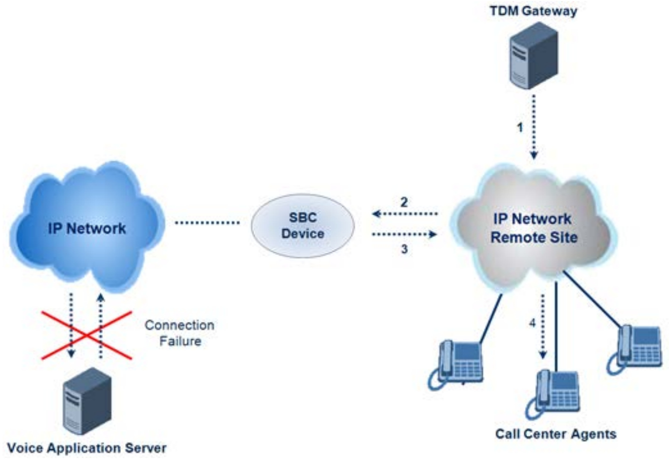


Figure 32-12: Call Survivability for Call Center

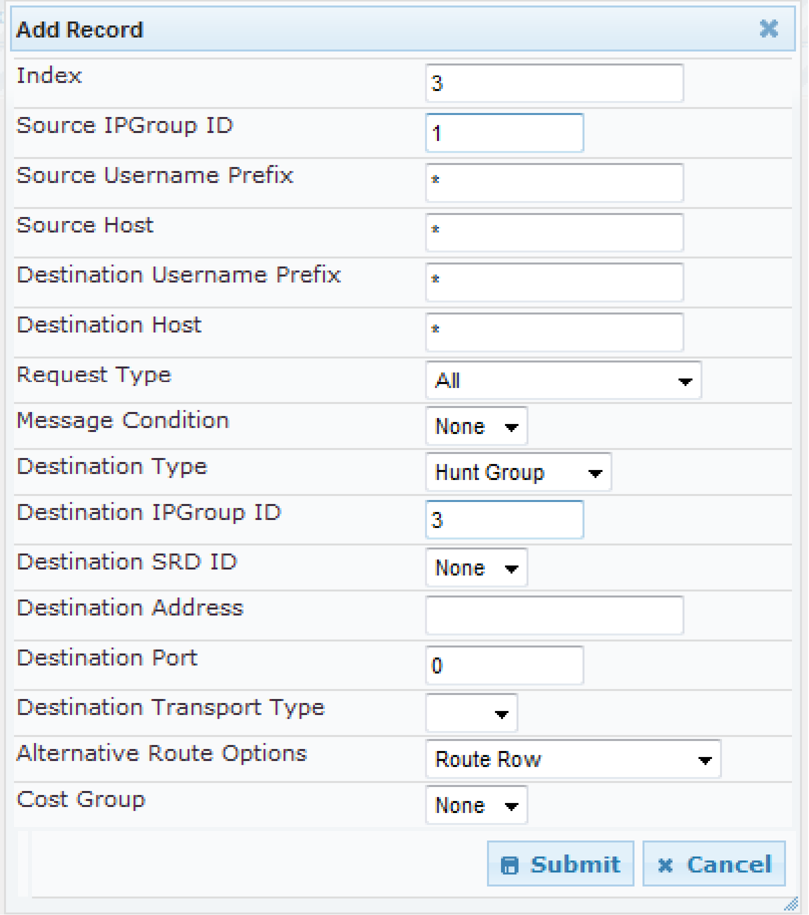


➢ To configure call survivability for a call center application:

1. In the IP Group table (see "Configuring IP Groups" on page 270), 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 545), 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 552), add the following IP-to-IP routing rules:
   - For normal operation:
     - Routing from TDM Gateway to Application server.
     - Routing from Application server to call center agents.
   - For call survivability mode: Routing from TDM Gateway to call center agents. This configuration is unique due to the following settings:
     - The 'Source IP Group ID' field is set to the IP Group of the TDM Gateway.
     - The 'Destination Type' field is set to **Hunt Group**, which is specifically used for call center survivability.
     - The 'Destination IP Group ID' field is set to the IP Group of the call center agents.

The figure below displays a routing rule example, assuming IP Group "1" represents the TDM Gateway and IP Group "3" represents the call center agents:

Figure 32-13: 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

32.9.4 Survivability Mode Display on Aastra IP Phones

If the SBC device is deployed in an Enterprise network with Aastra IP phones and connectivity with the WAN fails, the device provides call survivability by enabling communication between IP phone users within the LAN enterprise. In such a scenario, the device can be configured to notify the IP phones that it is currently operating in Survivability mode. When this occurs, the Aastra IP phones display the message, "StandAlone Mode" on their LCD screens. This feature is enabled by setting the SBCEnableAASTRASurvivabilityNotice parameter to 1.

When this feature is enabled and the SBC device is in Survivability mode, it responds to SIP REGISTER messages from the IP phones with a SIP 200 OK containing the following XML body:

```
Content-Type: application/xml
<?xml version="1.0" encoding="utf-8"?>
<LMIDocument version="1.0">
<LocalModeStatus>
   <LocalModeActive>true</LocalModeActive>
   <LocalModeDisplay>StandAlone Mode</LocalModeDisplay>
</LocalModeStatus>
</LMIDocument>
```

32.10 Call Forking

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

32.10.1 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 270).

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.

32.10.2 SIP Forking Initiated by SIP Proxy Server

The device can handle SIP forking responses received from a proxy server in response to an INVITE forwarded by the device from a UA. In other words, received responses with a different SIP To header 'tag' parameter for the request forwarded by the device. This occurs in scenarios, for example, where a proxy server forks the INVITE request to several UAs, and therefore, the SBC device may receive several replies for a single request. Forked SIP responses may result in a single SDP offer with two or more SDP answers during call setup. The SBC handles this scenario by "hiding" the forked responses from the INVITE-initiating UA. This is achieved by marking the UA that responded first to the INVITE as the active UA, and only requests/responses from that UA are subsequently forwarded. All other requests/responses from other UAs are handled by the SBC (SDP offers from these users are answered with an 'inactive' media).

The SBC supports two forking modes, configured by the SBCForkingHandlingMode parameter:

- Latch On First - only the first received 18x response is forwarded to the INVITE initiating UA, and disregards any subsequently received 18x forking responses (with or without SDP).
- Sequential - all 18x responses are forwarded to the INVITE initiating UA, one at a time in a sequential manner. If 18x arrives with an offer only, only the first offer is forwarded to the INVITE initiating UA.

The SBC also supports media synchronization for call forking. If the active UA is the first one to send the final response (e.g., 200 OK) the call is established and all other final responses are acknowledged and a BYE is sent if needed. If another UA sends the first final response, then it is possible that the SDP answer that was forwarded to the INVITE-initiating UA is not relevant, and media synchronization is needed between the two UAs. Media synchronization is done by sending a re-INVITE request immediately after the call is established. The re-INVITE is sent without an offer to the INVITE-initiating UA. This causes the UA to send an offer which is forwarded to the UA that confirmed the call. The media synchronization process is enabled by the EnableSBCMediaSync parameter.

32.10.3 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 552.

32.11 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.

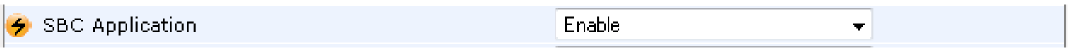
33 Enabling the SBC Application

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 621.

➢ **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 intentionaly left blank.

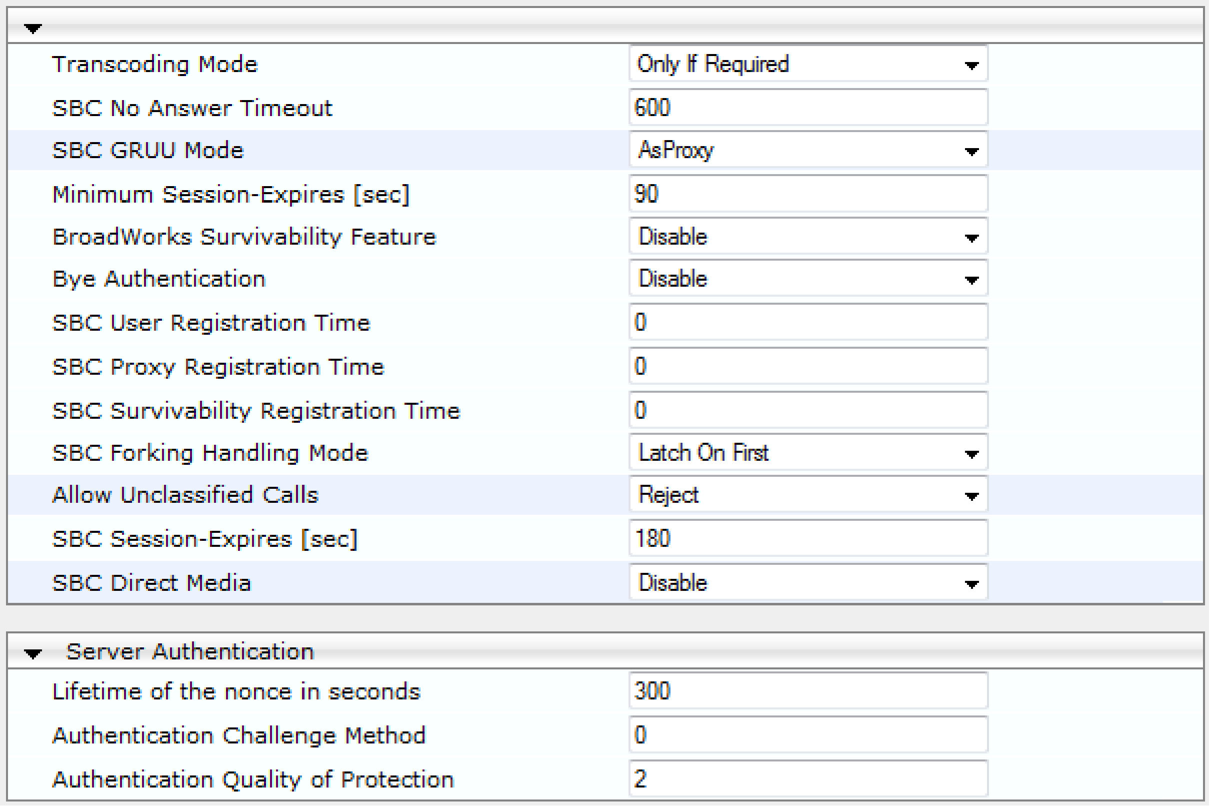
34 Configuring General Settings

The General Settings page allows you to configure general SBC parameters. For a description of these parameters, see "SBC Parameters" on page 983.

➢ **To configure general parameters:**

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

Figure 34-1: 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

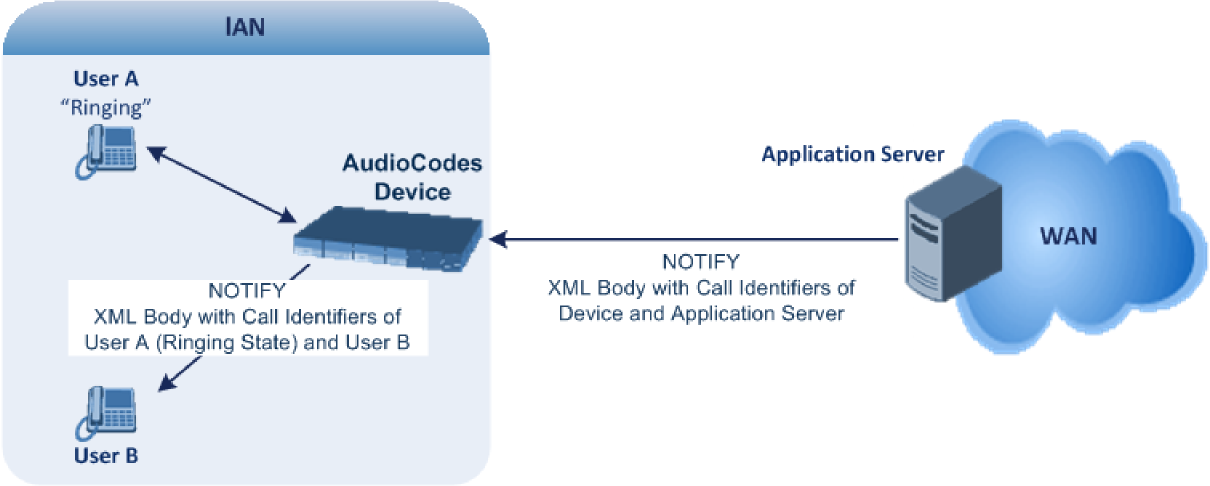
2. Configure the parameters as required.
3. Click **Submit**.
4. To save the changes to flash memory, see "Saving Configuration" on page 592.

34.1 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 34-2: Interworking NOTIFY XML Body for Application Server


IAN
User A
"Ringing"
AudioCodes Device
NOTIFY
XML Body with Call Identifiers of
User A (Ringing State) and User B
User B
Application Server
WAN
NOTIFY
XML Body with Call Identifiers of
Device and Application Server

To enable this feature, set the 'SBC Dialog-Info Interworking' (EnableSBCDialogInfoInterworking) parameter to **Enable**. When this feature is disabled, the device forwards the NOTIFY message as is, without modifying its XML body.

Below is an example of an XML body where the call-id, tags, and URIs have been replaced by the device:

```
<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
version="10" state="partial"
entity="sip:alice@example.com">
<dialog id="zxcvbnm3" call-id="67402270@10.132.10.150"
local-tag="1c137249965"
remote-tag="CCDORRTDRKIKWFVBRWYM" direction="initiator">
<state event="replaced">terminated</state>
</dialog>
<dialog id="sfhjsjk12" call-id="67402270@10.132.10.150"
local-tag="1c137249965"
remote-tag="CCDORRTDRKIKWFVBRWYM" direction="receiver">
<state reason="replaced">confirmed</state>
<replaces
call-id="67402270@10.132.10.150"
local-tag="1c137249965"
remote-tag="CCDORRTDRKIKWFVBRWYM"/>
<referred-by>
sip:bob-is-not-here@vm.example.net
</referred-by>
<local>
<identity display="Jason Forster">
sip:jforsters@home.net
</identity>
<target uri="sip:alice@pc33.example.com">
<param pname="+sip.rendering" pval="yes"/>
</target>
</local>
<remote>
<identity display="Cathy Jones">
sip:cjones@example.net
</identity>
```

```
<target uri="sip:line3@host3.example.net">
<param pname="actor" pval="attendant"/>
<param pname="automaton" pval="false"/>
</target>
</remote>
</dialog>
</dialog-info>
```

This page is intentionaly left blank.

35 Configuring Admission Control

The Admission Control table lets you configure up to 100 Call Admission Control rules (CAC). CAC rules define the maximum number of concurrent calls (SIP dialogs) permitted per IP Group or SRD, and per user (identified by its registered contact) belonging to these entities. CAC rules also define a guaranteed (*reserved*) number of concurrent calls. Thus, CAC rules can be useful for implementing Service Level Agreements (SLA) policies. CAC rules are also especially important for applications where VoIP and Data traffic contend on the WAN throughput, which may be limited by itself. For example, DSL WAN access interface is very limited in the uplink. By controlling the number of permitted calls, bandwidth can be reserved for specific Data applications.

CAC rules can be applied per SIP request type and SIP dialog direction (inbound and/or outbound). These relate to requests that initiate SIP dialogs and not the subsequent requests that can be of different type and direction. The SIP dialog-initiating request types can include INVITE, REGISTER, and/or SUBSCRIBE messages, or it can be configured to include the total number of all dialogs.

This feature also provides support for SIP-dialog rate control, using the “token bucket” mechanism. The token bucket is a control mechanism that dictates the rate of SIP-dialog setups based on the presence of tokens in the bucket – a logical container that holds aggregate SIP dialogs to be accepted or transmitted. Tokens in the bucket are removed ("cashed in") for the ability to setup a dialog. Thus, a flow can setup dialogs up to its peak burst rate if there are adequate tokens in the bucket and if the burst threshold is configured appropriately:

- Every SIP dialog setup request must attempt to take a token from the bucket.
- If there are no tokens, the request is dropped.
- New tokens are added to the bucket at a user-defined rate (token rate).
- If the bucket contains the maximum number of tokens, tokens to be added at that moment are dropped.

Reserved capacity is especially useful when the device operates with multiple SIP entities such as in a contact center environment handling multiple customers. For example, if the total call capacity of the device is 200 call sessions, a scenario may arise where one SIP entity may reach the maximum configured call capacity of 200 and thereby, leaving no available call resources for the other SIP entities. Thus, reserved capacity guarantees a minimum capacity for each SIP entity. If the reserved call capacity of a SIP entity is threatened by a new call for a different SIP entity, the device rejects the call to safeguard the reserved capacity.

Reserved call capacity can be configured for both an SRD and each of its associated IP Groups. In such a setup, the SRD's reserved call capacity must be greater or equal to the summation of the reserved call capacity of all these IP Groups. In other words, the SRD serves as the "parent" reserved call capacity. If the SRD's reserved call capacity is greater, the extra call capacity can be used as a shared pool between the IP Groups for unreserved calls when they exceed their reserved capacity. For example, assume that the reserved capacities for an SRD and its associated IP Groups are as follows:

- SRD reserved call capacity: 40
- IP Group ID 1 reserved call capacity: 10
- IP Group ID 2 reserved call capacity: 20

In this setup, the SRD offers a shared pool for unreserved call capacity of 10 [i.e., 40 – (10 + 20)]. If IP Group ID 1 needs to handle 15 calls, it is guaranteed 10 calls and the remaining 5 is provided from the SRD's shared pool. If the SDR's shared pool is currently empty and resources for new calls are required, the quota is taken from the device's total capacity, if available. For example, if IP Group ID 1 needs to handle 21 calls, it's guaranteed 10, the SRD's shared pool provides another 10, and the last call is provided from the device's total call capacity support (e.g., of 200).

Requests that reach the user-defined call limit (maximum concurrent calls and/or call rate) are sent to an alternative route, if configured in the IP-to-IP Routing table. If no alternative routing rule is located, the device rejects the SIP request with a SIP 486 "Busy Here" response.

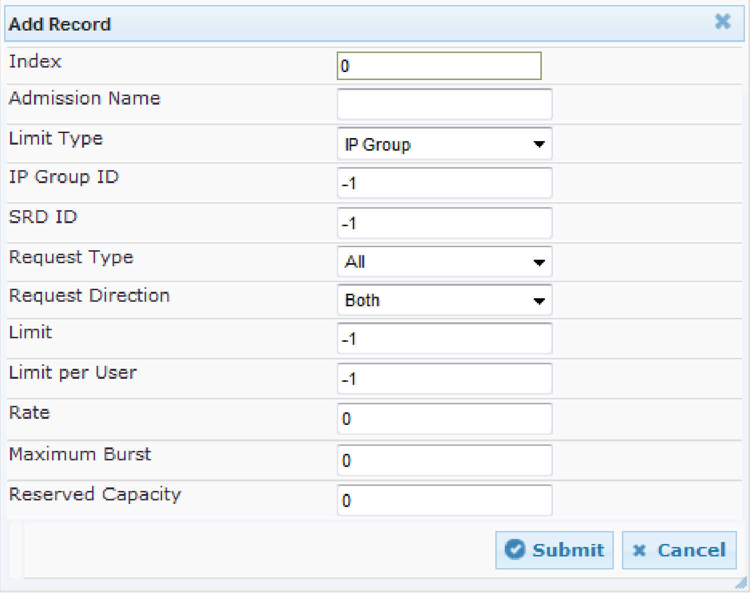
> **Note:** The device applies the CAC rule for the incoming leg immediately after the Classification process. If the call/request is rejected at this stage, no routing is performed. The enforcement for the outgoing leg is performed within each alternative route iteration. This is accessed from two places: one during initial classification/routing, and another during alternative routing process.

The following procedure describes how to configure CAC rules in the Web interface. You can also configure CAC rules using the table ini file parameter, SBCAdmissionControl or CLI command, configure voip > sbc sbc-admission-control.

➢ To configure a CAC rule:

1. Open the Admission Control page (**Configuration** tab > **VoIP** menu > **SBC** > **Admission Control**).
2. Click **Add**; the following dialog box appears:

Figure 35-1: Admission Control Page - Add Record Dialog Box



Add Record

Index	0
Admission Name	
Limit Type	IP Group
IP Group ID	-1
SRD ID	-1
Request Type	All
Request Direction	Both
Limit	-1
Limit per User	-1
Rate	0
Maximum Burst	0
Reserved Capacity	0

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

Table 35-1: Admission Control Table Parameter Description

Parameter	Description
Index **[SBCAdmissionControl_Index]**	Defines an index number for the new table record.
Admission Name CLI: admission-name **[SBCAdmissionControl_AdmissionControlName]**	Defines an arbitrary name to easily identify the Admission Control rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Limit Type	Defines the entity to which the rule applies.

Parameter	Description
CLI: limit-type **[SBCAdmissionControl_LimitType]**	- **[0]** IP Group (default) - **[1]** SRD
IP Group ID CLI: ip-group-id **[SBCAdmissionControl_IPGroupID]**	Defines the IP Group to which you want to apply the rule. The default value is -1 (i.e., all IP Groups). **Note:** This parameter is applicable only if 'Limit Type' is set to **IP Group**.
SRD ID CLI: srd-id **[SBCAdmissionControl_SRDID]**	Defines the SRD to which you want to apply the rule. The default value is -1 (i.e., all SRDs). **Note:** This parameter is applicable only if 'Limit Type' is set to **SRD**.
Request Type CLI: request-type **[SBCAdmissionControl_RequestType]**	Defines the SIP dialog-initiating request type to which you want to apply the rule (not the subsequent requests that can be of different type and direction). - **[0]** All = (Default) Includes the total number of all dialogs. - **[1]** INVITE - **[2]** SUBSCRIBE - **[3]** Other
Request Direction CLI: request-direction **[SBCAdmissionControl_RequestDirection]**	Defines the direction of the SIP request to which the rule applies. - **[0]** Both = (Default) Rule applies to inbound and outbound SIP dialogs. - **[1]** Inbound = Rule applies only to inbound SIP dialogs. - **[2]** Outbound = Rule applies only to outbound SIP dialogs.
Limit CLI: limit **[SBCAdmissionControl_Limit]**	Defines the maximum number of concurrent SIP dialogs per IP Group or SRD. You can also use the following special values: - **[0]** 0 = Block all these dialogs. - **[-1]** -1 = (Default) Unlimited.
Limit Per User CLI: limit-per-user **[SBCAdmissionControl_LimitPerUser]**	Defines the maximum number of concurrent SIP dialogs per user belonging to the specified IP Group or SRD. You can also use the following special values: - **[0]** 0 = Block all these dialogs. - **[-1]** -1 = (Default) Unlimited.

Parameter	Description
Rate CLI: rate **[SBCAdmissionControl_Rate]**	Defines the rate at which tokens are added to the token bucket per second (i.e., token rate). One token is added to the bucket every 1000 divided by the value of this parameter (in milliseconds). The default is 0 (i.e., unlimited rate). **Note:** The token bucket feature is per IP Group, SRD, SIP request type, and SIP request direction.
Maximum Burst CLI: max-burst **[SBCAdmissionControl_MaxBurst]**	Defines the maximum number of tokens (SIP dialogs) that the bucket can hold. The device only accepts a SIP dialog if a token exists in the bucket. Once the SIP dialog is accepted, a token is removed from the bucket. If a SIP dialog is received by the device and the token bucket is empty, then the device rejects the SIP dialog. Alternatively, if the bucket is full, for example, 100 tokens, and 101 SIP dialogs arrive (before another token is added to the bucket, i.e., faster than that defined in the Rate field), then the device accepts the first 100 SIP dialogs and rejects the last one. Dropped requests are replied with the SIP 486 “Busy Here” response. Dropped requests are not counted in the bucket. The default is 0 (i.e., unlimited SIP dialogs). **Note:** The token bucket feature is per IP Group, SRD, SIP request type, and SIP request direction.
Reservation CLI: reservation **[SBCAdmissionControl_Reservation]**	Defines the guaranteed (minimum) call capacity. **Notes:** ▪ An IP Group ID or SRD ID must be specified when this parameter is configured and the IP Group or SRD cannot be set to all (-1). ▪ Reserved call capacity is applicable only to INVITE and SUBSCRIBE messages. ▪ Reserved call capacity must be less than the maximum capacity (limit) configured for the CAC rule. ▪ The total reserved call capacity configured for all the CAC rules must be within the device's total call capacity support.

36 Configuring Coder Groups

36.1 Configuring Allowed Audio Coder Groups

The Allowed Audio Coders Group table lets you configure up to five Allowed Audio Coders Groups. An Allowed Audio Coders Group defines a list of audio media coders that can be used for a specific SIP entity. Each Allowed Audio Coders Group can be configured with up to 10 coders. The coders can include pre-defined audio coders (according to the installed Software License Key) and user-defined (string) coders for non-standard or unknown coders.

Allowed Audio Coders Groups are assigned to SIP entities, using IP Profiles (see "Configuring IP Profiles" on page 314). Coders that are not listed in the Allowed Audio Coders Group are removed from the SDP offer ('a=rtpmap' field) that is sent to the SIP entity. Only coders that are common between the coders in the SDP offer and the coders listed in the Allowed Audio Coders Group are used. Thus, Allowed Audio Coders Groups enable you to enforce the use of only specified coders. For more information, see "Restricting Coders" on page 513.

You can also use Allowed Audio Coders Groups to configure Extension coders. Extension coders are coders (listed in the Allowed Audio Coders Group) which the device adds to the SDP offer. For more information on Extension coders, see Coder Transcoding on page 513.

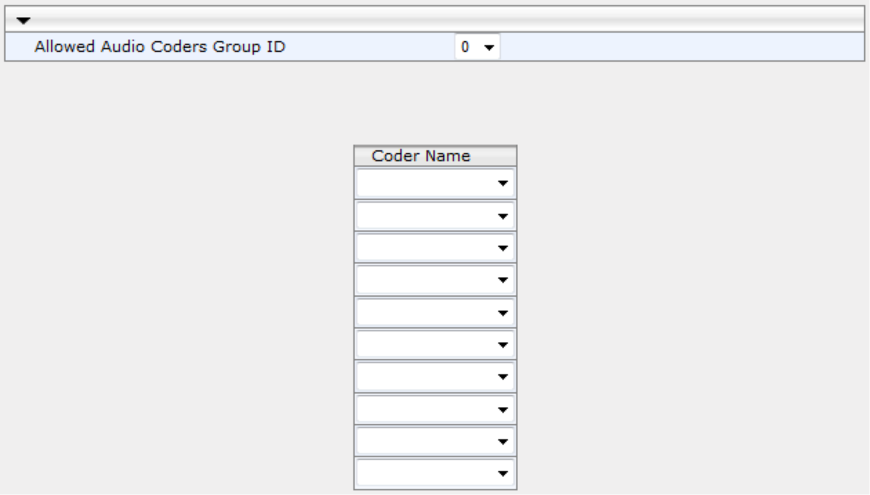
The order of appearance of the coders listed in the Allowed Audio Coders Group determines the priority (preference) of the coders in the SDP offer. The device arranges the SDP offer's coder list according to their order in the Allowed Audio Coders Group. The priority is in descending order, whereby the first coder in the list is given the highest priority and the last coder, the lowest priority. If Extension coders are also used, these coders can be added after the Allowed coders or arranged with the Allowed coders according to their appearance in the Allowed Coders Group. For more information, see "Prioritizing Coder List in SDP Offer" on page 515.

The following procedure describes how to configure Allowed Audio Coder Groups in the Web interface. You can also configure Allowed Audio Coder Groups using the table ini file parameter, AllowedCodersGroup or CLI command, configure voip > sbc allowed-coders-group group-0.

➢ To configure an Allowed Coders Group:

1. Open the Allowed Audio Coders Group page (**Configuration** tab > **VoIP** menu > **SBC** > **Allowed Audio Coders Group**).

Figure 36-1: Allowed Audio Coders Group Page



Allowed Audio Coders Group ID 0

2. Configure an Allowed Audio Coders Group according to the parameters described in the table below.
3. Click **Submit**, and then reset the device with a save ("burn") to flash memory.

Table 36-1: Allowed Audio Coders Group Table Parameter Descriptions

Parameter	Description
Allowed Coders Group ID **[AllowedCodersGroupX]**	Defines an index number for the new table record.
Coder Name CLI: name **[AllowedCodersGroupX_Name]**	Defines the audio coder. This can be a pre-defined coder or a user-defined coder. The valid value for user-defined coders is a string of up to 25 characters (case-insensitive). For example, "HD.123" (without quotes). **Note:** Each coder type (e.g., G.729) can be configured only once per Allowed Coders Group.

36.2 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 pre-defined video coders (according to the installed Software License Key) and user-defined (string) video coders for non-standard or unknown coders. Allowed Video Coders Groups are assigned to SIP entities, using IP Profiles (see "Configuring IP Profiles" on page 314). The video coders appear in the SDP media type "video" ('m=video' line). Coders that are not listed in the Allowed Video Coders Group are removed from the SDP offer that is sent to the SIP entity. Only coders that are common between the coders in the SDP offer and the coders listed in the Allowed Video Coders Group are used. Thus, Allowed Video Coders Groups enable you to enforce the use of only specified coders. For more information, see "Restricting Coders" on page 513.

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 515.

Currently, the Allowed Video Coder Groups table can only be configured using the ini file parameter, AllowedVideoCodersGroup or CLI command, configure voip/sbc allowed-video-coders-group group-0. The table below describes this parameter.

Table 36-2: Allowed Video Coders Group Table Parameter Descriptions

Parameter	Description
Allowed Coders Group ID **[AllowedVideoCodersGroupX]**	Defines an index number for the new table record.
Coder Name CLI: name **[AllowedVideoCodersGroupX_Name]**	Defines the video coder. This can be a pre-configured coder or a user-defined coder. The valid value for user-defined coders is a string of up to 25 characters (case-insensitive). For example, "WOW.789" (but without quotes). **Note:** Each coder type can be configured only once per Allowed Video Coders Group.

37 Routing SBC

This section describes the configuration of the routing entities for the SBC application. These include the following:

- Classification rules - see "Configuring Classification Rules" on page 545
- Message Condition rules - see "Configuring Message Condition Rules" on page 551
- IP-to-IP routing rules - see "Configuring SBC IP-to-IP Routing Rules" on page 552
- Alternative routing reasons - see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 561

37.1 Configuring Classification Rules

The Classification table lets you configure up to 100 Classification rules. A Classification rule classifies incoming SIP dialog-initiating requests (e.g., INVITE messages) to an IP Group from where the SIP dialog request was received. The identified IP Group is then used in the manipulation and routing processes. Classification rules also enhance security by allowing you to create a SIP access list, whereby classified calls can be denied (i.e., blacklist) or allowed (i.e., whitelist).

A Classification rule includes two areas:

- Matching characteristics (**Rule**) - characteristics of incoming IP call such as source IP address.
- Operation (**Action**) - if the incoming call matches the characteristics of the rule, the device classifies the call as belonging to an IP Group.

The Classification table is used to classify incoming SIP dialog requests only if the following classification stages fail:

1. **Classification Stage 1 - Registered Users Database:** The device searches its registration database to check if the incoming SIP dialog arrived from a registered user:
   - Compares the SIP Contact header of the received SIP dialog to the Contact of the registered user.
   - Compares the URL in the SIP P-Asserted-Identity/From header to the registered address-of-record (AOR).

   If this stage fails, the device proceeds to classification based on Proxy Set.
2. **Classification Stage 2 - Proxy Set:** If the database search fails, the device performs classification based on Proxy Set. This is done only if the 'Classify By Proxy Set' parameter is enabled for the IP Group (see "Configuring IP Groups" on page 270). The device checks whether the INVITE's IP address (if host name, then according to the dynamically resolved IP address list) is defined for a Proxy Set ID (in the Proxy Set table). If a Proxy Set ID has such an IP address, the device classifies the INVITE to the IP Group that is associated with this Proxy Set. (The Proxy Set ID is assigned to the IP Group using the IP Group table's 'Proxy Set ID' parameter.)

   > **Note:** For security, it is highly recommended to disable the Classify by Proxy Set feature so that the device can use the Classification table instead. This enables "strict" classification of incoming calls to IP Groups. In addition, in cases where multiple IP Groups are associated with the same Proxy Set ID, do **not** use the Classify by Proxy Set feature.

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

3. **Classification Stage 3 - Classification Table:** If classification based on Proxy Set fails (or disabled), the device uses the Classification table to classify the SIP dialog to an IP Group. If it locates a Classification rule whose characteristics (such as source IP address) match the incoming SIP dialog, the SIP dialog is assigned to the associated IP Group. In addition, if the Classification rule is defined as a whitelist, the SIP dialog is allowed and proceeds with the manipulation, routing and other SBC processes. If the Classification rule is defined as a blacklist, the SIP dialog is denied.

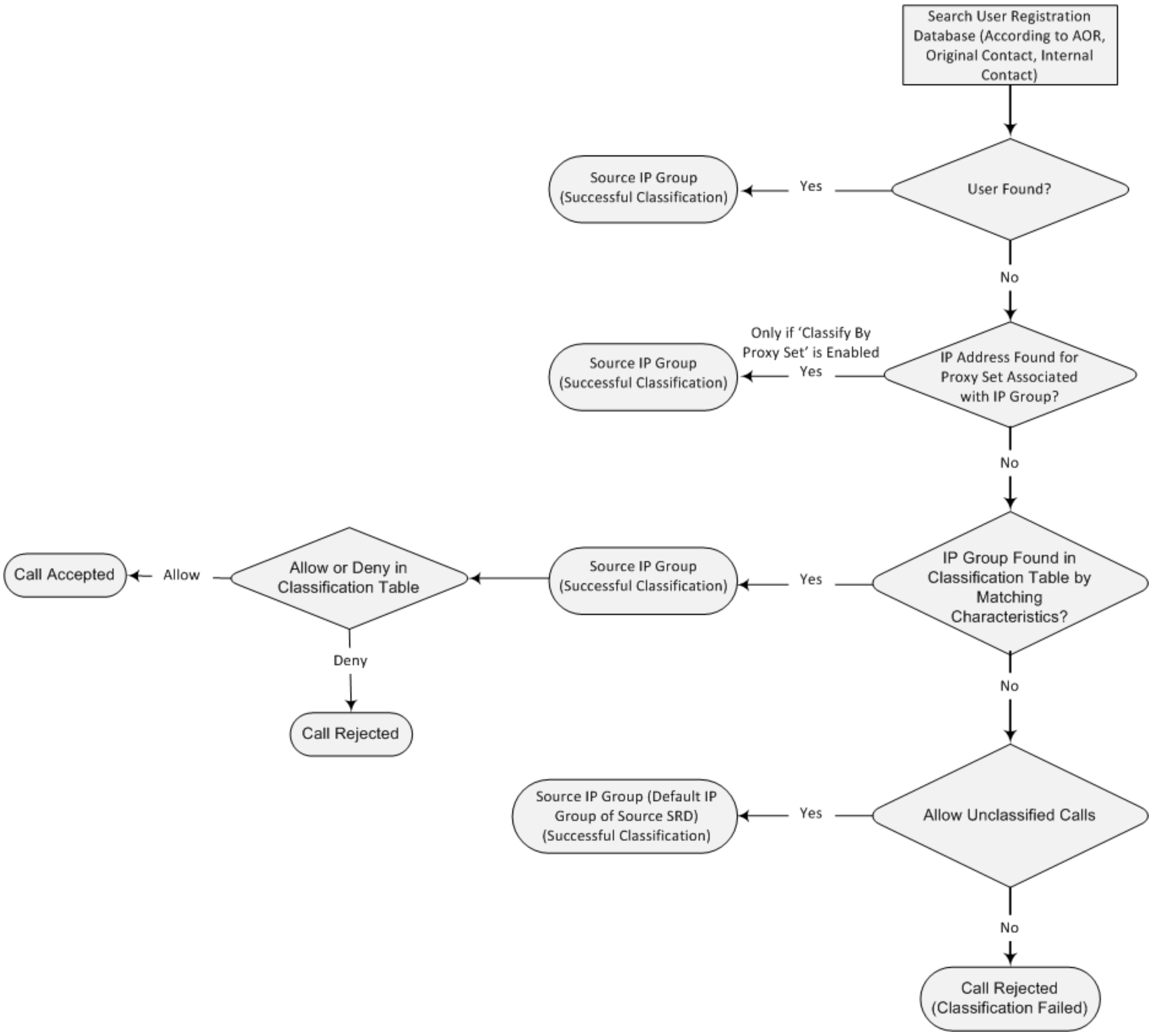
If the classification process fails, the device rejects or allows the call, depending on the setting of the 'Unclassified Calls' parameter (on the General Settings page - **Configuration** tab > **VoIP** menu > **SBC** > **General Settings**). If this parameter is set to **Allow**, the incoming SIP dialog is assigned to an IP Group as follows:

1. The device checks on which SIP listening port (e.g., 5061) the incoming SIP dialog request arrived and the SIP Interface which is configured with this port (in the SIP Interface table).
2. The device checks the SRD that is associated with this SIP Interface (in the SIP Interface table) and then classifies the SIP dialog with the first IP Group that is associated with this SRD. For example, if IP Groups 3 and 4 use the same SRD, the device classifies the call to IP Group 3.

Note: If classification for a SIP request fails and the device is configured to reject unclassified calls, the device can send a specific SIP response code per SIP interface. This is configured by the 'Classification Failure Response Type' parameter in the SIP Interface table (see "Configuring SIP Interfaces" on page 267).

The flowchart below illustrates the classification process:

Figure 37-1: Classification Process (Identifying IP Group or Rejecting Call)


Search User Registration Database (According to AOR, Original Contact, Internal Contact)
User Found?
Yes
Source IP Group (Successful Classification)
No
IP Address Found for Proxy Set Associated with IP Group?
Only if 'Classify By Proxy Set' is Enabled
Yes
Source IP Group (Successful Classification)
No
IP Group Found in Classification Table by Matching Characteristics?
Yes
Source IP Group (Successful Classification)
Allow or Deny in Classification Table
Allow
Call Accepted
Deny
Call Rejected
No
Allow Unclassified Calls
Yes
Source IP Group (Default IP Group of Source SRD) (Successful Classification)
No
Call Rejected (Classification Failed)

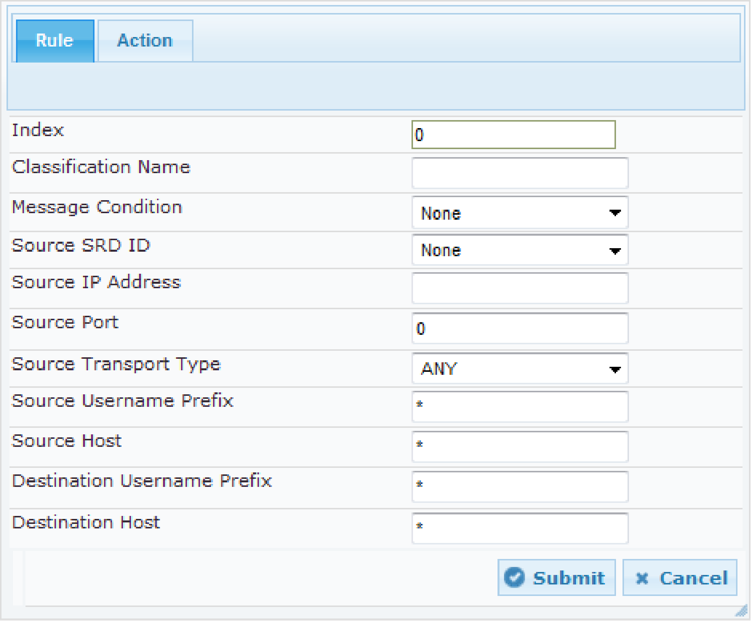
Note: The device saves incoming SIP REGISTER messages in its registration database. If the REGISTER message is received from a User-type IP Group, the device sends the message to the configured destination.

The following procedure describes how to configure Classification rules in the Web interface. You can also configure Classification rules using the table ini file parameter, Classification or CLI command, configure voip > sbc routing classification.

➢ To configure a Classification rule:

1. Open the Classification Table page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Classification Table**).
2. Click **Add**; the following dialog box appears:

Figure 37-2: Classification Table Page



Index	0
Classification Name	
Message Condition	None
Source SRD ID	None
Source IP Address	
Source Port	0
Source Transport Type	ANY
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*

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

Table 37-1: Classification Table Parameter Descriptions

Parameter	Description
Index **[Classification_Index]**	Defines an index number for the new table record.
Classification Name CLI: classification-name **[Classification_ClassificationName]**	Defines an arbitrary name to easily identify the Classification rule. The valid value is a string of up to 20 characters. By default, no name is defined.
Matching Characteristics - Rule	
Message Condition CLI: message-condition **[Classification_MessageCondition]**	Assigns a Message Condition rule, which can be used to classify the incoming SIP dialog. To configure Condition rules, see "Configuring Message Condition Rules" on page 551.
Source SRD ID CLI: src-srd-id **[Classification_SrcSRDID]**	Defines an SRD ID of the incoming SIP dialog. To configure SRDs, see "Configuring SRDs" on page 264. By default, no SRD is defined. **Note:** The SRDs are also associated with a port number as defined by the SIP Interface used by the SRD (see "Configuring SIP Interfaces" on page 267).
Source IP Address CLI: src-ip-address **[Classification_SrcAddress]**	Defines the source IP address (in dotted-decimal notation) of the incoming SIP dialog. The IP address can be configured using the following

Parameter	Description
	wildcards: ▪ "x" wildcard: represents single digits. For example, 10.8.8.xx represents all the addresses between 10.8.8.10 to 10.8.8.99. ▪ Asterisk (*) wildcard: represents any number between 0 and 255. For example, 10.8.8.* represents all addresses between 10.8.8.0 and 10.8.8.255. If this parameter is not configured or is configured as an asterisk (*), any source IP address is accepted.
Source Port CLI: src-port **[Classification_SrcPort]**	Defines the source port number of the incoming SIP dialog.
Source Transport Type CLI: src-transport-type **[Classification_SrcTransportType]**	Defines the source transport type (UDP, TCP, or TLS) of the incoming SIP dialog. ▪ **[-1]** ANY (Default) = All transport types ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS
Source Username Prefix CLI: src-user-name-prefix **[Classification_SrcUsernamePrefix]**	Defines the prefix of the source URI user part of the incoming SIP dialog. The URI is typically located in the SIP From header. However, you can configure the SIP header from where the device obtains the source URI. This is done in the IP Group table, using the 'Source URI Input' parameter. For more information on how the device obtains this URI, see "SIP Dialog Initiation Process" on page 502. 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 761. **Note:** For REGISTER requests, the source URL is obtained from the To header.
Source Host CLI: src-host **[Classification_SrcHost]**	Defines the prefix of the source URI host name. The URI is typically located in the SIP From header. However, you can configure the SIP header from where the device obtains the source URI. This is done in the IP Group table, using the 'Source URI Input' parameter. For more information on how the device obtains this URI, see "SIP Dialog Initiation Process" on page 502. The default is the asterisk (*) symbol, which represents any source host prefix. **Note:** For REGISTER requests, the source URL is obtained from the To header.
Destination Username Prefix CLI: dst-user-name-prefix **[Classification_DestUsernamePrefix]**	Defines the prefix of the destination Request-URI user part of the incoming SIP dialog. The default is the asterisk (*) symbol, which represents any destination username. The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 761.

Parameter	Description
Destination Host CLI: dst-host **[Classification_DestHost]**	Defines the prefix of the destination Request-URI host name of the incoming SIP dialog request. The default is the asterisk (*) symbol, which represents any destination host prefix.
Operation Rule - Action	
Action Type CLI: action-type **[Classification_ActionType]**	Defines a whitelist or blacklist for incoming SIP dialog requests that match the characteristics of the classification rule. ▪ **[0]** Deny = Blocks incoming SIP dialogs that match the characteristics of the Classification rule (blacklist). ▪ **[1]** Allow = (Default) Allows incoming SIP dialogs that match the characteristics of the Classification rule (whitelist) and assigns it to the associated IP Group. (default)
Source IP Group ID CLI: src-ip-group-id **[Classification_SrcIPGroupID]**	Defines an IP Group to which the incoming SIP dialog request must be assigned if this SIP dialog matches the matching characteristics. The IP Group is used for SBC routing and manipulations. To configure IP Groups, see "Configuring IP Groups" on page 270. By default, no IP Group is defined. **Note:** The IP Group must be associated with the assigned SRD.

37.1.1 Classification Based on URI of Selected Header Example

The following example describes how to configure classification of incoming calls to IP Groups, based on source URI in a specific SIP header.

This example assumes the following incoming INVITE message:

```
INVITE sip:8000@10.33.4.226 SIP/2.0
Via: SIP/2.0/UDP 10.33.4.226;branch=z9hG4bKVEBTDAHSUYRTEXEDEGJY
From: <sip:100@10.33.4.226>;tag=YSQQKXXREVDPYPTNFMWG
To: <sip:8000@10.33.4.226>
Call-ID: FKPNOYRNKROIMEGBSSKS@10.33.4.226
CSeq: 1 INVITE
Contact: <sip:100@10.33.4.226>
Route: <sip:2000@10.10.10.10.10>,<sip:300@10.10.10.30>
Supported: em,100rel,timer,replaces
P-Called-Party-ID: <sip:1111@10.33.38.1>
User-Agent: Sip Message Generator V1.0.0.5
Content-Length: 0
```

1. In the Classification table, add the following classification rules:

Index	Source Username Prefix	Destination Username Prefix	Destination Host	Source IP Group ID
0	333	-	-	1
1	1111	2000	10.10.10.10	2

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

37.2 Configuring Message Condition Rules

The Message Condition table lets you configure up to 20 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 545)
- IP-to-IP routing rules in the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 552)
- IP-to-IP outbound manipulation rules in the IP to IP Outbound Manipulation table (see "Configuring IP-to-IP Outbound Manipulations" on page 568)

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 299). You can configure simple Message Condition rules, for example, "header.to.host contains company", meaning SIP messages whose To header has a host part containing the string "company". You can configure complex rules using the "AND" or "OR" Boolean operands and also use regular expressions (regex), for example:

- "body.sdp regex pcmu" can be used to enable routing based on the offered codec (G.711 Mu) in the incoming SDP message.
- "body.sdp regex (AVP[0-9||\s]*\s8[\s||\n])" can be used to enable routing based on payload type 8 in the incoming SDP message.

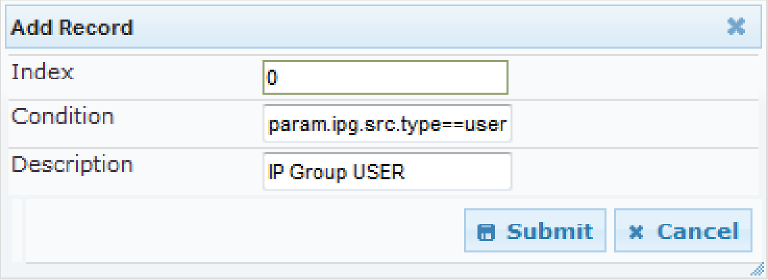
Note: For a description on SIP message manipulation syntax, refer to the *SIP Message Manipulations Quick Reference Guide.*

The following procedure describes how to configure Message Condition rules in the Web interface. You can also configure Message Condition rules using the table ini file parameter, ConditionTable or CLI command, configure voip > sbc routing condition-table.

➢ To configure a Message Condition rule:

1. Open the Message Condition Table page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Message Condition Table**).
2. Click **Add**; the following dialog box appears:

Figure 37-3: Condition Table Page - Add Record Dialog Box



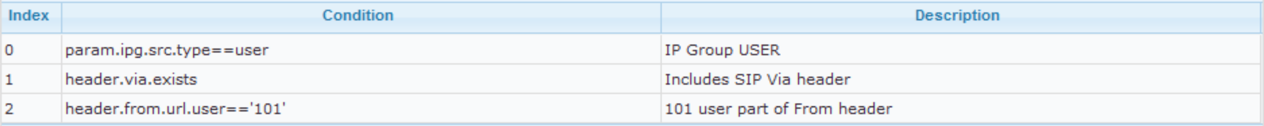
Add Record

Index	0
Condition	param.ipg.src.type==user
Description	IP Group USER

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

An example of configured Message Condition rules is shown in the figure below:

Figure 37-4: Condition Table Page



Index	Condition	Description
0	param.ipg.src.type==user	IP Group USER
1	header.via.exists	Includes SIP Via header
2	header.from.url.user=='101'	101 user part of From header

- **Index 0:** Incoming SIP dialog that is classified as belonging to a User-type IP Group.
- **Index 1:** Incoming SIP dialog that contains a SIP Via header.
- **Index 2:** Incoming SIP dialog with 101 as the user part in the SIP From header.

Table 37-2: Message Condition Table Parameter Descriptions

Parameter	Description
Index **[ConditionTable_Index]**	Defines an index number for the new table record.
Condition CLI: condition **[ConditionTable_Condition]**	Defines the Condition rule of the SIP message. The valid value is a string. **Note:** User and host parts must be enclosed in single quotes.
Description CLI: description **[ConditionTable_Description]**	Defines a brief description of the Condition rule.

37.3 Configuring SBC IP-to-IP Routing

The IP-to-IP Routing table lets you configure up to 500 SBC IP-to-IP routing rules. An IP-to-IP routing rule routes received SIP dialog messages (e.g., INVITE) to an IP destination. An IP-to-IP routing rule includes two areas:

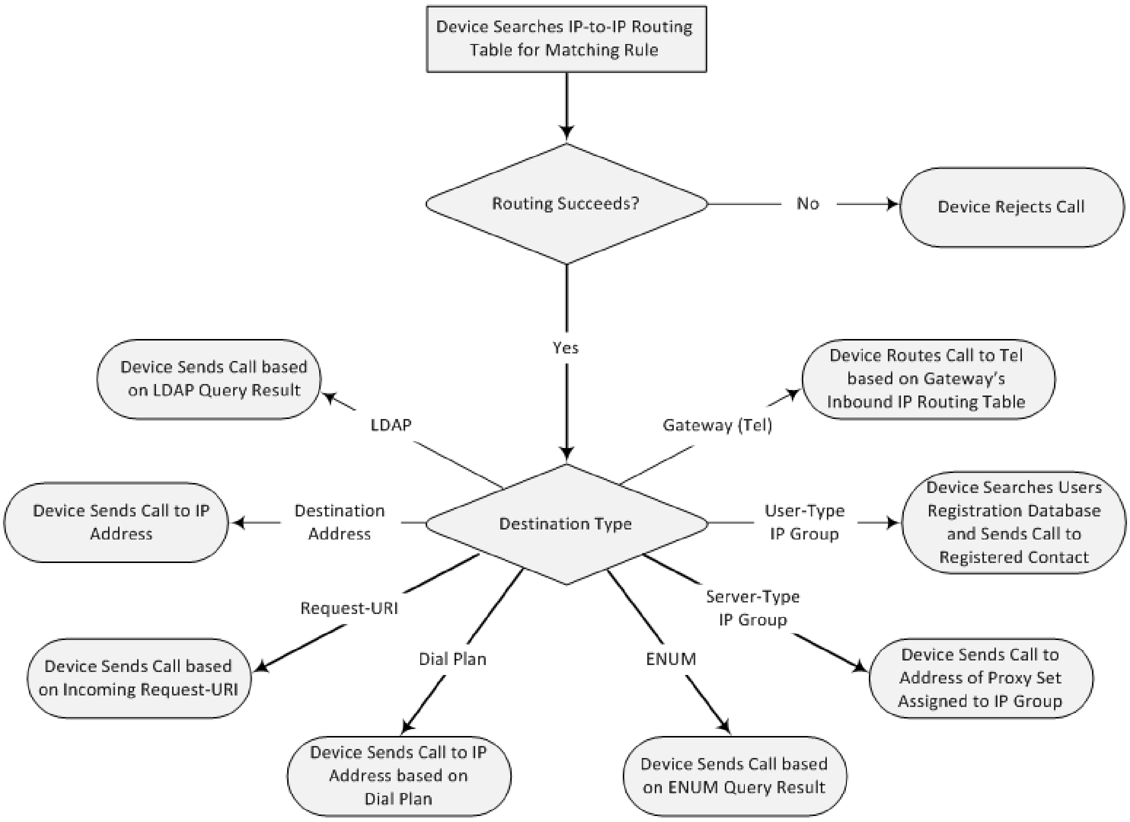
- Matching characteristics (Rule) - characteristics of incoming SIP message such as the IP Group from where the message was received.
- Operation (Action) - if the incoming SIP message matches the characteristics of the rule, the device routes the message to the configured destination

If the characteristics of an incoming call do not match the first rule in the table, the call characteristics are compared to those of the second rule, and so on until a matching rule is located. If no rule is matched, the call is rejected.

You can configure the IP-to-IP routing rule to send the call to any of the following IP destinations:

- According to registered user Contact listed in the device's database (only for User-type IP Groups).
- IP Group - the destination is the address configured for the Proxy Set associated with the IP Group (allows redundancy/load balancing).
- IP address in dotted-decimal notation or FQDN. Routing to a host name can be resolved using NAPTR/SRV/A-Record.
- Request-URI of incoming SIP dialog initiating requests.
- According to result of an ENUM query.
- Hunt Group - used for call survivability of call centers (see "Call Survivability for Call Centers" on page 528).
- IP address according to a specified Dial Plan index listed in the loaded Dial Plan file.
- According to result of LDAP query (for more information on LDAP-based routing, see "Routing Based on LDAP Active Directory Queries" on page 210).

The IP-to-IP routing rule can also send the IP call to the Tel side (i.e., Gateway call). The rule redirects the call to the Inbound IP Routing table where the device searches for a matching IP-to-Tel routing rule. This feature can also be done for alternative routing. If an IP-to-IP routing rule fails and it is configured with a "Gateway" routing rule as an alternative route, the device uses the Inbound IP Routing table to send the call to the Tel. The device identifies (internally) calls re-directed for alternative Gateway routing, by appending a user-defined string to the prefix destination Request-URI user part (by default, "acgateway-<prefix destination>", for example, acgateway-200). The device removes this prefix before sending it to the Tel side. To configure this prefix string, use the GWDirectRoutePrefix ini file parameter.


Device Searches IP-to-IP Routing Table for Matching Rule
Routing Succeeds?
No
Device Rejects Call
Yes
Device Sends Call based on LDAP Query Result
LDAP
Gateway (Tel)
Device Routes Call to Tel based on Gateway's Inbound IP Routing Table
Device Sends Call to IP Address
Destination Address
Destination Type
User-Type IP Group
Device Searches Users Registration Database and Sends Call to Registered Contact
Request-URI
Server-Type IP Group
Device Sends Call based on Incoming Request-URI
Dial Plan
ENUM
Device Sends Call to Address of Proxy Set Assigned to IP Group
Device Sends Call to IP Address based on Dial Plan
Device Sends Call based on ENUM Query Result

The IP-to-IP Routing table also provides the following features:

- **Alternative routing or load balancing:** In addition to the alternative routing/load balancing provided by the Proxy Set associated with the destination IP Group, the table allows the configuration of alternative routes whereby if a route fails, the next adjacent (below) rule in the table that is configured as 'Alt Route Ignore/Consider Inputs' are used. The alternative routes rules can be set to enforce the input matching criteria or to ignore any matching criteria. Alternative routing occurs upon one of the following conditions:
  - A request sent by the device is responded with one of the following:
    - SIP response code (i.e., 4xx, 5xx, and 6xx SIP responses) configured in the SBC Alternative Routing Reasons table (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 561).
    - SIP 408 Timeout or no response (after timeout).
  - The DNS resolution includes IP addresses that the device has yet to try (for the current call).

  Messages are re-routed with the same SIP Call-ID and CSeq header fields (increased by 1).

- **Re-routing of SIP requests:** This table enables you to configure "re-routing" rules of requests (e.g., INVITEs) that the device sends upon receipt of SIP 3xx responses or REFER messages. These rules are configured for destinations that do not support receipt of 3xx or REFER and where the device handles the requests locally (instead of forwarding the 3xx or REFER to the destination).
- **Least cost routing (LCR):** If the LCR feature is enabled, the device searches the routing table for matching routing rules and then selects the one with the lowest call cost. The call cost of the routing rule is done by assigning it a Cost Group. For configuring Cost Groups, see "Least Cost Routing" on page 234. If two routing rules have identical costs, then the rule appearing higher up in the table (i.e., first-matched rule) is used. If a selected route is unavailable, the device uses the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched routing rules with Cost Groups, according to the settings of the LCR parameter, LCRDefaultCost (see "Enabling LCR and Configuring Default LCR" on page 236).
- **Call Forking:** The IP-to-IP Routing table can be configured to route an incoming IP call to multiple destinations (call forking). The incoming call can be routed to multiple destinations of any type such as an IP Group or IP address. The device forks the call by sending simultaneous INVITE messages to all the specified destinations. It handles the multiple SIP dialogs until one of the calls is answered and then terminates the other SIP dialogs.

  Call forking is configured by creating a Forking group. A Forking group consists of a main routing rule ('Alternative Route Options' set to **Route Row**) whose 'Group Policy' is set to **Forking**, and one or more associated routing rules ('Alternative Route Options' set to **Group Member Ignore Inputs** or **Group Member Consider Inputs**). The group members must be configured in contiguous table rows to the main routing rule. If an incoming call matches the input characteristics of the main routing rule, the device routes the call to its destination and all those of the group members.

  An alternative routing rule can also be configured for the Forking group. The alternative route is used if the call fails for the Forking group (i.e., main route and all its group members). The alternative routing rule must be configured in the table row immediately below the last member of the Forking group. The 'Alternative Route Options' of this alternative route must be set to **Alt Route Ignore Inputs** or **Alt Route Consider Inputs**. The alternative route can also be configured with its own forking group members, where if the device uses the alternative route, the call is also sent to its group members. In this case, instead of setting the alternative route's 'Group Policy' to **None**, you must set it to **Forking**. The group members of the alternative route must be configured in the rows immediately below it.

The LCR feature can also be employed with call forking. The device calculates a maximum call cost for each Forking group and routes the call to the Forking group with the lowest cost. Thus, even if the call can successfully be routed to the main routing rule, a different routing rule can be chosen (even an alternative route, if configured) based on LCR. If routing to one Forking group fails, the device tries to route the call to the Forking group with the next lowest cost (main or alternative route), and so on. The prerequisite for this functionality is that the incoming call must successfully match the input characteristics of the main routing rule.

- **Dial Plan Prefix Tags for Representing Source / Destination Numbers:** If your deployment includes calls of many different called (source URI user name) and/or calling (destination URI user name) numbers that need to be routed to the same destination, you can employ user-defined prefix tags to represent these numbers. Thus, instead of configuring many routing rules, you need to configure only one routing rule using the prefix tag as the source and destination number matching characteristics, and a destination for the calls. For more information on prefix tags, see "Dial Plan Prefix Tags for SBC IP-to-IP Routing" on page 610.

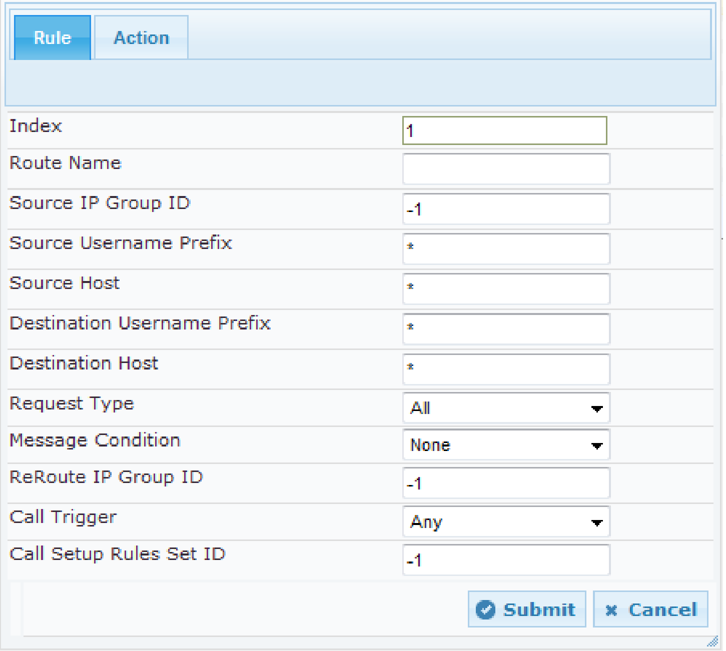
> **Note:** Call forking is not applicable to LDAP-based IP-to-IP routing rules.

The following procedure describes how to configure IP-to-IP routing rules in the Web interface. You can also configure IP-to-IP routing rules using the table ini file parameter, IP2IPRouting or CLI command, configure voip > sbc routing ip2ip-routing.

➢ To configure an IP-to-IP routing rule:

1. Open the IP-to-IP Routing Table page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **IP-to-IP Routing Table**).
2. Click **Add**; the following dialog box appears:

Figure 37-5: IP-to-IP Routing Table - Add Record Dialog Box



Index	1
Route Name	
Source IP Group ID	-1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Request Type	All
Message Condition	None
ReRoute IP Group ID	-1
Call Trigger	Any
Call Setup Rules Set ID	-1

3. Configure an IP-to-IP routing rule according to the parameters described in the table below.

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

Table 37-3: IP-to-IP Routing Table Parameter Descriptions

Parameter	Description
Index **[IP2IPRouting_Index]**	Defines an index number for the new table record.
Route Name CLI: route-name **[IP2IPRouting_RouteName]**	Defines an arbitrary name to easily identify the IP-to-IP routing rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Characteristics - Rule	
Source IP Group ID [IP2IPRouting_SrcIPGroupID] CLI: src-ip-group-id	Defines the IP Group from where the IP call was received. Typically, the IP Group of an incoming SIP dialog is determined (or classified) using the Classification table (see Configuring Classification Rules on page 545). The default is -1. To denote any IP Group, leave this field empty.
Source Username Prefix **[IP2IPRouting_SrcUsernamePrefix]** CLI: src-user-name-prefix	Defines the prefix of the user part of the incoming SIP dialog's source URI (usually the From URI). You can use special notations for denoting the prefix. To denote calls without a user part in the URI, use the $ sign. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 761. The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty.
Source Host **[IP2IPRouting_SrcHost]** CLI: src-host	Defines the host part of the incoming SIP dialog's source URI (usually the From URI). The default is the asterisk (*) symbol (i.e., any host name). If this rule is not required, leave this field empty.
Destination Username Prefix **[IP2IPRouting_DestUsernamePrefix]** CLI: dst-user-name-prefix	Defines the prefix of the incoming SIP dialog's destination URI (usually the Request URI) user part. You can use special notations for denoting the prefix. To denote calls without a user part in the URI, use the $ sign. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 761. The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty.
Destination Host **[IP2IPRouting_DestHost]** CLI: dst-host	Defines the host part of the incoming SIP dialog's destination URI (usually the Request-URI). The default is the asterisk (*) symbol (i.e., any destination host). If this rule is not required, leave this field empty.
Request Type [IP2IPRouting_RequestType] CLI: request-type	Defines the SIP dialog request type of the incoming SIP dialog. ▪ [0] All (default) ▪ [1] INVITE ▪ [2] REGISTER ▪ [3] SUBSCRIBE ▪ [4] INVITE and REGISTER ▪ [5] INVITE and SUBSCRIBE ▪ [6] OPTIONS

Parameter	Description
Message Condition **[IP2IPRouting_MessageCondition]** CLI: message-condition	Assigns a SIP message Condition rule. To configure Condition rules, see "Configuring Message Condition Rules" on page 551.
ReRoute IP Group ID **[IP2IPRouting_ReRouteIPGroupID]** CLI: re-route-ip-group-id	Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. This field is typically used for re-routing requests (e.g., INVITEs) when interworking is required for SIP 3xx redirect responses or REFER messages. For more information, see "Interworking SIP 3xx Redirect Responses" on page 519 and "Interworking SIP REFER Messages" on page 521, respectively. This parameter functions together with the 'Call Trigger' field (see below). The default is -1 (i.e., not configured).
Call Trigger **[IP2IPRouting_Trigger]** CLI: trigger	Defines the reason (i.e., trigger) for re-routing the SIP request: ▪ **[0]** Any = (Default) This routing rule is used for all scenarios (re-routes and non-re-routes). ▪ **[1]** 3xx = Re-routes the request if it was triggered as a result of a SIP 3xx response. ▪ **[2]** REFER = Re-routes the INVITE if it was triggered as a result of a REFER request. ▪ **[3]** 3xx or REFER = Applies to options [1] and [2]. ▪ **[4]** Initial only = This routing rule is used for regular requests that the device forwards to the destination. This rule is not used for re-routing of requests triggered by the receipt of REFER or 3xx.
Call Setup Rules Set Id CLI: call-setup-rules-set-id **[IP2IPRouting_CallSetupRulesSetId]**	Assigns a Call Setup Rule Set ID to the routing rule. The device performs the Call Setup rules of this Set ID if the incoming call matches the characteristics of this routing rule. The device routes the call to the destination according to the routing rule's configured action, only after it has performed the Call Setup rules. For configuring Call Setup rules, see "Configuring Call Setup Rules" on page 241.
Operation Routing Rule - Action	
Destination Type **[IP2IPRouting_DestType]** CLI: dst-type	Determines the destination type to which the outgoing SIP dialog is sent. ▪ **[0]** IP Group = (Default) The SIP dialog is sent to the IP Group’s Proxy Set (SERVER-type IP Group) or registered contact from the database (if USER-type IP Group). ▪ **[1]** Dest Address = The SIP dialog is sent to the address configured in the following fields: 'Destination SRD ID', 'Destination Address', 'Destination Port', and 'Destination Transport Type'. ▪ **[2]** Request URI = The SIP dialog is sent to the address indicated in the incoming Request-URI. If the fields 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these fields take precedence. ▪ **[3]** ENUM = An ENUM query is sent to include the

Parameter	Description

	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 528.
	▪ **[5]** Dial Plan = The IP destination is determined by a Dial Plan index of the loaded Dial Plan file. The syntax of the Dial Plan index in the Dial Plan file is as follows: <destination / called prefix number>,0,<IP destination>
	Note that the second parameter "0" is ignored. An example of a configured Dial Plan (# 6) in the Dial Plan file is shown below:
	`[ PLAN6 ]`
	`200,0,10.33.8.52     ; called prefix 200 is routed to destination 10.33.8.52`
	`201,0,10.33.8.52`
	`300,0,itsp.com       ; called prefix 300 is routed to destination itsp.com`
	Once the Dial Plan is defined, you need to assign it (0 to 7) to the routing rule as the destination in the 'Destination Address' parameter, where "0" denotes [PLAN1], "1" denotes [PLAN2], and so on.
	▪ [7] LDAP = LDAP-based routing.
	▪ [8] Gateway = The device routes the SBC call to the Tel side (Gateway call) using an IP-to-Tel routing rule in the Inbound IP Routing table (see Configuring Inbound IP Routing Table on page 406). The IP-to-Tel routing rule must be configured with the same call matching characteristics as this SBC IP-to-IP routing rule. This option is also used for alternative routing of an IP-to-IP route to the PSTN. In such a case, the IP-to-Tel routing rule must also be configured with the same call matching characteristics as this SBC IP-to-IP routing rule.

Destination IP Group ID **[IP2IPRouting_DestIPGroupID]** CLI: dst-ip-group-id	Defines the IP Group ID to where you want to route the call. The SIP dialog messages are sent to the IP address defined for the Proxy Set associated with this IP Group. If you select an IP Group, it is unnecessary to configure a destination IP address (in the 'Destination Address' field). However, if both parameters are configured, then the IP Group takes precedence. If the destination IP Group is of USER type, the device searches for a match between the Request-URI (of the received SIP dialog) to an AOR registration record in the device's database. The SIP dialog is then sent to the IP address of the registered contact. The default is -1. **Notes:** ▪ This parameter is only relevant if the parameter 'Destination Type' is set to **IP Group**. However, regardless of the settings of the parameter 'Destination Type', the IP Group is still used - only for determining the IP Profile or outgoing SRD. If neither IP Group nor SRD

Parameter	Description
	are defined in this table, the destination SRD is determined according to the source SRD associated with the Source IP Group (configured in the IP Group table, see "Configuring IP Groups" on page 270). If this table does not define an IP Group but only an SRD, the first IP Group associated with this SRD (in the IP Group table) is used. ▪ If the destination IP Group ID is of SERVER type, the request is routed according to the IP Group addresses. ▪ If the destination IP Group ID is of USER type, the request is routed according to the IP Group specific database (i.e., only to registered users of the selected database). ▪ If the destination IP Group ID is ANY USER ([-2]), the request is routed according to the general database (i.e., any matching registered user).
Destination SRD ID [IP2IPRouting_DestSRDID] CLI: dst-srd-id	Defines the SRD ID. The default is None. **Note:** The destination IP Group must belong to the destination SRD if both are configured in this table.
Destination Address **[IP2IPRouting_DestAddress]** CLI: dst-address	Defines the destination to where the call is sent. This can be an IP address or a domain name (e.g., domain.com). If ENUM-based routing is used (i.e., the 'Destination Type' parameter is set to **ENUM**) this parameter defines the IP address or domain name (FQDN) of the ENUM service, for example, e164.arpa, e164.customer.net or NRENum.net. The device sends the ENUM query containing the destination phone number to an external DNS server, configured in the Interface table. The ENUM reply includes a SIP URI (user@host) which is used as the destination Request-URI in this routing table. The valid value is a string of up to 50 characters. **Notes:** ▪ This parameter is applicable only if the 'Destination Type' parameter is set to **Dest Address** [1] or **ENUM** [3]. ▪ When using domain names, enter a DNS server IP address or alternatively, define these names in the Internal DNS table (see "Configuring the Internal SRV Table" on page 145). ▪ To terminate SIP OPTIONS messages at the device (i.e., to handle them locally), set this parameter to "internal".
Destination Port **[IP2IPRouting_DestPort]** CLI: dst-port	Defines the destination port to where the call is sent.
Destination Transport Type **[IP2IPRouting_DestTransportType]** CLI: dst-transport-type	Defines the transport layer type for sending the call: ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** If this parameter is not configured, the transport type is determined by the SIPTransportType parameter.

Parameter	Description
Alternative Route Options [IP2IPRouting_AltRouteOptions] CLI: alt-route-options	Determines whether this routing rule is the main routing rule or an alternative routing rule (to the rule defined directly above it in the table). ▪ [0] Route Row (default) = Main routing rule - the device first attempts to route the call to this route if the incoming SIP dialog's input characteristics matches this rule. ▪ [1] Alt Route Ignore Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route regardless of the incoming SIP dialog's input characteristics. ▪ [2] Alt Route Consider Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route only if the incoming SIP dialog matches this routing rule's input characteristics. ▪ [3] Group Member Ignore Inputs = This routing rule is a member of the Forking routing rule. The incoming call is also forked to the destination of this routing rule. The matching input characteristics of the routing rule are ignored. ▪ [4] Group Member Consider Inputs = This routing rule is a member of the Forking routing rule. The incoming call is also forked to the destination of this routing rule only if the incoming call matches this rule's input characteristics. **Notes:** ▪ The alternative routing entry ([1] or [2]) must be defined in the next consecutive table entry index to the Route Row entry (i.e., directly below it). For example, if Index 4 is configured as a Route Row, Index 5 must be configured as the alternative route. ▪ The Forking Group members must be configured in a table row that is immediately below the main Forking routing rule, or below an alternative routing rule for the main rule, if configured. ▪ For IP-to-IP alternative routing, configure alternative routing reasons upon receipt of 4xx, 5xx, and 6xx SIP responses (see Configuring SIP Response Codes for Alternative Routing Reasons on page 561). However, if no response, ICMP, or a SIP 408 response is received, the device attempts to use the alternative route even if no entries are configured in the 'SBC Alternative Routing Reasons' table. ▪ Multiple alternative route entries can be configured (e.g., Index 1 is the main route - Route Row - and indices 2 through 4 are configured as alternative routes).
Group Policy CLI: group-policy **[IP2IPRouting_GroupPolicy]**	Defines whether the routing rule includes call forking. ▪ **[0]** None (default) = Call uses only this route (even if Forking Group members are configured in the rows below it). ▪ **[1]** Forking = Call uses this route and the routes of Forking Group members, if configured (in the rows below it).

Parameter	Description
Cost Group **[IP2IPRouting_CostGroup]** CLI: cost-group	Assigns a Cost Group to the routing rule for determining the cost of the call. To configure Cost Groups, see "Configuring Cost Groups" on page 238. By default, no Cost Group is defined.

37.4 Configuring SIP Response Codes for Alternative Routing Reasons

The SBC Alternative Routing Reasons table lets you configure up to 20 SIP response codes for call release (termination) reasons. If a call (outgoing SIP dialog-initiating methods, e.g., INVITE, OPTIONS, and SUBSCRIBE messages) is released as a result of a configured SIP code (provided in SIP 4xx, 5xx, and 6xx), the device attempts to locate an alternative route for the call in the IP-to-IP Routing table. Alternative routing rules are configured with the 'Alternative Route Options' parameter set to **Alt Route Ignore Inputs** or **Alt Route Consider Inputs** (see "Configuring SBC IP-to-IP Routing Rules" on page 552).

Typically, the device performs alternative routing when there is no response at all to an INVITE message. This is done after a user-defined number of INVITE re-transmissions, configured by the SIPMaxRtx parameter. In such a scenario, the device issues itself the SIP response code 408 (Request Timeout). Alternative routing is only done if you have configured this response code in the SBC Alternative Routing Reasons table.

You can also configure alternative routing for the following proprietary response codes that are issued by the device itself:

- **805 IP Profile Call Limit:** The device generates this response code when Call Admission Control (CAC) limits (such as maximum concurrent calls) are exceeded for an IP Group (or SRD). The CAC rules are configured in the Admission Control table (see "Configuring Admission Control" on page 539). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity. In such a scenario, an alternative route configured in the IP-to-IP Routing table can be used.
- **806 Media Limits Exceeded:** The device generates this response code when the call is terminated due to crossed thresholds of QoE metrics such as MOS, packet delay, and packet loss (configured in the Quality of Experience Profile table) and/or media bandwidth (configured in the Bandwidth profile table). When this occurs, the device sends a SIP 480 (Temporarily Unavailable) response to the SIP entity. This is configured by 1) assigning an IP Group a QoE and/or Bandwidth profile that rejects calls if the threshold is crossed, 2) configuring 806 in the SBC Alternative Routing Reasons table and 3) configuring an alternative routing rule.

Notes:

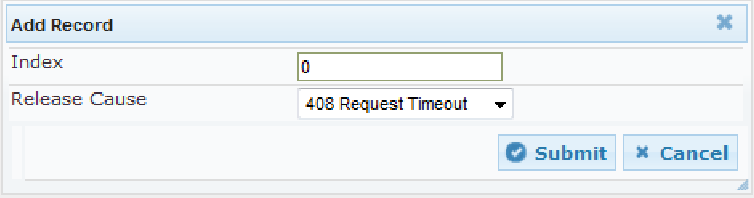
- If the device receives a SIP 408 response, an ICMP message, or no response, alternative routing is still performed even if the SBC Alternative Routing Reasons table is not configured.
- SIP requests belonging to an SRD or IP Group that have reached the call limit (maximum concurrent calls and/or call rate) as configured in the Call Admission table are sent to an alternative route if configured in the IP-to-IP Routing table for the SRD or IP Group. If no alternative routing rule is located, the device automatically rejects the SIP request with a SIP 486 (Busy Here) response.

The following procedure describes how to configure the SBC Alternative Routing Reasons table in the Web interface. You can also configure this table using the table ini file parameter, SBCAlternativeRoutingReasons or CLI command, configure voip > sbc routing sbc-alt-routing-reasons.

➢ **To configure SIP reason codes for alternative IP routing:**

1. Open the SBC Alternative Routing Reasons page (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **Alternative Routing Reasons**).
2. Click **Add**; the following dialog box appears:

Figure 37-6: Alternative Routing Reasons Table - Add Record



Add Record

Index: 0

Release Cause: 408 Request Timeout

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

Table 37-4: SBC Alternative Routing Reasons Table Parameter Descriptions

Parameter	Description
Index **[SBCAlternativeRoutingReasons_Index]**	Defines an index number for the new table record.
Release Cause CLI: rel-cause **[SBCAlternativeRoutingReasons_ReleaseCause]**	Defines a SIP response code for triggering the device's alternative routing mechanism.

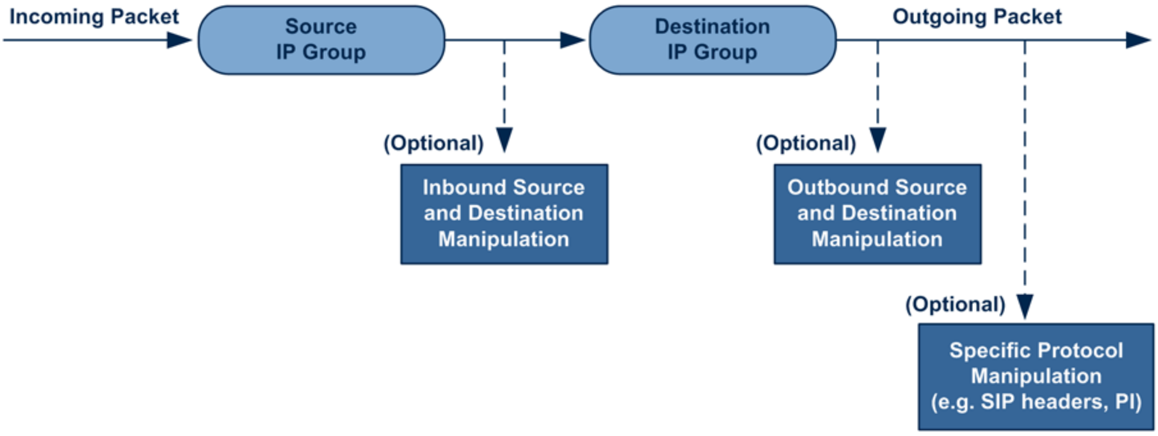
38 SBC Manipulations

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 299.

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 38-1: SIP URI Manipulation in IP-to-IP Routing


Incoming Packet
Source IP Group
Destination IP Group
Outgoing Packet
(Optional)
Inbound Source and Destination Manipulation
(Optional)
Outbound Source and Destination Manipulation
(Optional)
Specific Protocol Manipulation (e.g. SIP headers, PI)

You can also restrict source user identity in outgoing SIP dialogs in the Outbound Manipulation table (using the column PrivacyRestrictionMode). The device identifies an incoming user as restricted if one of the following exists:

- From header user is 'anonymous'.
- P-Asserted-Identity and Privacy headers contain the value 'id'.

All restriction logic is done after the user number has been manipulated.

Host name (source and destination) manipulations are simply host name substitutions with the names defined for the source and destination IP Groups respectively (if any, in the IP Group table).

Below is an example of a call flow and consequent SIP URI manipulations:

- **Incoming INVITE from LAN:**

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
Via: SIP/2.0/UDP 10.2.2.6;branch=z9hGLLLLLan
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OlLAN;paramer1
=abe
To: <sip:1000@10.2.2.3;user=phone>
Call-ID: USELLLAN@10.2.2.3
CSeq: 1 INVITE
```

```
Contact: <sip:7000@10.2.2.3>
Supported: em,100rel,timer,replaces
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK
User-Agent: Sip Message Generator V1.0.0.5
Content-Type: application/sdp
Content-Length: 155
v=0
o=SMG 791285 795617 IN IP4 10.2.2.6
s=Phone-Call
c=IN IP4 10.2.2.6
t=0 0
m=audio 6000 RTP/AVP 8
a=rtpmap:8 pcma/8000
a=sendrecv
a=ptime:20
```

- **Outgoing INVITE to WAN:**

```
INVITE sip: 9721000@ITSP;user=phone;x=y;z=a SIP/2.0
Via: SIP/2.0/UDP 212.179.1.12;branch=z9hGWwan
From:
<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;paramer1=abe
To: <sip: 9721000@ ITSP;user=phone>
Call-ID: USEVWWAN@212.179.1.12
CSeq: 38  INVITE
Contact: <sip:7000@212.179.1.12>
Supported: em,100rel,timer,replaces
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER
User-Agent: Sip Message Generator V1.0.0.5
Content-Type: application/sdp
Content-Length: 155
v=0
o=SMG 5  9  IN IP4 212.179.1.11
s=Phone-Call
c=IN IP4 212.179.1.11
t=0 0
m=audio 8000 RTP/AVP 8
a=rtpmap:8 pcma/8000
a=sendrecv
a=ptime:20
```

The SIP message manipulations in the example above (contributing to typical topology hiding) are as follows:

- Inbound source SIP URI user name from "7000" to "97000":

```
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OlLAN;paramer1
=abe
```

to

```
From:
<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;paramer1=abe
```

- Source IP Group name (i.e., SIP URI host name) from "10.2.2.6" to "IP_PBX":

```
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OlLAN;paramer1
=abe
```

to

```
From:
<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;paramer1=abe
```

- Inbound destination SIP URI user name from "1000" to 9721000":

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
To: <sip:1000@10.2.2.3;user=phone>
```

to

```
INVITE sip:9721000@ITSP;user=phone;x=y;z=a SIP/2.0
To: <sip:9721000@ITSP;user=phone>
```

- Destination IP Group name (SIP URI host name) from "10.2.2.3" to "ITSP":

```
INVITE sip:1000@10.2.2.3;user=phone;x=y;z=a SIP/2.0
To: <sip:1000@10.2.2.3;user=phone>
```

to

```
INVITE sip:9721000@ITSP;user=phone;x=y;z=a SIP/2.0
To: <sip:9721000@ITSP;user=phone>
```

38.1 Configuring IP-to-IP Inbound Manipulations

The IP to IP Inbound Manipulation table lets you configure up to 100 IP-to-IP Inbound Manipulation rules. An IP-to-IP Inbound Manipulation rule defines a manipulation sequence for the source or destination SIP URI user part of inbound SIP dialog requests. You can apply these manipulations to different SIP dialog message types (e.g., INVITE or REGISTER) and SIP headers as follows:

- Manipulated destination URI user part are done on the following SIP headers: Request-URI, To, and Remote-Party-ID (if exists)
- Manipulated source URI user part are done on the following SIP headers: From, P-Asserted-Identity (if exists), P-Preferred-Identity (if exists), and Remote-Party-ID (if exists)

An IP-to-IP Inbound Manipulation rule includes two areas:

- Matching characteristics (Rule) - characteristics of incoming SIP dialog such as source host name.
- Operation (Action) - if the incoming call matches the characteristics of the rule, the device manipulates the source or destination SIP URI user part of the SIP dialog (e.g., removes user-defined number of characters from the left of the SIP URI user part).

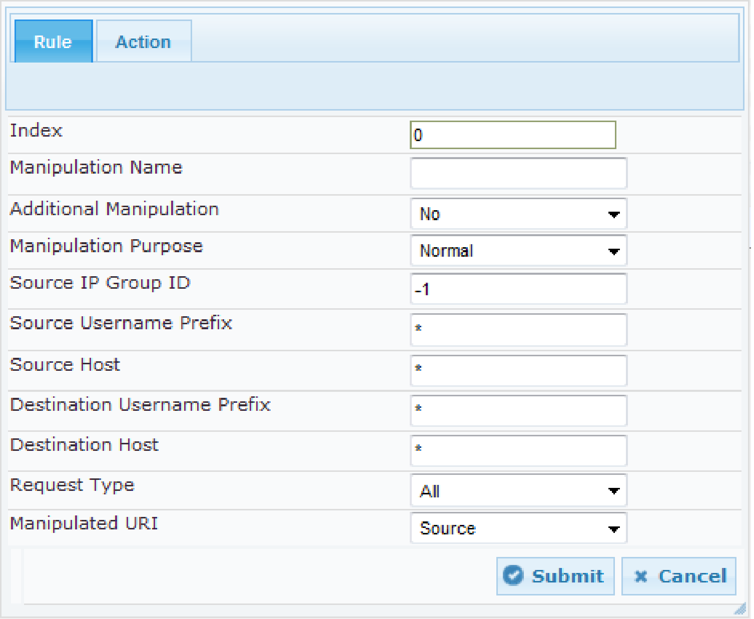
Note: The IP Group table can be used to configure a host name that overwrites the received host name. This manipulation can be done for source and destination IP Groups (see "Configuring IP Groups" on page 270).

The following procedure describes how to configure IP-to-IP Inbound Manipulation rules in the Web interface. You can also configure these rules using the table ini file parameter, IPInboundManipulation or CLI command, configure voip > sbc manipulations ip-inbound-manipulation.

➢ To configure an IP-to-IP Inbound Manipulation rule:

1. Open the IP to IP Inbound Manipulation page (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP to IP Inbound**).
2. Click **Add**; the following dialog box appears:

Figure 38-2: IP to IP Inbound Manipulation Page - Add Dialog Box



Index	0
Manipulation Name	
Additional Manipulation	No
Manipulation Purpose	Normal
Source IP Group ID	-1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Request Type	All
Manipulated URI	Source

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

Table 38-1: IP to IP Inbound Manipulation Parameter Descriptions

Parameter	Description
Index **[IPInboundManipulation_Index]**	Defines an index number for the new table record.
Manipulation Name CLI: manipulation-name **[IPInboundManipulation_ManipulationName]**	Defines an arbitrary name to easily identify the manipulation rule. The valid value is a string of up to 20 characters. By default, no value is defined.
Matching Characteristics - Rule	
Additional Manipulation CLI: is-additional-manipulation **[IPInboundManipulation_IsAdditionalManipulation]**	Determines whether additional SIP URI user part manipulation is done for the table entry rule listed directly above it. ▪ **[0]** No = (Default) Regular manipulation rule (not done in addition to the rule above it). ▪ **[1]** Yes = If the above row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule. **Note:** Additional manipulation can only be done on a different SIP URI, source or destination, to the rule configured in the row above as configured by the 'Manipulated URI' parameter (see below).
Manipulation Purpose CLI: purpose	Defines the purpose of the manipulation: ▪ **[0]** Normal = (Default) Inbound manipulations affect the

Parameter	Description
[IPInboundManipulation_ManipulationPurpose]	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 526.
Source IP Group ID CLI: src-ip-group-id **[IPInboundManipulation_SrcIpGroup]**	Defines the IP Group from where the incoming INVITE is received. The default is -1 (i.e., any IP Group).
Source Username Prefix CLI: src-user-name-prefix **[IPInboundManipulation_SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name (usually in the From header). The default is the asterisk (*) symbol (i.e., any source username prefix). **Note:** The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 761.
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 761.
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_Manipula**	Determines whether the source or destination SIP URI user part is manipulated. - **[0]** Source = (Default) Manipulation is done on the source

Parameter	Description
tedURI]	SIP URI user part. - **[1]** Destination = Manipulation is done on the destination SIP URI user part.
Operation Rule - Action	
Remove From Left CLI: remove-from-left **[IPInboundManipulation_RemoveFromLeft]**	Defines the number of digits to remove from the left of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".
Remove From Right CLI: remove-from-right **[IPInboundManipulation_RemoveFromRight]**	Defines the number of digits to remove from the right of the user name prefix. For example, if you enter 3 and the user name is "john", the new user name is "j". **Note:** If both 'Remove From Right' and 'Leave From Right' parameters are configured, the 'Remove From Right' setting is applied first.
Leave From Right CLI: leave-from-right **[IPInboundManipulation_LeaveFromRight]**	Defines the number of characters that you want retained from the right of the user name. **Note:** If both 'Remove From Right' and 'Leave From Right' parameters are configured, the 'Remove From Right' setting is applied first.
Prefix to Add CLI: prefix-to-add **[IPInboundManipulation_Prefix2Add]**	Defines the number or string that you want added to the front of the user name. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn".
Suffix to Add CLI: suffix-to-add **[IPInboundManipulation_Suffix2Add]**	Defines the number or string that you want added to the end of the user name. For example, if you enter '01' and the user name is "john", the new user name is "john01".

38.2 Configuring IP-to-IP Outbound Manipulations

The IP to IP Outbound Manipulation table lets you configure up to 100 IP-to-IP Outbound Manipulation rules. An IP-to-IP Outbound Manipulation rule defines a manipulation action for the SIP Request-URI user part (source or destination) or calling name of outbound SIP dialog requests. The IP-to-IP Outbound Manipulation rules can be applied to any SIP request type (e.g., INVITE). Manipulated destination URI user part are done on the SIP headers - Request URI, To, and Remote-Party-ID (if exists). Manipulated source URI user part are done on the SIP headers - From, P-Asserted (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists).

An IP-to-IP Outbound Manipulation rule includes two areas:

- Matching characteristics (Rule) - characteristics of incoming SIP dialog such as source host name. As the device performs outbound manipulations only after the routing process, the IP-to-IP Outbound Manipulation rule can also use destination IP Groups as matching characteristics.
- Operation (Action) - if the incoming call matches the characteristics of the rule, the device manipulates the source or destination SIP URI user part or calling name of the SIP dialog (e.g., removes user-defined number of characters from the left of the SIP URI user part).

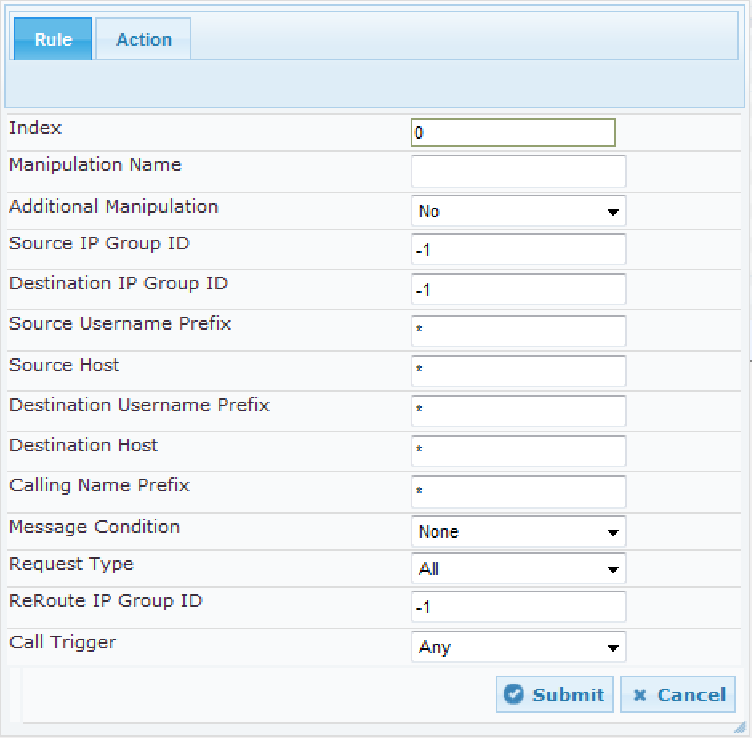
Note: SIP URI host name (source and destination) manipulations can also be configured in the IP Group table. These manipulations are simply host name substitutions with the names configured for the source and destination IP Groups, respectively.

The following procedure describes how to configure IP-to-IP Outbound Manipulation rules in the Web interface. You can also configure these rules using the table ini file parameter, IPOutboundManipulation or CLI command, configure voip > sbc manipulations ip-outbound-manipulation.

➢ To configure IP-to-IP outbound manipulation rules:

1. Open the IP to IP Outbound Manipulation page (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP to IP Outbound**).
2. Click **Add**; the following dialog box appears:

Figure 38-3: IP to IP Outbound Manipulation Page - Add Dialog Box



Index	0
Manipulation Name	
Additional Manipulation	No
Source IP Group ID	-1
Destination IP Group ID	-1
Source Username Prefix	*
Source Host	*
Destination Username Prefix	*
Destination Host	*
Calling Name Prefix	*
Message Condition	None
Request Type	All
ReRoute IP Group ID	-1
Call Trigger	Any

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

Table 38-2: IP to IP Outbound Manipulation Table Parameter Description

Parameter	Description
Index **[IPOutboundManipulation_Index]**	Defines an index number for the new table record.
Manipulation Name CLI: manipulation-name **[IPOutboundManipulation_ManipulationName]**	Defines an arbitrary name to easily identify the manipulation name. The valid value is a string of up to 20 characters. By default, no value is defined.

Parameter	Description
Matching Characteristics - Rule	
Additional Manipulation CLI: is-additional-manipulation **[IPOutboundManipulation_IsAdditionalManipulation]**	Determines whether additional manipulation is done for the table entry rule listed directly above it. ▪ **[0]** No = (Default) Regular manipulation rule - not done in addition to the rule above it. ▪ **[1]** Yes = If the previous table row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule. **Note:** Additional manipulation can only be done on a different item (source URI, destination URI, or calling name) to the rule configured in the row above (configured by the 'Manipulated URI' parameter).
Source IP Group ID CLI: src-ip-group-id **[IPOutboundManipulation_SrcIPGroupID]**	Defines the IP Group from where the INVITE is received. The default value is -1 (i.e., any IP Group).
Destination IP Group ID CLI: dst-ip-group-id **[IPOutboundManipulation_DestIPGroupID]**	Defines the IP Group to where the INVITE is to be sent. The default value is -1 (i.e., any IP Group).
Source Username Prefix CLI: src-user-name-prefix **[IPOutboundManipulation_SrcUsernamePrefix]**	Defines the prefix of the source SIP URI user name, typically used in the SIP From header. The default value is the asterisk (*) symbol (i.e., any source username prefix). The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 761.
Source Host CLI: src-host **[IPOutboundManipulation_SrcHost]**	Defines the source SIP URI host name - full name, typically in the From header. The default value is the asterisk (*) symbol (i.e., any source host name).
Destination Username Prefix CLI: dst-user-name-prefix **[IPOutboundManipulation_DestUsernamePrefix]**	Defines the prefix of the destination SIP URI user name, typically located in the Request-URI and To headers. The default value is the asterisk (*) symbol (i.e., any destination username prefix). The prefix can be a single digit or a range of digits. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 761.
Destination Host CLI: dst-host **[IPOutboundManipulation_DestHost]**	Defines the destination SIP URI host name - full name, typically located in the Request-URI and To headers. The default value is the asterisk (*) symbol (i.e., any destination host name).
Calling Name Prefix CLI: calling-name-prefix **[IPOutboundManipulation_CallingNamePrefix]**	Defines the prefix of the calling name (caller ID). The calling name appears in the SIP From header. The valid value is a string of up to 37 characters. By default, no prefix is defined.
Message Condition CLI: message-condition **[IPOutboundManipulation_MessageCondition]**	Assigns a Message Condition rule as a matching characteristic. Message Condition rules define required SIP message formats. For configuring Message Condition rules, see "Configuring Message Condition Rules" on page 551.
Request Type	Defines the SIP request type to which the manipulation rule

Parameter	Description
CLI: request-type **[IPOutboundManipulation_RequestType]**	is applied. ▪ **[0]** All = (Default) all SIP messages. ▪ **[1]** INVITE = All SIP messages except REGISTER and SUBSCRIBE. ▪ **[2]** REGISTER = Only SIP REGISTER messages. ▪ **[3]** SUBSCRIBE = Only SIP SUBSCRIBE messages. ▪ **[4]** INVITE and REGISTER = All SIP messages except SUBSCRIBE. ▪ **[5]** INVITE and SUBSCRIBE = All SIP messages except REGISTER.
ReRoute IP Group ID CLI: re-route-ip-group-id **[IPOutboundManipulation_ReRouteIPGroupID]**	Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. This parameter is typically used for re-routing requests (e.g., INVITEs) when interworking is required for SIP 3xx redirect responses or REFER messages. The default is -1 (i.e., not configured). **Notes:** ▪ This parameter functions together with the 'Call Trigger' parameter (see below). ▪ For more information on interworking of SIP 3xx redirect responses or REFER messages, see "Interworking SIP 3xx Redirect Responses" on page 519 and "Interworking SIP REFER Messages" on page 521, respectively.
Call Trigger CLI: trigger **[IPOutboundManipulation_Trigger]**	Defines the reason (i.e., trigger) for the re-routing of the SIP request: ▪ **[0]** Any = (Default) Re-routed for all scenarios (re-routes and non-re-routes). ▪ **[1]** 3xx = Re-routed if it triggered as a result of a SIP 3xx response. ▪ **[2]** REFER = Re-routed if it triggered as a result of a REFER request. ▪ **[3]** 3xx or REFER = Applies to options [1] and [2]. ▪ **[4]** Initial only = Regular requests that the device forwards to a destination. In other words, re-routing of requests triggered by the receipt of REFER or 3xx does not apply.
Operation Manipulation Rule - Action	
Manipulated Item CLI: manipulated-uri **[IPOutboundManipulation_IsAdditionalManipulation]**	Defines the element in the SIP message that you want manipulated. ▪ **[0]** Source URI = (Default) Manipulates the source SIP Request-URI user part. ▪ **[1]** Destination URI = Manipulates the destination SIP Request-URI user part. ▪ **[2]** Calling Name = Manipulates the calling name in the SIP message.
Remove From Left CLI: remove-from-left **[IPOutboundManipulation_RemoveFromLeft]**	Defines the number of digits to remove from the left of the manipulated item prefix. For example, if you enter 3 and the user name is "john", the new user name is "n".

Parameter	Description
Remove From Right CLI: remove-from-right **[IPOutboundManipulation_RemoveFromRight]**	Defines the number of digits to remove from the right of the manipulated item prefix. For example, if you enter 3 and the user name is "john", the new user name is "j".
Leave From Right CLI: leave-from-right **[IPOutboundManipulation_LeaveFromRight]**	Defines the number of digits to keep from the right of the manipulated item.
Prefix to Add CLI: prefix-to-add **[IPOutboundManipulation_Prefix2Add]**	Defines the number or string to add in the front of the manipulated item. For example, if you enter 'user' and the user name is "john", the new user name is "userjohn". If you set the 'Manipulated Item' parameter to **Source URI** or **Destination URI**, you can configure this parameter to a string of up 49 characters. If you set the 'Manipulated Item' parameter to **Calling Name**, you can configure this parameter to a string of up 36 characters.
Suffix to Add CLI: suffix-to-add **[IPOutboundManipulation_Suffix2Add]**	Defines the number or string to add at the end of the manipulated item. For example, if you enter '01' and the user name is "john", the new user name is "john01". If you set the 'Manipulated Item' parameter to **Source URI** or **Destination URI**, you can configure this parameter to a string of up 49 characters. If you set the 'Manipulated Item' parameter to **Calling Name**, you can configure this parameter to a string of up 36 characters.
Privacy Restriction Mode CLI: privacy-restriction-mode **[IPOutboundManipulation_PrivacyRestrictionMode]**	Determines user privacy handling (i.e., restricting source user identity in outgoing SIP dialogs). ▪ **[0]** Transparent = (Default) No intervention in SIP privacy. ▪ **[1]** Don't change privacy = The user identity remains the same as in the incoming SIP dialog. If a restricted number exists, the restricted presentation is normalized as follows: ✓ From URL header: anonymous@anonymous.invalid. ✓ If a P-Asserted-Identity header exists (either in the incoming SIP dialog or added by the device), a Privacy header is added with the value "id". ▪ **[2]** Restrict = The user identity is restricted (the restricted presentation is as mentioned above). ▪ **[3]** Remove Restriction = The device attempts to reveal the user identity by setting user values in the From header and removing the privacy "id" value if the Privacy header exists. If the From header user is anonymous, the value is taken from the P-Preferred-Identity, P-Asserted-Identity, or Remote-Party-ID header (if exists). The device identifies an incoming user as restricted if one of the following exists: ▪ From header user is anonymous. ▪ P-Asserted-Identity and Privacy headers contain the value "id". **Note:** All restriction logic is performed after the user number has been manipulated.

Part VII

Cloud Resilience Package

39 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. It does this by using its Call Survivability feature, enabling the branch users to call one another or make external calls through the device's PSTN gateway interface (if configured).

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 621.
- 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 1021.
- 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 39-1: 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	▪ 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
numbers are learned dynamically in the registration process) - Survivability indication to IP phone - Call hold and retrieve - Call transfer (if the 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.

40 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 Settings" on page 535
- "Configuring Admission Control" on page 539
- "Configuring Allowed Audio Coder Groups" on page 543
- "Configuring Classification Rules" on page 545
- "Configuring Message Condition Rules" on page 551
- "Configuring SBC IP-to-IP Routing Rules" on page 552
- "Configuring SIP Response Codes for Alternative Routing Reasons" on page 561
- "Configuring IP-to-IP Inbound Manipulations" on page 565
- "Configuring IP-to-IP Outbound Manipulations" on page 568

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".

40.1 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 621.

➢ **To enable the CRP application:**

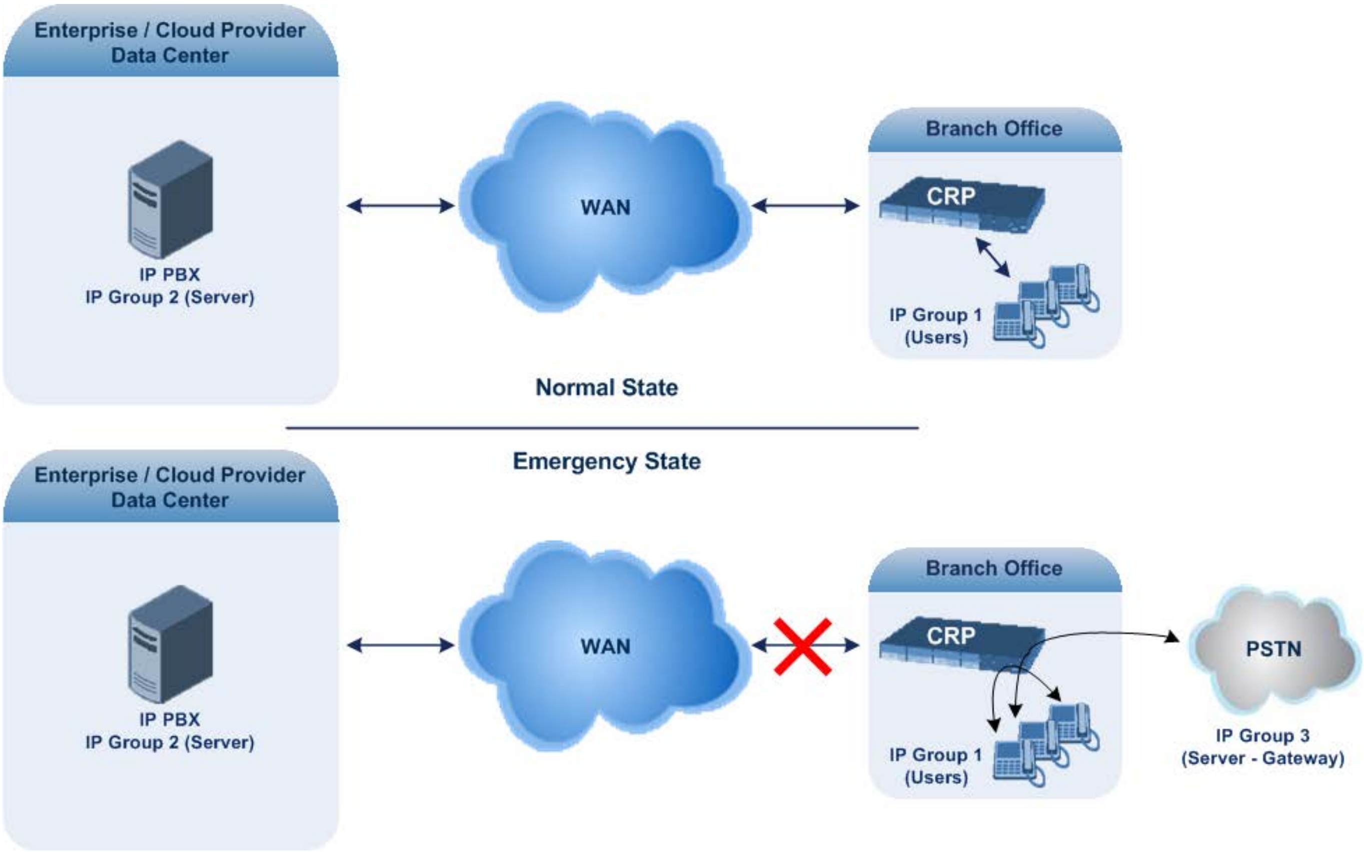
1. Open the Applications Enabling page (**Configuration** tab > **VoIP** menu > **Applications Enabling** > **Applications Enabling**).
2. From the 'CRP Application' drop-down list, select **Enable**.
3. Click **Submit**, and then reset the device with a burn-to-flash for your settings to take effect.

40.2 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 40-1: CRP in Normal & Auto Answer to Registrations Modes

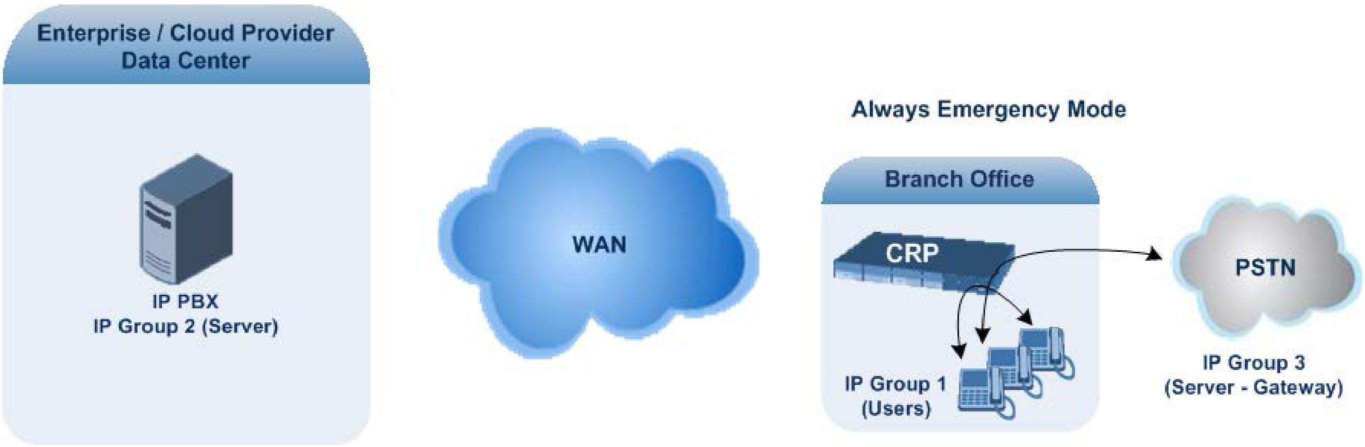


- **Auto Answer to Registrations:** This mode is the same as the Normal mode, except that the CRP registers the branch users in its registration database instead of forwarding them to the IP PBX.

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 40-2: 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**.

40.3 Pre-Configured IP Groups

For CRP, the device is pre-configured with the following IP Groups in the IP Group table:

Table 40-1: Pre-configured IP Groups in the IP Group Table

Index	Type	Description
1	User	Users
2	Server	Proxy
3	Server	Gateway

These IP Groups represent the following IP entities:

- **"Users" IP Group:** LAN users (e.g., IP phones) at the branch office
- **"Server" IP Group:** Server (e.g., hosted IP PBX at the Enterprise's headquarters)
- **"Gateway" IP Group:** 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 580).

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 270.

40.4 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 578.

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 552.

40.4.1 Normal Mode

The pre-configured IP-to-IP routing rules for the Normal CRP call survivability mode are shown in the table below:

Table 40-2: Pre-Configured IP-to-IP Routing Rules for CRP Normal Mode

Index	Source IP Group ID / Emergency	Request Type	Destination Type	Destination IP Group ID	Destination Address	Alternative Route Options
1	*	OPTIONS	Dest Address	-	Internal	Route Row
3	1	All	IP Group	2	-	Route Row
4	1	All	IP Group	1	-	Alternative
5	1	All	IP Group	3	-	Alternative
6	2	All	IP Group	3	-	Route Row
7[1]	2	All	IP Group	3	-	Route Row
8	3	All	IP Group	2	-	Route Row
9	3	All	IP Group	1	-	Alternative

Note 1: Index 7 appears only if the CRPGatewayFallback parameter is enabled (see "Configuring PSTN Fallback" on page 582).

40.4.2 Emergency Mode

The pre-configured IP-to-IP routing rules for the Emergency CRP call survivability mode are shown in the table below:

Table 40-3: Pre-Configured IP-to-IP Routing Rules for Emergency Mode

Mode	Index	Source IP Group ID / Emergency	Request Type	Destination Type	Destination IP Group ID	Destination Address	Alternative Route Options
Always Emergency	1	*	OPTIONS	Dest Address	-	Internal	Route Row
	4	1	All	IP Group	1	-	Route Row
	5	1	All	IP Group	3	-	Alternative
	9	3	All	IP Group	1	-	Route Row

40.4.3 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 40-4: Pre-Configured IP-to-IP Routing Rule for Auto Answer to Registrations Mode

Mode	Index	Source IP Group ID	Request Type	Destination Type	Destination IP Group ID	Destination Address	Alternative Route Options
Auto Answer to Registrations	1	*	OPTIONS	Dest Address	-	Internal	Route Row
	2[1]	*	REGISTER	IP Group	-2	-	Route Row
	3	1	All	IP Group	2	-	Route Row
	4	1	All	IP Group	1	-	Alternative
	5	1	All	IP Group	3	-	Alternative
	6	2	All	IP Group	1	-	Route Row
	7[2]	2	All	IP Group	3	-	Route Row
	8	3	All	IP Group	2	-	Route Row
	9	3	All	IP Group	1	-	Alternative

Notes:

1. For the routing rule of Index 2, the destination is the source IP Group (i.e., from where the REGISTER message was received).
2. Index 7 appears only if the CRPGatewayFallback parameter is enabled (see "Configuring PSTN Fallback" on page 582).

40.5 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 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).

Part VIII

Data-Router Configuration

41 Introduction

Notes:

- For data-router configuration, refer to the *CLI Reference Guide*.
- Web-based management for data-router functionality of the MSBR series products is not supported. Instead, CLI is used to configure this functionality. However, AudioCodes recommends using CLI scripting to configure all other functionality as well (i.e., VoIP and System) through the CLI.

This page is intentionally left blank.

Part IX

Maintenance

42 Basic Maintenance

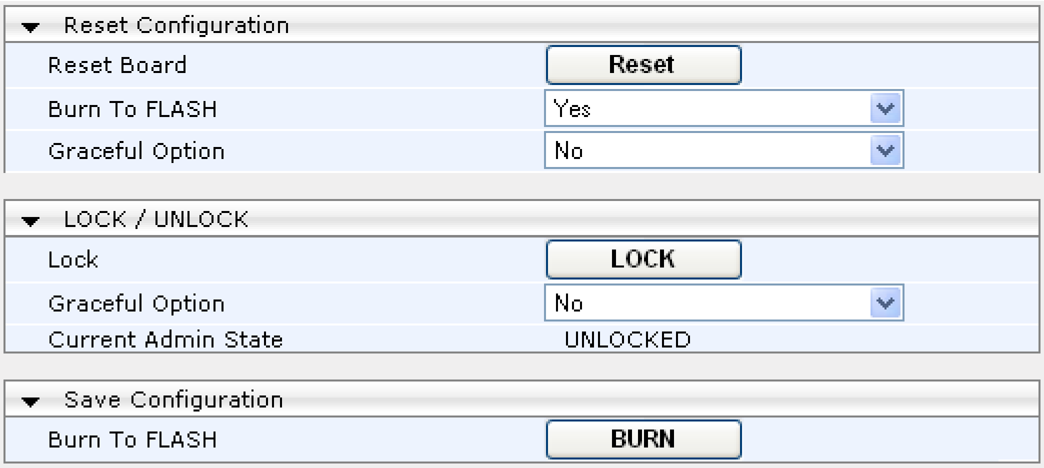
The Maintenance Actions page allows you to perform the following:

- Reset the device - see "Resetting the Device" on page 589
- Lock and unlock the device - see "Locking and Unlocking the Device" on page 591
- Save configuration to the device's flash memory - see "Saving Configuration" on page 592

➢ **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 42-1: Maintenance Actions Page



▼ Reset Configuration

Reset Board	Reset
Burn To FLASH	Yes
Graceful Option	No

▼ LOCK / UNLOCK

Lock	LOCK
Graceful Option	No
Current Admin State	UNLOCKED

▼ Save Configuration

Burn To FLASH	BURN

42.1 Resetting the Device

The Maintenance Actions page allows you to remotely reset the device. In addition, before resetting the device, you can choose the following options:

- Save the device's current configuration to the device's flash memory (non-volatile).
- Perform a graceful shutdown, whereby device reset starts only after a user-defined time (i.e., timeout) or after no more active traffic exists (the earliest thereof).

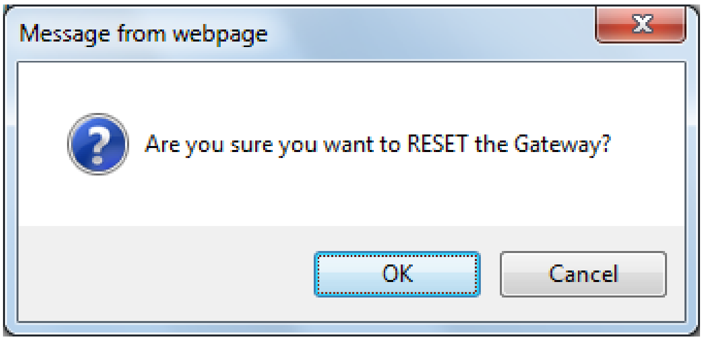
Notes:

- Throughout the Web interface, parameters displayed with a lightning ⚡ symbol are not applied on-the-fly and require that you reset the device for them to take effect.
- When you modify parameters that require a device reset, once you click the **Submit** button in the relevant page, the toolbar displays "Reset" (see "Toolbar Description" on page 48) 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 49).
- Upon reboot, the device restores the settings from its configuration file. However, if reboot attempts fail three times consecutively, the device resets the configuration file by restoring factory defaults before attempting to reboot.

➢ To reset the device:

1. Open the Maintenance Actions page (see "Basic Maintenance" on page 589).
2. Under the 'Reset Configuration' group, from the 'Burn To FLASH' drop-down list, select one of the following options:
   - **Yes:** The device's current configuration is saved (*burned*) to the flash memory prior to reset (default).
   - **No:** Resets the device without saving the current configuration to flash (discards all unsaved modifications).
3. Under the 'Reset Configuration' group, from the 'Graceful Option' drop-down list, select one of the following options:
   - **Yes:** Reset starts only after the user-defined time in the 'Shutdown Timeout' field (see Step 4) expires or after no more active traffic exists (the earliest thereof). In addition, no new traffic is accepted.
   - **No:** Reset starts regardless of traffic, and any existing traffic is terminated at once.
4. In the 'Shutdown Timeout' field (relevant only if the 'Graceful Option' in the previous step is set to **Yes**), enter the time after which the device resets. Note that if no traffic exists and the time has not yet expired, the device resets.
5. Click the **Reset** button; a confirmation message box appears, requesting you to confirm.

Figure 42-2: Reset Confirmation Message Box



Message from webpage

Are you sure you want to RESET the Gateway?

6. Click **OK** to confirm device reset; if the parameter 'Graceful Option' is set to **Yes** (in Step 3), the reset is delayed and a screen displaying the number of remaining calls and time is displayed. When the device begins to reset, a message appears notifying you of this.

42.2 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.

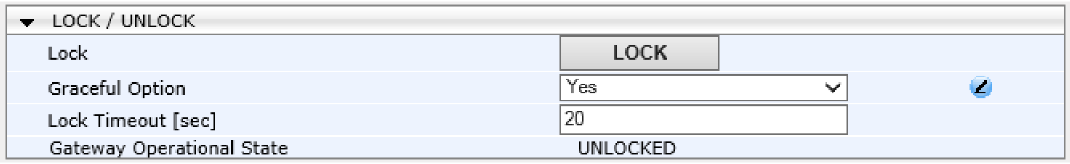
42.3 Locking and Unlocking the Device

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 589).
2. Scroll down to the 'LOCK / UNLOCK' group:

Figure 42-3: Locking the Device



LOCK / UNLOCK

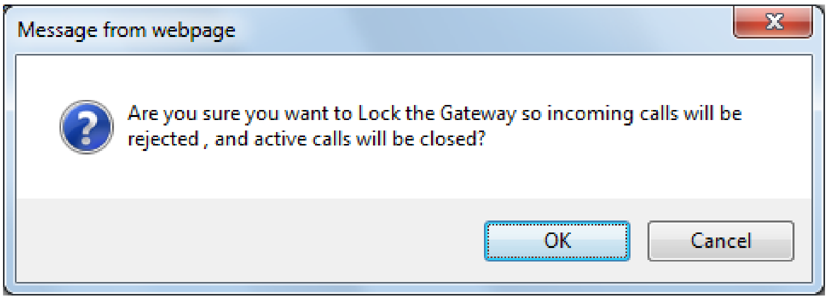
Lock	LOCK
Graceful Option	Yes
Lock Timeout [sec]	20
Gateway Operational State	UNLOCKED

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

   Note: These options are only available if the current status of the device is in UNLOCKED state.
4. If you set 'Graceful Option' to **Yes** (in the previous step), then in the 'Lock Timeout' field, enter the time (in seconds) after which the device locks. If no traffic exists and the time has not yet expired, the device locks immediately.

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

Figure 42-4: Device Lock Confirmation Message Box



Message from webpage

Are you sure you want to Lock the Gateway so incoming calls will be rejected , and active calls will be closed?

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 60).

42.4 Saving Configuration

The Maintenance Actions page allows you to save (*burn*) the current parameter configuration (including loaded auxiliary files) to the device's *non-volatile* memory (i.e., flash). The parameter modifications that you make throughout the Web interface's pages are temporarily saved (to the *volatile* memory - RAM) when you click the **Submit** button on these pages. Parameter settings that are saved only to the device's RAM revert to their previous settings after a hardware/software reset (or power failure). Therefore, to ensure that your configuration changes are retained, you must save them to the device's flash memory using the burn option described below.

➢ **To save the changes to the non-volatile flash memory:**

1. Open the Maintenance Actions page (see "Basic Maintenance" on page 589).
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 591).
- 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 589).
- The Home page's General Information pane displays whether the device is currently "burning" the configuration (see "Viewing the Home Page" on page 60).

43 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.

44 Resetting Channels

44.1 Resetting an Analog Channel

You can inactivate (*reset*) an FXO or FXS analog channel. This is sometimes useful, for example, when the device (FXO) is connected to a PBX and the communication between the two can't be disconnected (e.g., when using reverse polarity). This is done in the Web interface's Home page.

➢ **To reset an analog channel:**

1. Open the Home page.
2. Click the required **FXS** or **FXO** port icon; a shortcut menu appears.
3. From the shortcut menu, choose **Reset Channel**; the channel is changed to inactive and the port icon is displayed in gray.

44.2 Restarting a B-Channel

You can restart a specific B-channel belonging to an ISDN or CAS trunk, using the SNMP MIB variable, acTrunkISDNCommonRestartBChannel or the EMS management tool (refer to the *EMS User's Manual*). This may be useful, for example, for troubleshooting specific voice channels.

Notes:

- If a voice call is currently in progress on the B-channel, it is disconnected when the B-channel is restarted.
- B-channel restart can only be done if the D-channel of the trunk to which it belongs is synchronized (Layer 2).
- B-channel restart does not affect the B-channel's configuration.

This page is intentionally left blank.

45 Disabling Analog Ports

You can disable an analog port (FXS or FXO) on the device. When disabled, the port cannot be used and no signaling is transmitted through the port. In addition, the LED of the analog port is lit red. By default, all the analog ports are enabled.

To disable an analog port, use the following CLI command:

```
(config-voip)# interface fxs-fxo
(fxs-fxo)# analog-port-enable <module>/<port> [on|off]
```

For example, to disable port 2 on module 1:

```
(fxs-fxo)# analog-port-enable 1/2 off
```

This page is intentionally left blank.

46 Software Upgrade

This chapter describes various software update procedures.

46.1 Loading Auxiliary Files

Various Auxiliary files can be installed on the device. These Auxiliary files provide the device with additional configuration settings. The table below lists the different types of Auxiliary files:

Table 46-1: 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.
CAS	CAS auxiliary files containing the CAS Protocol definitions for CAS-terminated trunks (for various types of CAS signaling). You can use the supplied files or construct your own files. Up to eight different CAS files can be installed on the device. For more information, see CAS Files on page 605.
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 600.
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 604.
Dial Plan	Provides dialing plans, for example, to know when to stop collecting dialed digits and start forwarding them or for obtaining the destination IP address for outbound IP routing. For more information, see "Dial Plan File" on page 606.
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 614.

The Auxiliary files can be loaded to the device using one of the following methods:

- Web interface.
- TFTP: This is done by specifying the name of the Auxiliary file in an *ini* file (see Auxiliary and Configuration Files Parameters) and then loading the *ini* file to the device. The Auxiliary files listed in the *ini* file are then automatically loaded through TFTP during device startup. If the *ini* file does not contain a specific auxiliary file type, the device uses the last auxiliary file of that type that was stored on its non-volatile memory.

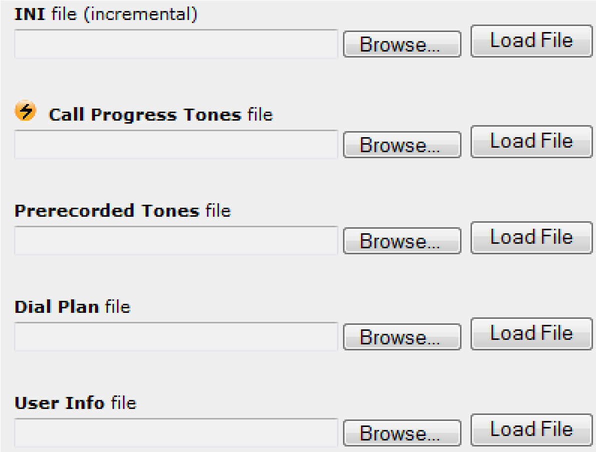
Notes:

- You can schedule automatic loading of updated auxiliary files using HTTP/HTTPS. For more information on automatic updates, see Automatic Update Mechanism.
- When loading an *ini* file using this Web page, parameters that are excluded from the loaded *ini* file retain their current settings (*incremental*).
- Saving an auxiliary file to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device by performing a graceful lock as described in "Locking and Unlocking the Device" on page 591.
- For deleting auxiliary files, see "Viewing Device Information" on page 661.

The following procedure describes how to load Auxiliary files using the Web interface.

➢ **To load auxiliary files to the device using the Web interface:**

1. Open the Load Auxiliary Files page (**Maintenance** tab > **Software Update** menu > **Load Auxiliary Files**).



Note: The appearance of certain file load fields depends on the installed Software License Key.

2. Click the **Browse** button corresponding to the file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name and path of the file appear in the field next to the **Browse** button.
3. Click the **Load File** button corresponding to the file you want to load.
4. Repeat steps 2 through 3 for each file you want to load.
5. Save the loaded auxiliary files to flash memory, see "Saving Configuration" on page 592 and reset the device (if you have loaded a Call Progress Tones file), see "Resetting the Device" on page 589.

46.1.1 Call Progress Tones File

The Call Progress Tones (CPT) and Distinctive Ringing (for analog interfaces only) auxiliary file includes the definitions of the CPT (levels and frequencies) that are detected / generated by the device.

The CPT for analog interfaces is comprised of two sections:

- The first section contains the definitions of the Call Progress Tones (levels and frequencies) that are detected/generated by the device.
- The second section contains the characteristics of the Distinctive Ringing signals that are generated by the device (see Distinctive Ringing on page 603).

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
    - [8] Confirmation Tone
    - [9] Call Waiting Tone - heard by the called party
    - [15] Stutter Dial Tone
    - [16] Off Hook Warning Tone
    - **[17]** Call Waiting Ringback Tone - heard by the calling party
    - **[18]** Comfort Tone
    - **[23]** Hold Tone
    - **[46]** Beep Tone
  - **Tone Modulation Type:** Amplitude Modulated (1) or regular (0)
  - **Tone Form:** The tone's format can be one of the following:

    - Continuous (1)
    - Cadence (2)
    - Burst (3)
- **Low Freq [Hz]:** Frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to AM tones.
- **High Freq [Hz:** Frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).
- **Low Freq Level [-dBm]:** Generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).
- **High Freq Level:** Generation level of 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).
- **First Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For continuous tones, this parameter defines the detection period. For burst tones, it defines the tone's duration.
- **First Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, this parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, this parameter is ignored.
- **Second Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Second Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.
- **Third Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Third Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.
- **Fourth Signal On Time [10 msec]:** 'Signal On' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Fourth Signal Off Time [10 msec]:** 'Signal Off' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.
- **Carrier Freq [Hz]:** Frequency of the carrier signal for AM tones.
- **Modulation Freq [Hz]:** Frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).
- **Signal Level [-dBm]:** Level of the tone for AM tones.
- **AM Factor [steps of 0.02]:** Amplitude modulation factor (valid range from 1 to 50). Recommended values from 10 to 25.

Notes:

- When the same frequency is used for a continuous tone and a cadence tone, the 'Signal On Time' parameter of the continuous tone must have a value that is greater than the 'Signal On Time' parameter of the cadence tone. Otherwise, the continuous tone is detected instead of the cadence tone.
- The tones frequency must differ by at least 40 Hz between defined tones.

For example, to configure the dial tone to 440 Hz only, enter the following text:

```
[NUMBER OF CALL PROGRESS TONES]
Number of Call Progress Tones=1
#Dial Tone
[CALL PROGRESS TONE #0]
Tone Type=1
Tone Form =1 (continuous)
Low Freq [Hz]=440
High Freq [Hz]=0
Low Freq Level [-dBm]=10 (-10 dBm)
High Freq Level [-dBm]=32 (use 32 only if a single tone is
required)
First Signal On Time [10msec]=300; the dial tone is detected after
3 sec
First Signal Off Time [10msec]=0
Second Signal On Time [10msec]=0
Second Signal Off Time [10msec]=0
```

46.1.1.1 Distinctive Ringing

Distinctive Ringing is applicable only to FXS interfaces. Using the Distinctive Ringing section of the Call Progress Tones auxiliary file, you can create up to 16 Distinctive Ringing patterns. Each ringing pattern configures the ringing tone frequency and up to four ringing cadences. The same ringing frequency is used for all the ringing pattern cadences. The ringing frequency can be configured in the range of 10 to 200 Hz with a 5 Hz resolution.

Each of the ringing pattern cadences is specified by the following parameters:

- **Burst Ring On Time:** Configures the cadence to be a burst cadence in the entire ringing pattern. The burst relates to On time and the Off time of the same cadence. It must appear between 'First/Second/Third/Fourth' string and the 'Ring On/Off Time' This cadence rings once during the ringing pattern. Otherwise, the cadence is interpreted as cyclic: it repeats for every ringing cycle.
- **Ring On Time:** Specifies the duration of the ringing signal.
- **Ring Off Time:** Specifies the silence period of the cadence.

The Distinctive Ringing section of the *ini* file format contains the following strings:

- **[NUMBER OF DISTINCTIVE RINGING PATTERNS]:** Contains the following key:
    - 'Number of Distinctive Ringing Patterns' defining the number of Distinctive Ringing signals that are defined in the file.
- **[Ringing Pattern #X]:** Contains the Xth ringing pattern definition (starting from 0 and not exceeding the number of Distinctive Ringing patterns defined in the first section minus 1) using the following keys:
    - **Ring Type:** Must be equal to the Ringing Pattern number.
    - **Freq [Hz]:** Frequency in hertz of the ringing tone.
    - **First (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the first cadence on-off cycle.
    - **First (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the first cadence on-off cycle.
    - **Second (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the second cadence on-off cycle.
    - **Second (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the second cadence on-off cycle.
    - **Third (Burst) Ring On Time [10 msec]:** 'Ring On' period (in 10 msec units) for the third cadence on-off cycle.
    - **Third (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the third cadence on-off cycle.

- **Fourth (Burst) Ring On Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.
- **Fourth (Burst) Ring Off Time [10 msec]:** 'Ring Off' period (in 10 msec units) for the fourth cadence on-off cycle.

Note: In SIP, the Distinctive Ringing pattern is selected according to the Alert-Info header in the INVITE message. For example:
Alert-Info:<Bellcore-dr2>, or Alert-Info:<http://…/Bellcore-dr2>
'dr2' defines ringing pattern #2. If the Alert-Info header is missing, the default ringing tone (0) is played.

An example of a **ringing burst** definition is shown below:

```
#Three ringing bursts followed by repeated ringing of 1 sec on and
3 sec off.
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=1
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=25
First Burst Ring On Time [10msec]=30
First Burst Ring Off Time [10msec]=30
Second Burst Ring On Time [10msec]=30
Second Burst Ring Off Time [10msec]=30
Third Burst Ring On Time [10msec]=30
Third Burst Ring Off Time [10msec]=30
Fourth Ring On Time [10msec]=100
Fourth Ring Off Time [10msec]=300
```

An example of **various ringing signals** definition is shown below:

```
[NUMBER OF DISTINCTIVE RINGING PATTERNS]
Number of Ringing Patterns=3
#Regular North American Ringing Pattern
[Ringing Pattern #0]
Ring Type=0
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 1
[Ringing Pattern #1]
Ring Type=1
Freq [Hz]=20
First Ring On Time [10msec]=200
First Ring Off Time [10msec]=400
#GR-506-CORE Ringing Pattern 2
[Ringing Pattern #2]
Ring Type=2
Freq [Hz]=20
First Ring On Time [10msec]=80
First Ring Off Time [10msec]=40
Second Ring On Time [10msec]=80
Second Ring Off Time [10msec]=400
```

46.1.2 Prerecorded Tones File

The CPT file mechanism has several limitations such as a limited number of predefined tones and a limited number of frequency integrations in one tone. To overcome these limitations and provide tone generation capability that is more flexible, the Prerecorded

Tones (PRT) file can be used. If a specific prerecorded tone exists in the PRT file, it takes precedence over the same tone that exists in the CPT file and is played instead of it.

Notes:

- The PRT file only generates (plays) tones; detection of tones is according to the CPT file.
- Playing tones from the PRT file does not require DSP resources. For local generation of tones, the device requires DSP resources. In addition, if DSPs are being used in a current call (for whatever reason), only local tone generation is supported (tone play from the PRT file is not supported).
- For SBC calls, the PRT file supports only calls that use the G.711 coder.
- For SBC calls, the PRT file supports only the ringback tone and hold tone.

The PRT is a .dat file containing a set of prerecorded tones that can be played by the device. 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 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
- 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 599).

46.1.3 CAS Files

The CAS auxiliary files contain the CAS Protocol definitions that are used for CAS-terminated trunks. You can use the supplied files or construct your own files. Up to eight files can be loaded to the device. Different files can be assigned to different trunks (CASTableIndex_x) and different CAS tables can be assigned to different B-channels (CASChannelIndex).

The CAS files can be loaded to the device using the Web interface or *ini* file (see "Loading Auxiliary Files" on page 599).

Note: All CAS files loaded together must belong to the same trunk type (i.e., either E1 or T1).

46.1.4 Dial Plan File

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

46.1.4.1 Creating a Dial Plan File

The Dial Plan file is a text-based file that can contain up to 8 Dial Plans (Dial Plan indices) and up to 8,000 rules (lines). The general syntax rules for the Dial Plan file are as follows (syntax specific to the feature is described in the respective section):

- Each Dial Plan index must begin with a Dial Plan name enclosed in square brackets "[...]" on a new line.
- Each line under the Dial Plan index defines a rule.
- Empty lines are ignored.
- Lines beginning with a semicolon ";" are ignored. The semicolon can be used for comments.

Creating a Dial Plan file is similar for all Dial Plan features. The main difference is the syntax used in the Dial Plan file and the method for selecting the Dial Plan index.

➢ **To create a Dial Plan file:**

1. Create a new file using a text-based editor (such as Notepad) and configure your Dial Plans, as required.
2. Save the file with the *ini* file extension name (e.g., mydialplanfile.ini).
3. Convert the *ini* file to a *dat* binary file, using AudioCodes DConvert utility. For more information, refer to *DConvert Utility User's Guide*.
4. Load the converted file to the device, as described in "Loading Auxiliary Files" on page 599.
5. Select the Dial Plan index that you want to use. This depends on the feature and is described in the respective section.

46.1.4.2 External Dial Plan File

The device enables you to configure multiple dialing plans in an external Dial Plan file, which can be installed on the device. If a Dial Plan file is implemented, the device first attempts to locate a matching digit pattern in a specified Dial Plan index listed in the file and if not found, attempts to locate a matching digit pattern in the Digit Map. The Digit Map is configured by the 'Digit Mapping Rules' parameter, located in the DTMF & Dialing page (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **DTMF and Supplementary** > **DTMF & Dialing**).

The Dial Plan is used for the following:

- ISDN Overlap Dialing, FXS, and FXO collecting digit mode (Tel-to-IP calls): The file allows the device to know when digit collection ends, after which it starts sending all the collected (or dialed) digits in the outgoing INVITE message. This also provides enhanced digit mapping.
- CAS E1 MF-CR2 (Tel-to-IP calls): Useful for E1 MF-CR2 variants that do not support I-15 terminating digits (e.g., in Brazil and Mexico). The Dial Plan file allows the device to detect end-of-dialing in such cases. The CasTrunkDialPlanName_x ini file parameter determines which dial plan (in the Dial Plan file) to use for a specific trunk.

Notes:

- To use the Dial Plan file, you must also use a special CAS .dat file that supports this feature. For more information, contact your AudioCodes sales representative.
- For E1 CAS MFC-R2 variants, which don't support terminating digit for the called party number, usually I-15, the Dial Plan file and the DigitMapping parameter are ignored. Instead, you can define a Dial Plan template per trunk using the parameter CasTrunkDialPlanName_x.

The Dial Plan file can contain up to 8 Dial Plans (Dial Plan indices), with a total of up to 8,000 dialing rules (lines) of distinct prefixes (e.g. area codes, international telephone number patterns) for the PSTN to which the device is connected.

The Dial Plan file is created in a textual *ini* file with the following syntax:

```
<called number prefix>,<total digits to wait before sending>
```

- Each new Dial Plan index begins with a Dial Plan name enclosed in square brackets "[...]" on a new line.
- Each line under the Dial Plan index defines a dialing prefix and the number of digits expected to follow that prefix. The prefix is separated by a comma "," from the number of additional digits.
- The prefix can include numerical ranges in the format [x-y], as well as multiple numerical ranges [n-m][x-y] (no comma between them).
- The prefix can include the asterisk "*" and number "#" signs.
- The number of additional digits can include a numerical range in the format x-y.
- Empty lines are ignored.
- Lines beginning with a semicolon ";" are ignored. The semicolon can be used for comments.

Below shows an example of a Dial Plan file (in *ini*-file format), containing two dial plans:

```
; Example of dial-plan configuration.
; This file contains two dial plans:
[ PLAN1 ]
; Destination cellular area codes 052, 054, and 050 with 8 digits.

052,8
054,8
050,8
; Defines International prefixes 00, 012, 014.
; The number following these prefixes may
; be 7 to 14 digits in length.
00,7-14
012,7-14
014,7-14
; Defines emergency number 911. No additional digits are expected.
911,0
[ PLAN2 ]
; Defines area codes 02, 03, 04.
; In these area codes, phone numbers have 7 digits.
0[2-4],7
; Operator services starting with a star: *41, *42, *43.
; No additional digits are expected.
*4[1-3],0
```

The following procedure provides a summary on how to create a Dial Plan file and select the required 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., mydialplans.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. Install the converted file on the device, as described in "Loading Auxiliary Files" on page 599.
5. The required Dial Plan is selected using the 'Dial Plan Index' parameter. This parameter can be set to **0** through **7**, where **0** denotes PLAN1, **1** denotes PLAN2, and so on.

Notes:

- The Dial Plan file must not contain overlapping prefixes. Attempting to process an overlapping configuration by the DConvert utility results in an error message specifying the problematic line.
- The Dial Plan index can be selected globally for all calls (as described in the previous procedure), or per specific calls using Tel Profiles.
- It may be useful to configure both Dial Plan file and Digit Maps. For example, the Digit Map can be used for complex digit patterns (which are not supported by the Dial Plan file) and the Dial Plan can be used for long lists of relatively simple digit patterns. In addition, as timeout between digits is not supported by the Dial Plan, the Digit Map can be used to configure digit patterns that are shorter than those defined in the Dial Plan or left at default (MaxDigits parameter). For example, the “xx.T” digit map instructs the device to use the Dial Plan and if no matching digit pattern is found, it waits for two more digits and then after a timeout (TimeBetweenDigits parameter), it sends the collected digits. Therefore, this ensures that calls are not rejected as a result of their digit pattern not been completed in the Dial Plan.
- By default, if no matching digit pattern is found in both the Dial Plan and Digit Map, the device rejects the call. However, if you set the DisableStrictDialPlan parameter to 1, the device attempts to complete the call using the MaxDigits and TimeBetweenDigits parameters. In such a setup, it collects the number of digits configured by the MaxDigits parameters. If more digits are received, it ignores the settings of this parameter and collects the digits until the inter-digit timeout configured by the TimeBetweenDigits parameter is exceeded.

46.1.4.3 Dial Plan Prefix Tags for Routing

46.1.4.3.1 Dial Plan Prefix Tags for IP-to-Tel Routing

For deployments requiring many IP-to-Tel routing rules that exceed the maximum number of rules that can be configured in the Inbound IP Routing table, you can employ user-defined string labels (tags) to represent the many different prefix calling (source) and called (destination) numbers. The prefix tags are used in the Inbound IP Routing table (see "Configuring Inbound IP Routing Table" on page 406) as source and destination number matching characteristics for the routing rule. Prefix tags are typically implemented when you have calls of many different called or calling numbers that need to be routed to the same destination. Thus, instead of configuring a routing rule for each prefix number, you need to configure only one routing rule using the prefix tag.

For example, this feature is useful in deployments that need to handle hundreds of call routing scenarios such as for a large geographical area (a state in the US). Such an area could consist of hundreds of local area codes as well as codes for international calls. The local calls and international calls would need to be routed to different SIP trunks. Thus, instead of configuring many routing rules for each call destination type, you can simply configure two routing rules, one with a unique prefix tag representing the different local area codes and the other with a prefix tag representing international calls.

Note: When using prefix tags, you need to configure manipulation rules to remove the tags before the device sends the calls to their destinations.

You configure prefix tags in the Dial Plan file, using the following syntax:

```
[ PLAN<index> ]
<prefix number>,0,<prefix tag>
```

where:

- *Index* is the Dial Plan index
- *prefix number* is the called or calling number prefix (ranges can be defined in brackets)
- *prefix tag* is the user-defined prefix tag of up to nine characters, representing the prefix number

Each prefix tag type - called or calling - must be configured in a dedicated Dial Plan index number. For example, Dial Plan 1 can be for called prefix tags and Dial Plan 2 for calling prefix tags.

The example Dial Plan file below defines the prefix tags "LOCL"and "LONG" 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,LONG
425100,0,LONG
....
```

Notes:

- Called and calling prefix tags can be used in the same routing rule.
- Dial Plan Prefix Tags are not applicable to FXS and FXO interfaces.

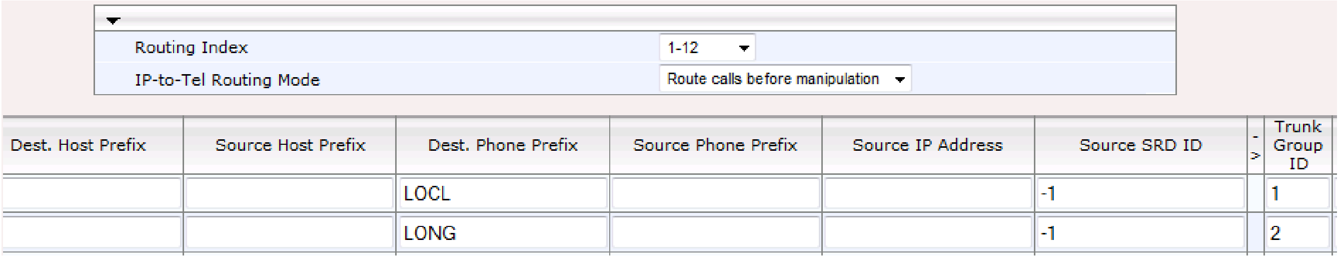
The following procedure describes how to configure IP-to-Tel routing using prefix tags.

➢ To configure IP-to-Tel routing using prefix tags:

1. Configure a Dial Plan file with prefix tags, and then load the file to the device.
2. On the Routing General Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **General Parameters**), specify the Dial Plan indices (e.g., 1) where you configured the prefix tags:
   - Prefix tags for called number prefixes: 'IP-to-Tel Tagging Destination Dial Plan Index' parameter
   - Prefix tags for calling number prefixes: 'IP-to-Tel Tagging Source Dial Plan Index' parameter

3. Open the Inbound IP Routing table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Routing** > **IP to Trunk Group Routing**).
   a. From the 'IP-to-Tel Routing Mode' drop-down list, select **Route calls before manipulation** so that the device performs the routing processing before manipulation.
   b. Configure routing rules using the prefix tags as matching characteristics for destination or source number prefixes:
      - Prefix tags for called number prefixes: 'Dest. Phone Prefix'. For example, configure two routing rules:
        - ✓ Set this field to "LOCL" and the 'Trunk Group ID' field to 1 (local Trunk Group).
        - ✓ Set this field to "LONG" and the 'Trunk Group ID' field to 2 (long distance Trunk Group).
      - Prefix tags for calling number prefixes: 'Source Phone Prefix'.

Figure 46-1: Configuring Dial Plan File Label for IP-to-Tel Routing



Routing Index	1-12
IP-to-Tel Routing Mode	Route calls before manipulation

Dest. Host Prefix	Source Host Prefix	Dest. Phone Prefix	Source Phone Prefix	Source IP Address	Source SRD ID	->	Trunk Group ID
		LOCL			-1		1
		LONG			-1		2

4. Configure manipulation rules to remove the prefix called tags:
   a. Open the Destination Phone Number Manipulation Table for IP-to-Tel Calls table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Dest Number IP->Tel**).
   b. In the 'Destination Prefix' field, enter the prefix called tag (e.g., "LOCL").
   c. In the 'Stripped Digits From Left' field, enter the number of characters in the prefix called tag (e.g., "4").
5. Configure manipulation rules to remove the prefix calling tags:
   a. Open the Source Phone Number Manipulation Table for IP-to-Tel Calls table (**Configuration** tab > **VoIP** menu > **GW and IP to IP** > **Manipulations** > **Source Number IP->Tel**).
   b. In the 'Source Prefix' field, enter the prefix calling tag.
   c. In the 'Stripped Digits From Left' field, enter the number of characters in the prefix calling tag.

46.1.4.3.2 Dial Plan Prefix Tags for SBC IP-to-IP Routing

For deployments requiring many SBC IP-to-IP routing rules that exceed the maximum number of rules that can be configured in the IP-to-IP Routing table, you can employ user-defined string labels (tags) to represent the many different prefix calling (source) and called (destination) numbers. The prefix tags are used in the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 552) as source and destination URI user parts matching characteristics for the routing rule. Prefix tags are typically implemented when you have calls of many different called or calling numbers that need to be routed to the same destination. Thus, instead of configuring a routing rule for each prefix number, you need to configure only one routing rule using the prefix tag.

For example, this feature is useful in deployments that need to handle hundreds of call routing scenarios such as for a large geographical area (a state in the US). Such an area could consist of hundreds of local area codes as well as codes for international calls. The local calls and international calls would need to be routed to different SIP trunks. Thus, instead of configuring many routing rules for each call destination type, you can simply configure two routing rules, one with a unique prefix tag representing the different local area codes and the other with a prefix tag representing international calls.

> **Note:** When using prefix tags, you need to configure manipulation rules to remove the tags before the device sends the calls to their destinations.

You configure prefix tags in the Dial Plan file, using the following syntax:

```
[ PLAN<index> ]
<prefix number>,0,<prefix tag>
```

where:

- *Index* is the Dial Plan index
- *prefix number* is the called or calling number prefix (ranges can be defined in brackets)
- *prefix tag* is the user-defined prefix tag of up to nine characters, representing the prefix number

Each prefix tag type - called or calling - must be configured in a dedicated Dial Plan index number. For example, Dial Plan 1 can be for called prefix tags and Dial Plan 2 for calling prefix tags.

The example Dial Plan file below defines the prefix tags "LOCL"and "INTL" to represent different called number prefixes for local and long distance calls:

```
[ PLAN1 ]
42520[3-5],0,LOCL
425207,0,LOCL
42529,0,LOCL
425200,0,INTL
425100,0,INTL
....
```

> **Note:** Called and calling prefix tags can be used in the same routing rule.

The following procedure describes how to configure IP-to-IP routing using prefix tags.

➢ To configure IP-to-IP routing using prefix tags:

1. Configure a Dial Plan file with prefix tags, and then load the file to the device.
2. Add the prefix tags to the numbers of specific incoming calls using Inbound IP-to-IP Manipulation rules:
   a. Open the IP to IP Inbound Manipulation table (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP-to-IP Inbound**), and then click **Add**.
   b. Click the **Rule** tab, and then configure matching characteristics for the incoming call (e.g., set 'Source IP Group ID' to "1").
   c. From the 'Manipulated URI' drop-down list, select **Source** to add the tag to the calling URI user part, or **Destination** to add the tag to the called URI user part.
   d. Click the **Action** tab, and then enter the Dial Plan index for which you configured your prefix tag, in the 'Prefix to Add' or 'Suffix to Add' fields, using the following syntax: $DialPlan<x>, where *x* is the Dial Plan index (0 to 7). For example, if the called number is 4252000555, the device manipulates it to LOCL4252000555.
3. Add an SBC IP-to-IP routing rule using the prefix tag to represent the different source or destination URI user parts:

   a. Open the IP-to-IP Routing table (**Configuration** tab > **VoIP** menu > **SBC** > **Routing SBC** > **IP-to-IP Routing Table**), and then click **Add**.
   b. Click the **Rule** tab, and then enter the prefix tag in the 'Source Username Prefix' or 'Destination Username Prefix' fields (e.g., "LOCL", without the quotes).
   c. Continue configuring the rule as required.

4. Configure a manipulation rule to remove the prefix tags before the device sends the message to the destination:
   a. Open the IP to IP Outbound Manipulation table (**Configuration** tab > **VoIP** menu > **SBC** > **Manipulations SBC** > **IP-to-IP Outbound**), and then click **Add**.
   b. Click the **Rule** tab, and then configure matching characteristics for the incoming call (e.g., set 'Source IP Group ID' to "1"), including calls with the prefix tag (in the 'Source Username Prefix' or 'Destination Username Prefix' fields, enter the prefix tag to remove).
   c. Click the **Action** tab, and then in the 'Remove from Left' or 'Remove from Right' fields (depending on whether you added the tag at the beginning or end of the URI user part, respectively), enter the number of characters making up the tag.

46.1.4.4 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 Tel-to-IP /IP-to-IP and SBC calls. This enables the mapping of called numbers to IP addresses (in dotted-decimal notation) or FQDNs (up to 15 characters).

➢ **To configure routing to an IP destination based on Dial Plan:**

1. Create the Dial Plan file. The syntax of the Dial Plan index for this feature is as follows:

```
<destination / called prefix number>,0,<IP destination>
```

   Note: The second parameter "0" is not used and ignored.

   An example of a configured Dial Plan (# 6) in the Dial Plan file is shown below:

```
[ PLAN6 ]
200,0,10.33.8.52    ; called prefix 200 is routed to
10.33.8.52
201,0,10.33.8.52
300,0,itsp.com      ; called prefix 300 is routed to itsp.com
```

2. Convert the file to a loadable file and then load it to the device (see "Creating a Dial Plan File" on page 606).
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.
   - Tel-to-IP/IP-to-IP Calls (Gateway application): In the Outbound IP Routing table, do the following:
     a. In the 'Destination Address' field, enter the required Dial Plan index using the following syntax:

        DialPlan<index>

        Where "DialPlan0" denotes [PLAN1] in the Dial Plan file, "DialPlan1" denotes [PLAN2], and so on.

Note: The "DialPlan" string is case-sensitive.

46.1.4.5 Modifying ISDN-to-IP Calling Party Number

The device can use the Dial Plan file to change the Calling Party Number value (source number) of the incoming ISDN call when sending to IP. For this feature, the Dial Plan file supports the following syntax:

<ISDN Calling Party Number>,0,<new calling number>

- The first number contains the calling party number (or its prefix) received in the ISDN call SETUP message. The source number can also be a range, using the syntax [x-y] in the Dial Plan file. This number is used as the display name in the From header of the outgoing INVITE.
- The second number must always be set to "0".
- The third number is a string of up to 12 characters containing the mapped number that is used as the URI user part in the From and Contact headers of the outgoing INVITE.

The Dial Plan index used in the Dial Plan file for this feature is defined by the Tel2IPSourceNumberMappingDialPlanIndex parameter.

An example of such a configuration in the Dial Plan file is shown below:

```
[ PLAN1 ]
; specific received number changed to 04343434181.
0567811181,0,04343434181
; number range that changes to 04343434181.
056788118[2-4],0,04343434181
```

If we take the first Dial Plan rule in the example above (i.e., "0567811181,0,04343434181"), the received Calling Number Party of 0567811181 is changed to 04343434181 and sent to the IP with a SIP INVITE as follows:

```
Via: SIP/2.0/UDP 211.192.160.214:5060;branch=z9hG4bK3157667347
From: <sip:04343434181@kt.co.kr:5060>;tag=de0004b1
To: sip:01066557573@kt.co.kr:5060
Call-ID: 585e60ec@211.192.160.214
CSeq: 1 INVITE
Contact:<sip:04343434181@211.192.160.214:5060;transport=udp>
```

The initial Dial Plan text file must be converted to *.dat file format using the DConvert utility. This is done by clicking the DConvert's **Process Dial Plan File** button. For more information, refer to *DConvert Utility User's Guide*.

You can load this *.dat file to the device using the Web interface (see "Loading Auxiliary Files" on page 599), AcBootP utility, or using the Auto-update mechanism from an external HTTP server.

Notes:

- Tel-to-IP routing is performed on the original source number if the parameter 'Tel to IP Routing Mode' is set to 'Route calls before manipulation'.
- Tel-to-IP routing is performed on the modified source number as defined in the Dial Plan file, if the parameter 'Tel To IP Routing Mode' is set to 'Route calls after manipulation'.
- Source number Tel-to-IP manipulation is performed on the modified source number as defined in the Dial Plan file.

46.1.5 User Information File

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

46.1.5.1 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 to **Enable**.
3. Save this setting to the device with a reset for the setting to take effect.

46.1.5.2 Gateway User Information for PBX Extensions and "Global" Numbers

The GW User Info table contains user information that can be used for the following Gateway-related features:

- **Mapping (Manipulating) PBX Extension Numbers with Global Phone Numbers:** maps PBX extension number, connected to the device, with any "global" phone number (alphanumerical) for the IP side. In this context, the "global" phone number serves as a routing identifier for calls in the "IP world" and the PBX extension uses this mapping to emulate the behavior of an IP phone. This feature is especially useful in scenarios where unique or non-consecutive number translation per PBX is needed. This number manipulation feature supports the following call directions:
    - IP-to-Tel Calls: Maps the called "global" number (in the Request-URI user part) to the PBX extension number. For example, if the device receives an IP call destined for "global" number 638002, it changes this called number to the PBX extension number 402, and then sends the call to the PBX extension on the Tel side.

    Note: If you have configured regular IP-to-Tel manipulation rules (see "Configuring Source/Destination Number Manipulation" on page 373), the device applies these rules before applying the mapping rules of the User Info table.

    - Tel-to-IP Calls: Maps the calling (source) PBX extension to the "global" number. For example, if the device receives a Tel call from PBX extension 402, it changes this calling number to 638002, and then sends call to the IP side with this calling number. In addition to the "global" phone number, the display name (caller ID) configured for the PBX user in the User Info table is used in the SIP From header.

    Note: If you have configured regular Tel-to-IP manipulation rules (see "Configuring Source/Destination Number Manipulation" on page 373), the device applies these rules before applying the mapping rules of the User Info table.

    - IP-to-IP Calls: Maps SIP From (calling number) and To (called number) of IP PBX extension numbers with "global" numbers. For example, if the device receives a call from IP PBX extension number 402 (calling / SIP From) that is destined to IP PBX extension number 403 (called / SIP To), the device changes both these numbers into their "global" numbers 638002 and 638003, respectively.
- **Registering Users:** The device can register each PBX user configured in the User Info table. For each user, the device sends a SIP REGISTER to an external IP-based

Registrar server, using the "global" number in the From/To headers. If authentication is necessary for registration, the device sends the user's username and password, configured in the User Info table, in the SIP MD5 Authorization header.

You can configure up to 500 mapping rules in the GW User Info table. These rules can be configured using any of the following methods:

- Web interface - see "Configuring GW User Info Table in Web Interface" on page 615
- CLI - see Configuring GW User Info Table in CLI on page 616
- Loadable User Info file - see "Configuring GW User Info Table in Loadable Text File" on page 617

Notes:

- To enable user registration, set the following parameters on the Proxy & Registration page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Proxy & Registration**) as shown:
  - √ 'Enable Registration': **Enable** (IsRegisterNeeded is set to 1).
  - √ 'Registration Mode': **Per Endpoint** (AuthenticationMode is set to 0).
- For FXS ports, when the device needs to send a new SIP request with the Authorization header (e.g., after receiving a SIP 401 response), it uses the username and password configured in the Authentication table (see Configuring Authentication per Port on page 477). To use the username and password configured in the User Info file, set the 'Password' parameter to any value other than its default value.

46.1.5.2.1 Configuring GW User Info Table in Web Interface

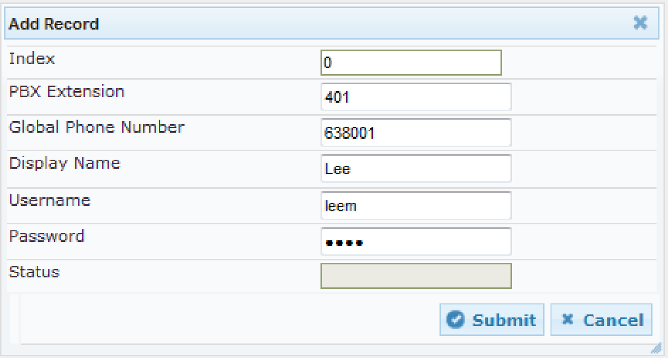
The following procedure describes how to configure and register users in the GW User Info table in the Web interface.

Note: If a User Info file is loaded to the device (as described in "Configuring GW User Info Table in Loadable Text File" on page 617), 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 GW User Info table in the Web interface:**

1. Open the GW User Info Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **User Information** > **GW User Info Table**).
2. Click **Add**: the following dialog box appears:

Figure 46-2: GW User Info Table in Web Interface



Add Record

Index	0
PBX Extension	401
Global Phone Number	638001
Display Name	Lee
Username	leem
Password	••••
Status	

3. Configure the GW User Info table parameters according to the table below.

4. Click **Submit**.
5. To save the changes to flash memory, see "Saving Configuration" **on page 592.**

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 46-2: GW User Info Table Parameter Descriptions

Parameter	Description
Index **[GWUserInfoTable_Index]**	Defines an index for the new table record.
PBX Extension **[GWUserInfoTable_PBXExtension]**	Defines the PBX extension number. The valid value is a string of up to 10 characters.
Global Phone Number **[GWUserInfoTable_GlobalPhoneNumber]**	Defines the "global" phone number for the IP side. The valid value is a string of up to 20 characters.
Display Name **[GWUserInfoTable_DisplayName]**	Defines the Caller ID of the PBX extension. The valid value is a string of up to 30 characters.
Username **[GWUserInfoTable_Username]**	Defines the username for registering the user when authentication is necessary. The valid value is a string of up to 40 characters.
Password **[GWUserInfoTable_Password]**	Defines the password for registering the user when authentication is necessary. The valid value is a string of up to 20 characters.
Status	(Read-only field) Displays the status of the user - "Registered" or "Not Registered".

46.1.5.2.2 Configuring GW User Info Table in CLI

The GW 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 gw-user-info <index, e.g.,
1>
(gw-user-info-1)# username JohnDee
(gw-user-info-1)# <activate | exit>
```

- To delete a specific user, use the no command:

```
(sip-def-proxy-and-reg)# no user-info gw-user-info <index,
e.g., 1>
```

- To view all table entries:

```
(sip-def-proxy-and-reg)# user-info gw-user-info display
---- gw-user-info-0 ----
  pbx-ext (405)
  global-phone-num (405)
  display-name (Ext405)
  username (user405)
  password (0aGzoKfh5uI=)
  status (not-resgistered)
```

- To view a specific entry (example):

```
(sip-def-proxy-and-reg)# user-info gw-user-info <index, e.g.,
0>
(gw-user-info-0)# display
  pbx-ext (405)
  global-phone-num (405)
  display-name (Ext405)
  username (user405)
  password (0aGzoKfh5uI=)
  status (not-resgistered)
```

- To search a user by pbx-ext:

```
(sip-def-proxy-and-reg)# user-info find <pbx-ext e.g., 405>
405: Found at index 0 in GW user info table, not registered
```

46.1.5.2.3 Configuring GW User Info Table in Loadable Text File

The GW 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 599
- *ini* file, using the UserInfoFileName parameter - see "Auxiliary and Configuration File Name Parameters" on page 773
- Automatic Update mechanism, using the UserInfoFileURL parameter - see Automatic Update Mechanism

To add mapping rules to the User Info file, use the following syntax:

```
[ GW ]
FORMAT
PBXExtensionNum,GlobalPhoneNum,DisplayName,UserName,Password
```

Where:

- *[GW]* indicates that this part of the file is the GW User Info table
- *PBXExtensionNum* is the PBX extension number (up to 10 characters)
- *GlobalPhoneNum* is the "global" phone number (up to 20 characters) for the IP side
- *DisplayName* is the Caller ID (string of up to 30 characters) of the PBX extension
- *UserName* is the username (string of up to 40 characters) for registering the user when authentication is necessary
- *Password* is the password (string of up to 20 characters) for registering the user when authentication is necessary

Each line in the file represents a mapping rule of a single PBX extension user.

Notes:

- Make sure that the last line in the User Info file ends with a carriage return (i.e., by pressing the <Enter> key).
- To modify the GW 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:

```
[ GW ]
FORMAT
PBXExtensionNum,GlobalPhoneNum,DisplayName,UserName,Password
401 , 638001 , Mike , miked , 1234
```

```
402 , 638002 , Lee , leem, 4321
403 , 638003 , Sue , suer, 8790
404 , 638004 , John , johnd, 7694
405 , 638005 , Pam , pame, 3928
406 , 638006 , Steve , steveg, 1119
407 , 638007 , Fred , frede, 8142
408 , 638008 , Maggie , maggiea , 9807
```

46.1.5.3 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 500 users (table rows) in the SBC User Info table. The SBC User Info table can be configured using any of the following methods:

- Web interface - see "Configuring SBC User Info Table in Web Interface" on page 618
- CLI - see Configuring SBC User Info Table in CLI on page 619
- Loadable User Info file - see "Configuring SBC User Info Table in Loadable Text File" on page 620

46.1.5.3.1 Configuring SBC User Info Table in Web Interface

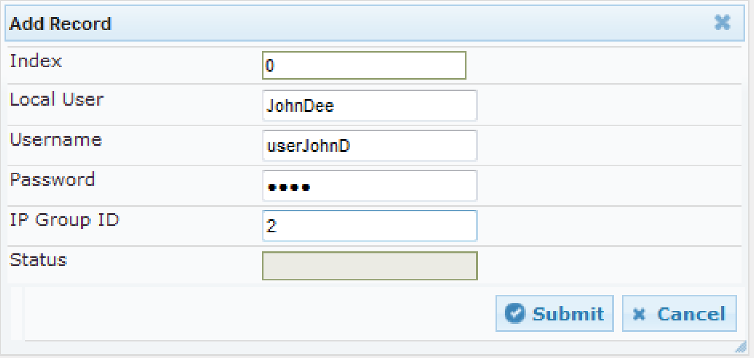
The following procedure describes how to configure the SBC User Info table in the Web interface.

Note: If any User Info file is loaded to the device, all previously configured entries are removed from the table in the Web interface and replaced with the entries from the loaded User Info file.

➢ **To configure the SBC User Info table in the Web interface:**

1. Open the SBC User Info Table page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **User Information** > **SBC User Info Table**).
2. Click **Add**; the following dialog box appears:

Figure 46-3: SBC User Info Table Page



Add Record

Index	0
Local User	JohnDee
Username	userJohnD
Password	••••
IP Group ID	2
Status	

3. Configure the SBC User Info table parameters according to the table below.

4. Click **Submit**.
5. To save the changes to flash memory, see "Saving Configuration" **on page 592.**

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 46-3: SBC User Info Table Parameter Descriptions

Parameter	Description
Index **[SBCUserInfoTable_Index]**	Defines an index for the new table record.
Local User **[SBCUserInfoTable_LocalUser]**	Defines the user and is used as the Request-URI user part for the AOR in the database. The valid value is a string of up to 10 characters.
Username **[SBCUserInfoTable_Username]**	Defines the username for registering the user when authentication is necessary. The valid value is a string of up to 40 characters.
Password **[SBCUserInfoTable_Password]**	Defines the password for registering the user when authentication is necessary. The valid value is a string of up to 20 characters.
IP Group ID **[SBCUserInfoTable_IPGroupID]**	Defines the IP Group ID to which the user belongs and is used as the Request-URI source host part for the AOR in the database.
Status **[SBCUserInfoTable_Status]**	(Read-only field) Displays the status of the user - "Registered" or "Not Registered".

46.1.5.3.2 Configuring SBC User Info Table in CLI

The SBC User Info table can be configured in the CLI using the following commands:

- To add and/or modify a user (example):

```
# configure voip
(config-voip)# sip-definition proxy-and-registration
(sip-def-proxy-and-reg)# user-info sbc-user-info <index, e.g.,
1>
(sbc-user-info-1)# username JohnDee
(sbc-user-info-1)# <activate | exit>
```

- To delete a specific user, use the no command:

```
(sip-def-proxy-and-reg)# no user-info sbc-user-info <index,
e.g., 1>
```

- To view all table entries:

```
(sip-def-proxy-and-reg)# user-info sbc-user-info display
---- sbc-user-info-0 ----
  local-user (JohnDee)
  username (userJohn)
  password (s3fn+fn=)
  ip-group-id (1)
  status (not-resgistered)
---- sbc-user-info-1 ----
 local-user (SuePark)
  username (userSue)
  password (t6sn+un=)
  ip-group-id (1)
  status (not-resgistered)
```

- To view a specific entry (example):

```
(sip-def-proxy-and-reg)# user-info sbc-user-info <index, e.g.,
0>
(sbc-user-info-0)# display
  local-user (JohnDee)
  username (userJohn)
  password (s3fn+fn=)
  ip-group-id (1)
  status (not-resgistered)
```

- To search a user by local-user:

```
(sip-def-proxy-and-reg)# user-info find <local-user, e.g.,
JohnDoe>
JohnDee: Found at index 0 in SBC user info table, not
registered
```

46.1.5.3.3Configuring 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. This User Info file is the same file used for the GW User Info table. Thus, this file can include both Gateway and SBC user information.

You can load the User Info file using any of the following methods:

- Web interface - see "Loading Auxiliary Files" on page 599
- *ini* file, using the UserInfoFileName parameter - see "Auxiliary and Configuration File Name Parameters" on page 773
- Automatic Update mechanism, using the UserInfoFileURL parameter - see Automatic Update Mechanism

To add SBC users to the SBC User Info file, use the following syntax:

```
[ SBC ]
FORMAT LocalUser,UserName,Password,IPGroupID
john,john_user,john_pass,2
sue,sue_user,sue_pass,1
```

where:

- *[SBC]* indicates that this part of the file is the SBC User Info table
- *LocalUser* is the user and is used as the Request-URI user part for the AOR in the database
- *UserName* is the user's authentication username
- *Password* is the user's authentication password
- *IPGroupID* is the IP Group ID to which the user belongs and is used as the Request-URI source host part for the AOR in the database

Note: To modify the SBC User Info table using a User Info file, you need to load to the device a new User Info file containing your modifications.

46.2 Software License Key

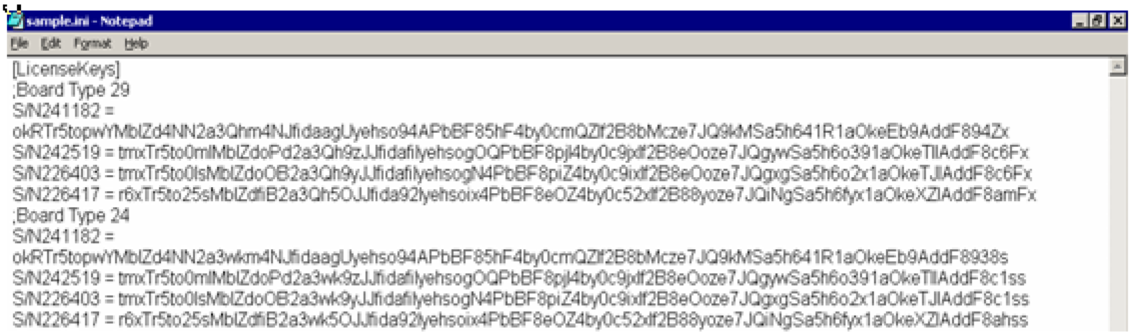
The device is shipped with a pre-installed Software License Key, which determines the device's supported features, capabilities, and available resources. You can upgrade or change your device's supported features by purchasing and installing a new Software License Key to match your requirements.

> **Note:** The availability of certain Web pages depends on the installed Software License Key.

46.2.1 Obtaining the Software License Key File

Before you can install a new Software License Key, you need to obtain a Software License Key file for your device with the required features from your AudioCodes representative. The Software License Key is an encrypted key in string format that is associated with the device's serial number ("S/N") and supplied in a text-based file. If you need a Software License Key for more than one device, the Software License Key file can include multiple Software License Keys (see figure below). In such cases, each Software License Key in the file is associated with a unique serial number identifying the specific device. When loading such a Software License Key file, the device installs only the Software License Key that is associated with its serial number.

Figure 46-4: Software License Key File with Multiple S/N Lines



```
[LicenseKeys]
;Board Type 29
S/N241182 =
okRTr5topwYMblZd4NN2a3Qhm4NJfidaagUyehso94APbBF85hF4by0cmQZlf2B8bMcze7JQ9kMSa5h641R1aOkeEb9AddF894Zx
S/N242519 = tmxTr5to0mlMblZdoPd2a3Qh9zJJfidafilyehsogOQPbBF8pjl4by0c9jxlf2B8eOoze7JQgwSa5h6o391aOkeTllAddF8c6Fx
S/N226403 = tmxTr5to0lsMblZdoOB2a3Qh9yJJfidafilyehsogN4PbBF8piZ4by0c9ixlf2B8eOoze7JQgxgSa5h6o2x1aOkeTJlAddF8c6Fx
S/N226417 = r6xTr5to25sMblZdfiB2a3Qh5OJJfida92lyehsoix4PbBF8eOZ4by0c52xlf2B88yoze7JQiNgSa5h6fyx1aOkeXZlAddF8amFx
;Board Type 24
S/N241182 =
okRTr5topwYMblZd4NN2a3wkm4NJfidaagUyehso94APbBF85hF4by0cmQZlf2B8bMcze7JQ9kMSa5h641R1aOkeEb9AddF8938s
S/N242519 = tmxTr5to0mlMblZdoPd2a3wk9zJJfidafilyehsogOQPbBF8pjl4by0c9jxlf2B8eOoze7JQgwSa5h6o391aOkeTllAddF8c1ss
S/N226403 = tmxTr5to0lsMblZdoOB2a3wk9yJJfidafilyehsogN4PbBF8piZ4by0c9ixlf2B8eOoze7JQgxgSa5h6o2x1aOkeTJlAddF8c1ss
S/N226417 = r6xTr5to25sMblZdfiB2a3wk5OJJfida92lyehsoix4PbBF8eOZ4by0c52xlf2B88yoze7JQiNgSa5h6fyx1aOkeXZlAddF8ahss
```

➢ **To obtain a Software License Key:**

1. Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**) and make a note of the device's MAC address and/or serial number:
   - 'MAC Address' field displays the MAC address.
   - 'Serial Number' field displays the serial number.
2. If you need a Software License Key for more than one device, repeat Step 1 for each device.
3. Send the MAC address and/or serial number to your AudioCodes representative when requesting the required Software License Key.
4. When you receive the new Software License Key file, check the file as follows:
   a. Open the file with any text-based program such as Notepad.
   b. Verify that the first line displays "[LicenseKeys]".

c. Verify that the file contains one or more lines in the following format:

```
"S/N<serial number> = <Software License Key string>"
```

For example: "S/N370604 = jCx6r5tovCIKaBBbhPtT53Yj..."

d. Verify that the "S/N" value reflects the serial number of your device. If you have multiple Software License Keys, ensure that each "S/N" value corresponds to a device.

Warning: Do not modify the contents of the Software License Key file.

5. Install the Software License Key on the device, as described in "Installing the Software License Key" on page 622.

46.2.2 Installing the Software License Key

Once you have received your Software License Key file from your AudioCodes representative, you can install it on the device using one of the following management tools:

- Web interface - see "Installing Software License Key using Web Interface" on page 623
- CLI - see Installing Software License Key using CLI on page 624
- AudioCodes EMS - refer to the EMS User's Manual or EMS Product Description

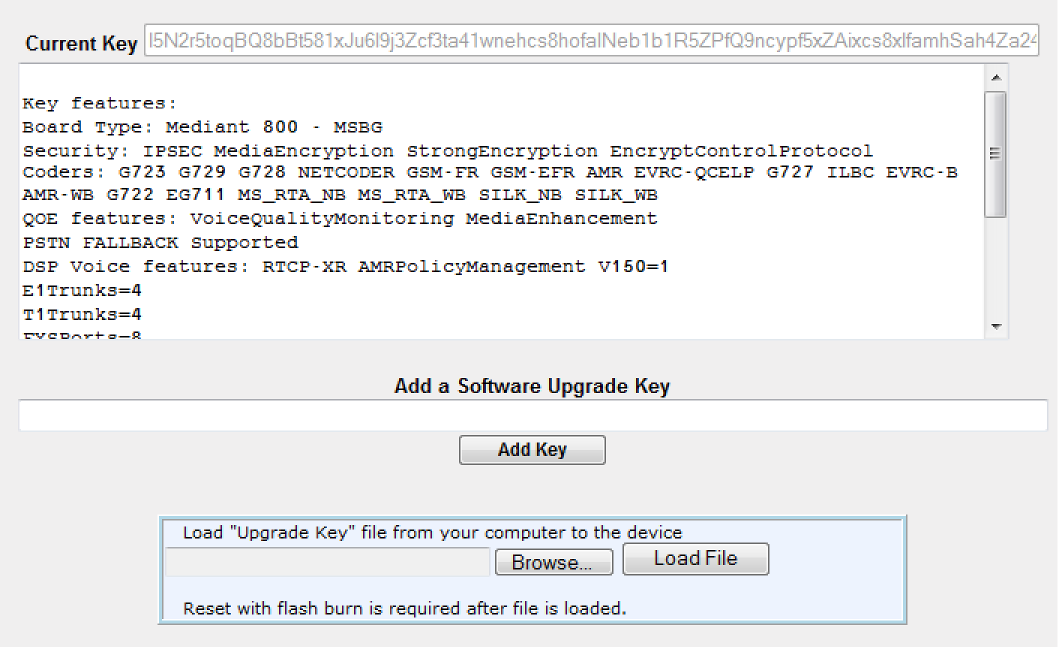
Note: When you install a new Software License Key, it is loaded to the device's non-volatile flash memory and overwrites the previously installed Software License Key.

46.2.2.1 Installing Software License Key using Web Interface

The following procedure describes how to install the Software License Key in the Web interface.

➢ **To install the Software License Key in the Web interface:**

1. Open the Software Upgrade Key Status page (**Maintenance** tab > **Software Update** menu > **Software Upgrade Key**).



Current Key l5N2r5toqBQ8bBt581xJu6l9j3Zcf3ta41wnehcs8hofalNeb1b1R5ZPfQ9ncypf5xZAixcs8xlfamhSah4Za24

```
Key features:
Board Type: Mediant 800 - MSBG
Security: IPSEC MediaEncryption StrongEncryption EncryptControlProtocol
Coders: G723 G729 G728 NETCODER GSM-FR GSM-EFR AMR EVRC-QCELP G727 ILBC EVRC-B
AMR-WB G722 EG711 MS_RTA_NB MS_RTA_WB SILK_NB SILK_WB
QOE features: VoiceQualityMonitoring MediaEnhancement
PSTN FALLBACK Supported
DSP Voice features: RTCP-XR AMRPolicyManagement V150=1
E1Trunks=4
T1Trunks=4
FXSPorts=8
```

Add a Software Upgrade Key
Add Key
Load "Upgrade Key" file from your computer to the device
Browse...
Load File

Reset with flash burn is required after file is loaded.

2. Back up the Software License Key currently installed on the device, as a precaution. If the new Software License Key does not comply with your requirements, you can re-load this backup to restore the device's original capabilities.
   a. In the 'Current Key' field, select the entire text string and copy it to any standard text file (e.g., Notepad).
   b. Save the text file with any file name and file extension (e.g., key.txt) to a folder on your computer.
3. Depending on whether you are loading a Software License Key file with a single Software License Key (i.e., one "S/N") or with multiple Software License Keys (i.e., more than one "S/N"), do one of the following:
   - **Loading a File with a Single Software License Key:**
     a. Open the Software License Key file using a text-based program such as Notepad.
     b. Copy-and-paste the string from the file to the 'Add a Software Upgrade Key' field.
     c. Click the **Add Key** button.
   - **Loading a File with Multiple Software License Keys:**
     a. In the 'Load Upgrade Key file ...' field, click the **Browse** button and navigate to the folder in which the Software License Key file is located on your computer.
     b. Click **Load File**; the new key is installed on the device.

   If the Software License Key is valid, it is burned to the device's flash memory and displayed in the 'Current Key' field.

4. Verify that the Software License Key was successfully installed, by doing one of the following:
   - In the Software Upgrade Key Status page, check that the listed features and capabilities activated by the installed Software License Key match those that were ordered.
   - Access the Syslog server and ensure that the following message appears in the Syslog server:

     "S/N___ Key Was Updated. The Board Needs to be Reloaded with ini file\n"
5. Reset the device; the new capabilities and resources enabled by the Software License Key are active.

> **Note:** If the Syslog server indicates that the Software License Key was unsuccessfully loaded (i.e., the "SN_" line is blank), do the following preliminary troubleshooting procedures:
>
> 1. Open the Software License Key file and check that the "S/N" line appears. If it does not appear, contact AudioCodes.
> 2. Verify that you have loaded the correct file. Open the file and ensure that the first line displays "[LicenseKeys]".
> 3. Verify that the content of the file has not been altered.

46.2.2.2 Installing Software License Key using CLI

To install the Software License Key using CLI, use the following commands:

- To install the Software License Key:

```
(config-system)# feature-key <"string enclosed in double
quotation marks">
```

- To view the Software License Key:

```
show system feature-key
```

46.3 Software Upgrade Wizard

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

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.
- Instead of manually upgrading the device, you can use the device's Automatic Update feature for automatic provisioning (see Automatic Provisioning on page 631).
- You can also upgrade the device's firmware by loading a .cmp file from an external USB hard drive connected to the device's USB port. For more information, see USB Storage Capabilities on page 657.

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>
  ```

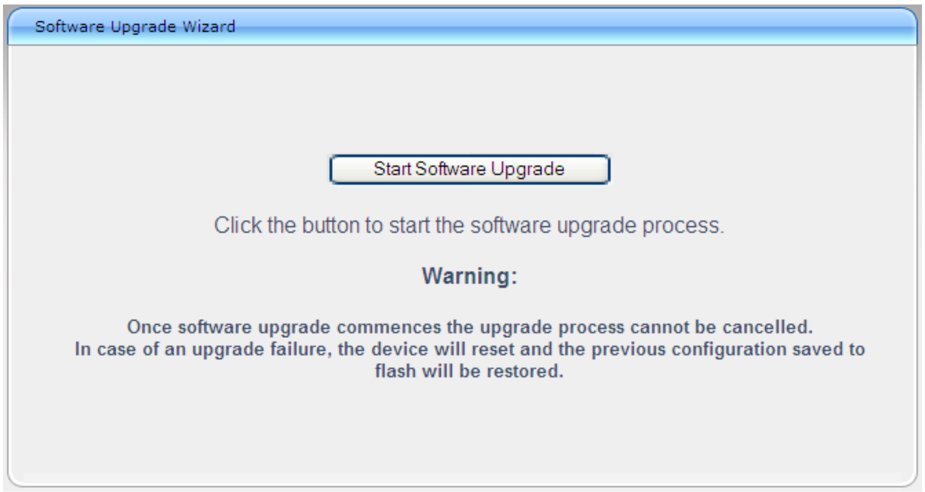
- CLI script file:

  ```
  copy cli-script from <URL>
  ```

➢ To load files using the Software Upgrade Wizard:

1. Make sure that you have installed a new Software License Key (see Software License Key on page 621) 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 591). 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 629).
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 46-5: Start Software Upgrade Wizard Screen



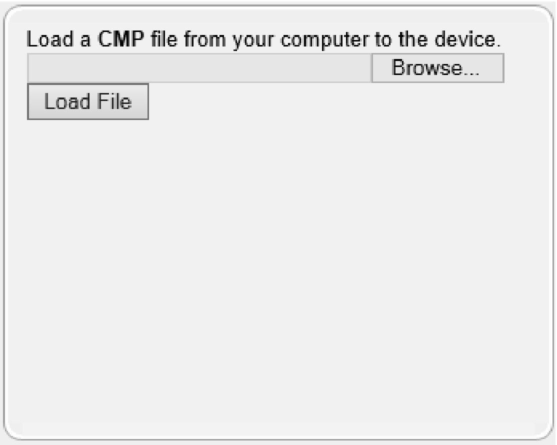
Click the button to start the software upgrade process.

Warning:

Once software upgrade commences the upgrade process cannot be cancelled.
In case of an upgrade failure, the device will reset and the previous configuration saved to flash will be restored.

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

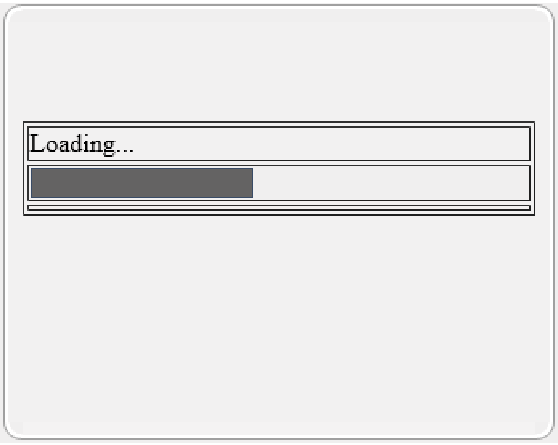
Figure 46-6: Software Upgrade Wizard - Load CMP File


Load a CMP file from your computer to the device.

> **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.

Figure 46-7: Software Upgrade Wizard – CMP File Loading Progress Bar



Loading...

8. 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.

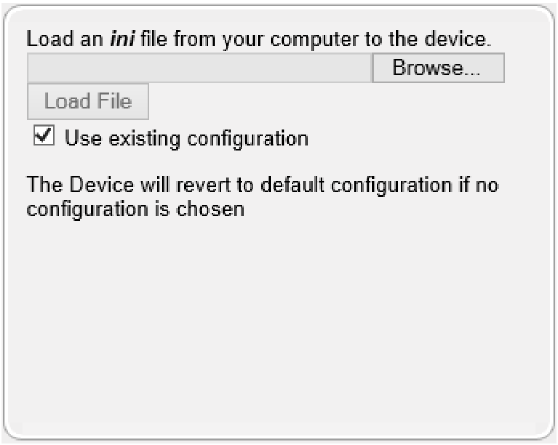
> **Note:** Device reset may take a few minutes (even up to 30 minutes) depending on cmp file version.

9. 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.
10. The wizard page for loading an ini file provides you with the following options:
    - **Load a new ini file:** In the 'Load an ini file...' field, click **Browse**, and then navigate to where the ini file is located on your computer. Select the file, and then click **Load File**; the device loads the ini file.

> **Note:** If you use the wizard to load an ini file, parameters excluded from the ini file are assigned default values (according to the .cmp file running on the device) and thereby, overwrite values previously configured for these parameters.

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

Figure 46-8: Software Upgrade Wizard – Load INI File



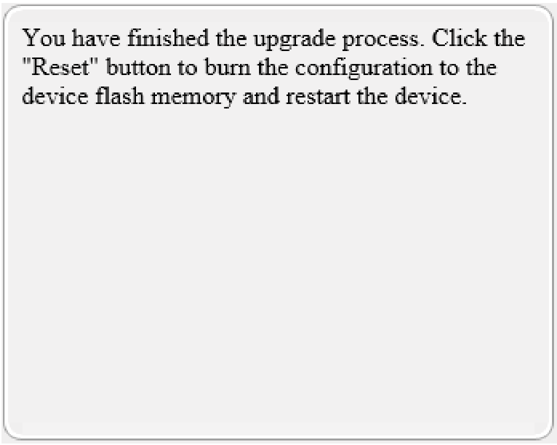
Load an *ini* file from your computer to the device.

☑ Use existing configuration

The Device will revert to default configuration if no configuration is chosen

11. 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 46-9: Software Upgrade Wizard – Files Loaded



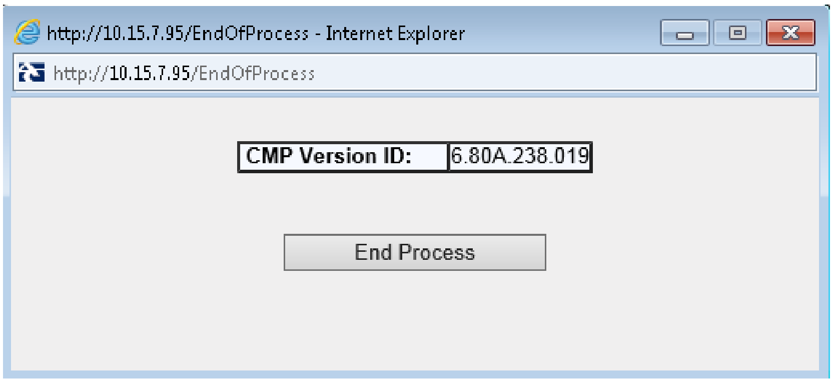
You have finished the upgrade process. Click the "Reset" button to burn the configuration to the device flash memory and restart the device.

12. 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 46-10: Software Upgrade Process Completed Successfully


http://10.15.7.95/EndOfProcess - Internet Explorer
http://10.15.7.95/EndOfProcess
End Process

CMP Version ID:	6.80A.238.019

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

46.4 Backing Up and Loading Configuration File

You can save a copy/backup of the device's current configuration settings as an *ini* file to a folder on your computer, using the Configuration File page. The saved file includes only parameters that were modified and parameters with other than default values. The Configuration File page also allows you to load an *ini* file to the device. If the device has "lost" its configuration, you can restore the device's configuration by loading the previously saved *ini* file or by simply loading a newly created *ini* file.

You can also save the current configuration to a remote server or USB and update configuration from an external USB hard drive connected to the device's USB port. For more information, see USB Storage Capabilities on page 657.

```
# copy cli-script to <URL of TFTP/HTTP/HTTPS server or USB>
```

For example:

- Remote server:

  ```
  # copy cli-script to tftp://192.168.0.3/config-device1.txt
  ```

- USB:

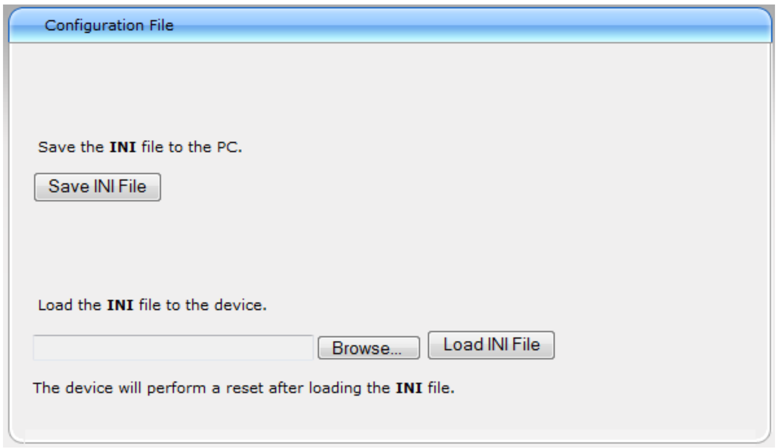
  ```
  # copy cli-script to usb://config-device1.txt
  ```

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 46-11: Configuration File Page



Configuration File

Save the **INI** file to the PC.

Save INI File

Load the **INI** file to the device.

Browse... Load INI File

The device will perform a reset after loading the **INI** file.

2. To save the *ini* file to a folder on your computer:
   a. Click the **Save INI File** button; the File Download dialog box appears.
   b. Click the **Save** button, navigate to the folder where you want to save the file, and then click **Save**.
3. To load the *ini* file to the device:
   a. Click the **Browse** button, navigate to the folder where the file is located, select the file, and then click **Open**; the name and path of the file appear in the field beside the **Browse** button.
   b. Click the **Load INI File** button, and then at the prompt, click **OK**; the device uploads the file and then resets. Once complete, the Web Login screen appears, requesting you to enter your user name and password.

47 Automatic Provisioning Mechanisms

This chapter describes the device's automatic provisioning mechanisms.

47.1 Automatic Configuration Methods

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

Automatic Provisioning Methods

BootP / TFTP	DHCP		Automatic Update Methods				SNMP (EMS)
	67	66	HTTP/S	TFTP	FTP	NFS	
Only Rescue Mode (LAN)	No	No	Yes	Yes	Yes	Yes	Only VoIP configuration

47.1.1 DHCP-based Provisioning

A DHCP server can be configured to automatically provide each device with a temporary IP address so that individual MAC addresses are not required. The DHCP server can provide additional networking parameters such as subnet mask, default gateway, primary and secondary DNS server, and two SIP server addresses. These network parameters have a time limit, after which the device must 'renew' its lease from the DHCP server.

The device can use a host name in the DHCP request. The host name is set to acl_nnnnn, where *nnnnn* denotes the device's serial number. The serial number is the last six digits of the MAC address converted to decimal representation. In networks that support this feature and if the DHCP server registers this host name to a DNS server, you can access the device (through a Web browser) using the URL, http://acl_<serial number> (instead of using the device's IP address). For example, if the device's MAC address is 00908f010280, the DNS name is acl_66176.

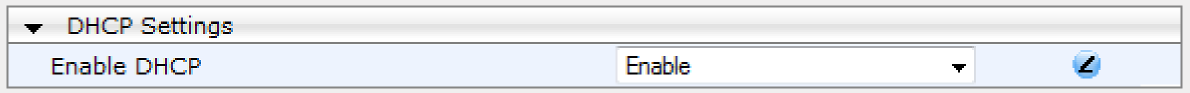
Notes:

- When using DHCP to acquire an IP address, the Interface table, VLANs and other advanced configuration options are disabled.
- For additional DHCP parameters, see DHCP Parameters.

➢ **To enable DHCP:**

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

Figure 47-1: Enabling DHCP - Application Settings Page



DHCP Settings

Enable DHCP: Enable

2. From the 'Enable DHCP" drop-down list, select **Enable**.
3. Click **Submit**.
4. To activate the DHCP process, perform a cold reset by pressing the device's hardware reset button. A soft reset using the Web interface doesn't trigger the DHCP process and if done, this parameter reverts to **Disable**.

The 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, during operation, the device's IP address is changed as a result of a DHCP renewal, the device automatically resets.
- If the DHCP server denies the use of the device's current IP address and specifies a different IP address (according to RFC 1541), the device must change its networking parameters. If this occurs while calls are in progress, they are not automatically rerouted to the new network address. Therefore, administrators are advised to configure DHCP servers to allow renewal of IP addresses.
- If the device's network cable is disconnected and then reconnected, a DHCP renewal is performed (to verify that the device is still connected to the same network). The device also includes its product name in the DHCP Option 60 Vendor Class Identifier. The DHCP server can use this product name to assign an IP address accordingly.
- After power-up, the device performs two distinct DHCP sequences. Only in the second sequence is DHCP Option 60 included. If the device is software reset (e.g., from the Web interface or SNMP), only a single DHCP sequence containing Option 60 is sent.

47.1.2 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 631 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 634.

47.1.3 FTP- or NFS-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 632 is that the protocol in the URL is "ftp" (instead of "http").

Notes:

- Unlike FTP, NFS is not NAT-safe.
- NFS v2/v3 is also supported.

47.1.4 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.

47.2 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. However, the device may also be deployed without any initial configuration and then automatically provisioned by triggering the Automatic Update feature using the Zero Configuration feature, discussed in detail in Zero Configuration on page 646.

Notes:

- For a description of all the Automatic Update parameters, see Automatic Update Parameters or refer to the CLI Reference Guide.
- For additional security, use HTTPS or FTPS. The device supports HTTPS (RFC 2818) and FTPS using the AUTH TLS method <draft-murray-auth-ftp-ssl-16>.

47.2.1 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- the configuration file can be one of the following types, depending on required configuration:
    - ini File: Contains ini file parameters only, which configures only System and VoIP functionalities (not Data-Routing functionality).
    - CLI script files: Contains CLI commands related only to device configuration (not commands such as show, debug or copy). The file can be used for configuring all the device's functionalities (i.e., System, VoIP, and Data Routing). You can use one of the following types of CLI script files, the only difference being the way that they configure the device:
        - CLI Script: The file updates the device's configuration only according to the file's configuration settings. The device's existing configuration settings (not included in the file) are retained. The device does not undergo a reset and therefore, this file typically contains configuration settings that do not require a device reset. If a reset is required, for example, to apply certain settings, you must include the following CLI command (root level) at the end of the file:

          ```
          # reload if-needed
          ```

          The URL of the server where this file is located is configured by the AUPDCliScriptURL ini file parameter or CLI command, configure system > automatic-update > cli-script <URL>.

- ♦ Startup Script: The file updates the device's configuration according to the file's configuration settings and sets all other parameters that are not included in the file to factory defaults. The file causes two device resets in order to apply the settings. Therefore, this file typically contains the Automatic Update settings and other configuration settings that require a device reset.

  The URL of the server where this file is located is configured by the AUPDStartupScriptURL ini file parameter or CLI command, configure system > automatic-update > startup-script <URL>.

Note: You can use any filename extension for the CLI script files.

47.2.2 File Location for Automatic Update

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

The Automatic Update feature is done per file and configured by specifying the file name and URL address of the provisioning server where the file is located. For a description of the parameters used to configure URLs per file, see Automatic Update Parameters. Below are examples for configuring the file names and their URLs for Automatic Update:

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# cli-script https://company.com/cli/<MAC>
(automatic-update)# startup-script
https://company.com/startup/<MAC>
(automatic-update)# voice-configuration
http://www.company.com/configuration.ini
(automatic-update)# call-progress-tones
http://www.company.com/call_progress.dat
(automatic-update)# auto-firmware
http://www.company.com/SIP_F6.80A.008.cmp
```

The URL of the HTTP server where the files are located can include an IPv6 address or a host name (FQDN) which is resolved into an IPv6 address by a DNS server. The IPv6 URL must be enclosed in square brackets:

- URL with host name (FQDN) for DNS resolution into an IPv6 address:

```
http://[FQDN]:<port>/<filename>
```

- URL with IPv6 address:

```
http://[IPv6 address]:<port>/<filename>
```

An example of an IPv6 URL for Automatic Update is shown below:

```
(automatic-update)# firmware
http://[2000::1]:80/F6.80A.222.0070.cmp
```

Note: For configuration files (Startup Script, and CLI Script), the file name in the URL can automatically contain the device's MAC address for enabling the device to download a file unique to the device. For more information, see 'MAC Address Automatically Inserted in Configuration File Name' on page 640.

47.2.3 Triggers for Automatic Update

The Automatic Update feature can be triggered by the following:

- When the device is initially deployed (first-time deployment) in the network. This trigger is referred to as Zero Configuration (see Zero Configuration on page 646).
- 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.

47.2.4 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.

47.2.5 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 636). 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. To configure the certificate, use the CLI command, use-zero-conf-certificate. If set to yes, the device uses the installed "Zero Conf" certificate (pre-provisioned during production); otherwise, it uses the "regular" certificates (used for Web and SIP applications).
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. The header is used for both the Automatic Update and Zero Configuration features. By default, the header includes the device's model name, MAC address, and currently installed software and configuration versions. Based on its own dynamic applications for logic decision making, the provisioning server uses this information to check if it has relevant files available for the device and determines which files must be downloaded (working in conjunction with the HTTP If-Modified-Since header, described further on in this section).

   You can configure the information that is sent in the User-Agent header, using the AupdHttpUserAgent parameter or CLI command, configure system > http-user-agent. The information can include any user-defined string value or the following supported string variable tags (case-sensitive):

   - **<NAME>** - product name, according to the installed Software License Key
   - **<MAC>** - device's MAC address
   - **<VER>** - software version currently installed on the device, e.g., "6.80.200.001"
   - **<CONF>** - configuration version, as configured in the ini file parameter, INIFileVersion or CLI command, configuration-version

   The device automatically populates these tag variables with actual values in the sent header:

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

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

   ```
   Mozilla/4.0 (compatible; AudioCodes; Mediant; 6.80.200.001)
   ```

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), CLI Script, Startup Script 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 (CLI Script, Startup Script) 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 640.

Notes:

- When this method is used, there is typically no need for the provisioning server to check the device’s current firmware version using the HTTP-User-Agent header.
- The Automatic Update feature assumes that the Web server conforms to the HTTP standard. If the Web server ignores the If-Modified-Since header or doesn’t provide the current date and time during the HTTP 200 OK response, the device may reset itself repeatedly. To overcome this problem, modify the update frequency, using the ini file parameter AutoUpdateFrequency or CLI command configure system > automatic update > update-frequency.

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

  Auxiliary Files:

  - ini:

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

  - CLI:

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

Software (.cmp) File:

- ini:

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

- CLI:

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

Notes:

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

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

47.2.6 File Download Sequence

Whenever the Automatic Update feature is triggered (see 'Triggers for Automatic Update' on page 636), the device attempts to download each file from the configured URLs, in the following order:

1. CLI Script file
2. Startup Script file
3. Periodic software file (.cmp) download
4. One-time software file (.cmp) download
5. Auxiliary file(s)

The following files automatically instruct the device to reset:

- Startup Script file
- 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:

- CLI Script file: Use the reload if-needed CLI command

Notes:

- 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.
- If more than one file needs to be updated:
  - √ CLI Script and cmp: The device downloads and applies the CLI Script file on the currently ("old") installed software version. It then downloads and installs the cmp file with a reset. Therefore, the CLI Script file MUST have configuration compatible with the "old" software version.
  - √ Startup Script and cmp: The device downloads both files, resets, applies the new cmp, and then applies the configuration from the Startup Script file on the new software version.
  - √ CLI Script and Startup Script: The device downloads and applies both files; but the Startup Script file overwrites all the configuration of the CLI Script file.

47.2.7 Cyclic Redundancy Check on Downloaded Configuration Files

You can enable the device to perform cyclic redundancy checks (CRC) on downloaded configuration files (Startup Script, or CLI Script) 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.

47.2.8 MAC Address Automatically Inserted in Configuration File Name

You can configure the file name of the configuration file (Startup Script, and CLI Script) in the URL to automatically include the MAC address of the device. As described in 'File

Location for Automatic Update' on page 635, 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 case-sensitive string, "<MAC>" anywhere in the configured file name of the URL, for example:

```
(automatic-update)# cli-script
https://company.com/files/cli_script_<MAC>.txt
(automatic-update)# startup-script
https://company.com/files/startup_<MAC>.txt
```

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, startup_00908F033512.txt. Therefore, you can configure all the devices with the same URL and file name.

47.2.9 Automatic Update Configuration Examples

This section provides a few examples on configuring the Automatic Update feature.

47.2.9.1 Automatic Update for Single Device

This simple example describes how to configure the Automatic Update feature for updating a single device. In this example, the device queries the provisioning server for software, configuration and auxiliary files every 24 hours.

➢ **To set up Automatic Provisioning for single device (example):**

1. Set up an HTTP Web server (e.g., http://www.company.com) and place all the required configuration files on this server.
2. Configure the device with the IP address of the DNS server for resolving the domain name (e.g., http://www.company.com) that is used in the URL of the provisioning server. You configure this in the Interface table:
   - ini File:

     ```
     [ InterfaceTable ]
     FORMAT InterfaceTable_Index =
     InterfaceTable_ApplicationTypes,
     InterfaceTable_InterfaceMode, InterfaceTable_IPAddress,
     InterfaceTable_PrefixLength, InterfaceTable_Gateway,
     InterfaceTable_VlanID, InterfaceTable_InterfaceName,
     InterfaceTable_PrimaryDNSServerIPAddress,
     InterfaceTable_SecondaryDNSServerIPAddress,
     InterfaceTable_UnderlyingDevice;
     InterfaceTable 0 = 6, 10, 10.15.7.95, 16, 10.15.0.1, 1,
     "Voice", 80.179.52.100, 0.0.0.0, "vlan 1";
     [ \InterfaceTable ]
     ```

   - CLI:

     ```
     # configure voip
     (config-voip)# interface network-if 0
     (network-if-0)# primary-dns 80.179.52.100
     ```

3. Configure the device with the following Automatic Update settings:

   c. 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
```

d. 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'
```

e. 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'
```

f. 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.

47.2.9.2 Automatic Update from NFS, FTP and HTTP 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:

- NFS server (Version 2) at 10.13.2.10 for storing the CPT file.
- 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 (Startup Script).
- 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. **CPT file:**

a. Set up an NFS server and copy the CPT file to the directory */usr/shared/public* on the NFS server.

b. Configure the device with the NFS server:

```
[ NFSServers ]
FORMAT NFSServers_Index = NFSServers_HostOrIP,
NFSServers_RootPath, NFSServers_NfsVersion,
NFSServers_AuthType, NFSServers_UID, NFSServers_GID,
NFSServers_VlanType;
NFSServers 0 = "10.31.2.10", "/usr/share/public/", 2, 1, 0,
1, 1;
[ \NFSServers ]
```

c. Configure the device with the URL path of the CPT file:

- ini File:

```
CptFileURL =
'file://10.31.2.10/usr/share/public/usa_tones.dat'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# call-progress-tones
'file://10.31.2.10/usr/share/public/usa_tones.dat'
```

2. **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'
```

3. **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'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# auto-firmware 'http://www.company.com/sw.cmp'
(automatic-update)# startup-script
https://company.com/files/startup_script.txt
```

4. 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 ]
```

5. Configure the device to perform the Automatic Update process daily at 03:00 (3 a.m):

- ini File:

```
AutoUpdateFrequency = '03:00'
```

- CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# update-frequency 03:00
```

47.2.9.3 Automatic Update for Mass Deployment

This example describes how to configure the Automatic Update feature for updating multiple devices (i.e., mass deployment) using an HTTP provisioning server. In this example, all the devices are configured to download the same "master" configuration file. This file serves as the configuration template and instructs the devices which files to download and how often to perform the Automatic Update process. In addition, the master file also instructs each device to download an ini configuration file whose file name contains the MAC address of the device.

The example scenario is as follows:

- All devices download a "master" configuration file that contains the following:
  - Common configuration shared by all device's.
  - Specific configuration that instructs each device to download a specific configuration file based on the device's MAC address, using the special string "<MAC>" in the URL, as described in 'MAC Address Automatically Inserted in Configuration File Name' on page 640.
- 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_startup.txt" with the following settings:
   - Common configuration for all devices:
     - ♦ 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:
     - ♦ CLI:

```
# configure system
(config-system)# automatic update
(automatic-update)# cli-script
https://company.com/files/cli_script_<MAC>.txt
```

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)# cli-script https://company.com/files/master_startup.txt
```

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.

47.3 Zero Configuration

The device's Zero Configuration feature enables automatic, remote configuration of newly deployed, non-configured devices, using AudioCodes HTTPS Redirect server. This feature offers an almost plug-and-play experience for quick-and-easy initial deployment of multiple devices at the end-customer's premises.

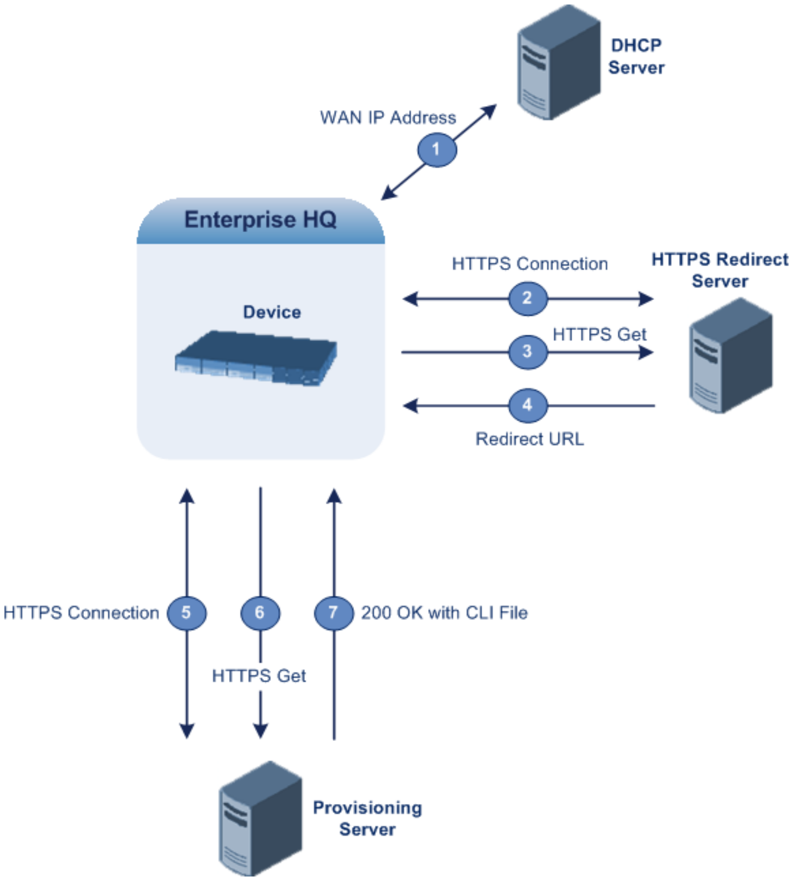
Once the device is powered up and an Internet connection is established, the device activates the Zero Configuration mechanism. The device then connects to the HTTPS Redirect server, which in turn provides the device with the URL of the provisioning server. The device connects to this provisioning server from where it can download software, configuration and/or auxiliary files. Typically, the provisioning server sends the device only a configuration file, which contains settings for the Automatic Update feature only. Once applied to the device and a reset occurs, the Automatic Update mechanism begins (described in 'HTTP/S-Based Provisioning using the Automatic Update Feature' on page 634).

Note: The Zero Configuration feature is available only if the device is installed with 1) a Software License Key enabling this feature and 2) a special "Zero Conf" TLS certificate (TLS Context). If your device was originally purchased and shipped without the Zero Configuration feature and you now wish to have this feature, contact your AudioCodes sales representative on how to order the feature. The procedure typically requires you to return your device as a return merchandise authorization (RMA) so that AudioCodes can install the Software License Key file and Zero Configuration TLS Certificate on your device.

47.3.1 Zero Configuration Process

The Zero Configuration process is summarized below:

Figure 47-2: Zero Configuration Process



1. When you power up the device, the device acquires an IP address for its WAN Ethernet interface from a DHCP server.
2. The device establishes an HTTPS connection with AudioCodes HTTPS Redirect server, using the factory default URL, "redirect.audiocodes.com". For security, communication between the device and the HTTPS Redirect server is encrypted (HTTPS) and setup with mutual authentication. The device uses a special factory-set certificate to authenticate itself with the HTTPS Redirect server and to verify authenticity of the latter. The device is shipped with a factory-configured Zero Configuration certificate (TLS Context):
   - Certificate signed by "Zero Conf" CA.
   - Trusted Storage with the following:
     - Certificate of "Zero Conf" CA
     - Certificates of "well-known" CAs (e.g., VeriSign)

   a. The device verifies the TLS certificate of the Redirect server, using the certificate authority (CA) certificate "Zero Conf", preconfigured on the device during production.

   b. If the Redirect server requests validation of the client certificate, the device provides it.

> **Note:** The certificates issued to both the device and Redirect server have very long validity periods (01/01/2000 to 01/01/2030) and thus, validation verifications succeed even when the device has incorrect time settings.

3. The device sends an HTTPS Get request with its MAC address to the Redirect server.
4. If the Redirect server is configured to service the device (i.e., based on the device's MAC address), it replies with an HTTPS 301/302 Moved Permanently / Found redirect response that contains the URL of the provisioning server were the provisioning files to be downloaded are located; otherwise, it responds with an HTTPS 404 Not Found response.
5. If the device receives a 301/302 redirect response, it updates its time and date (obtained from the X-Timestamp header in the redirect response) and establishes an HTTP/S connection with the new URL (provisioning server). If the redirect URL (where the configuration file is stored) also uses HTTPS, the device can use a regular certificate or the Zero Configuration certificate to authenticate itself and validate the server's certificate if a trusted root certificate (regular) is configured. This is configured by the ini file parameter AupdUseZeroConfCerts, or CLI command configure system > automatic-update > use-zero-conf-certs. If the server requests a client certificate, the device presents its "Zero Conf" certificate (signed by the "Zero Conf" CA).
6. The device sends an HTTPS Get request to the provisioning server. The request contains an HTTP User-Agent header that identifies the device (model, MAC address, and firmware version).
7. The provisioning server sends a 200 OK response with a CLI file for configuring the device. This file can be the CLI Script file or the CLI Startup Script file. The type of file depends on your implementation of Zero Configuration and automatic provisioning, specific to your deployment needs. You can contact your AudioCodes sales representative for an explanation on various design concepts for implementing Zero Configuration. For information on the differences between these two files, see 'Files Provisioned by Automatic Update' on page 634.

   One option is for the provisioning server to send a CLI Startup Script file with the 200 OK response. The file would typically contain only configuration settings for the Automatic Update feature. This would include URLs of provisioning server(s) from where the device can download the software (.cmp file), configuration (CLI Script file), and/or auxiliary files (such as Call Progress Tone file). The device applies the settings of the CLI Startup Script file and restores all other parameters not included in the file to default values. As the Startup Script file initiates a device reset, the Automatic Update mechanism is subsequently activated and the device contacts the provisioning server(s) at the configured URLs for downloading the required files. For more information on the Automatic Update process, see 'HTTP/S-Based Provisioning using the Automatic Update Feature' on page 634.

Upon receipt of a 200 OK response from the provisioning server, the device considers Zero Configuration as complete and does not repeat the process on subsequent reboots (resets or power on-off scenarios). If the device does not receive a 200 OK from the provisioning server, the device repeats the Zero Configuration if the device later reboots.

If at any stage, you restore the device to factory defaults (e.g., by running the write factory CLI command or by pressing the hardware reset push-button), the device repeats the Zero Configuration process after the subsequent reboot.

Note: If the device is configured with multiple WAN interfaces, Zero Configuration is attempted on all configured WAN interfaces, sequentially.

47.3.2 Configuring Zero Configuration

The following procedure describes how to set up Zero Configuration. This is only a typical setup procedure; your specific deployment environment (e.g., provisioning server's capabilities) may require a slightly different setup.

➢ **To set up and activate Zero Configuration:**

1. Configure the Zero Configuration feature on the device:
   a. Establish a CLI session with the device.
   b. Enable the Zero Configuration feature (enabled by default):
   ```
   # configure system
   (config-system># automatic-update
   (automatic-update)# zero-conf on
   ```
   c. Configure the URL of the HTTPS Redirect server (default is https://redirect.audiocodes.com/<MAC address of the device>):
   ```
   # configure system
   (config-system># automatic-update
   (automatic-update)# zero-conf-server <URL>
   ```
2. Set up the HTTPS Redirect server with the following:
   - MAC address of the device(s) that you want to service.
   - Redirect URL of the provisioning server to where you want to redirect the device and from where the device can download the required files.

   The above configuration may be done through a third-party, Web-based management interface or SOAP/XML interface of your choosing, which may be integrated with the Service Provider’s provisioning system. For more information, contact AudioCodes support.
3. Create a CLI-based configuration file containing settings relating **only** to the Automatic Update feature (see 'Automatic Update Configuration Examples' on page 641), and place it on your provisioning server (to where the Redirect server redirects the device).
4. Place the configuration, software (.cmp), and/or auxiliary files on a provisioning server. This can be the same provisioning server as in Step 3 or any other provisioning server(s).

5. Set up a DHCP server for assigning an IP address to the device's WAN Ethernet interface.
6. Power down and then power up the device to trigger Zero Configuration.
7. Cable the device to the WAN network.

47.3.3 Using Zero Configuration with Automatic Update

Zero Configuration is typically used in combination with the Automatic Update feature, described in detail in 'HTTP/S-Based Provisioning using the Automatic Update Feature' on page 634. In such a setup, the Zero Configuration process begins first and only after it completes (successfully or not), does the Automatic Update process begin.

The typical method for using Zero Configuration with Automatic Update is described below. However, your specific deployment architecture may require some adjustments to the method in order to suit your requirements.

1. **Zero Configuration:**
   a. The non-configured device connects to the Redirect server.
   b. The Redirect server redirects the device to the URL of the provisioning server.
   c. The provisioning server holds the Startup Script file, which has been created beforehand. The file contains configuration settings for the Automatic Update mechanism as well as other configuration settings that require a device reset in order for them to take effect. An example of such a Startup Script file is shown below:

```
(config-system)# automatic-update
(automatic-update)# use-zero-conf-certs on
(automatic-update)# auto-firmware
https://www.company.com/device/v.6.80A.227.005.cmp
(automatic-update)# call-progress-tones
https://www.company.com/call_progress.dat
(automatic-update)# cli-script https://company.com/cli/<MAC>
(automatic-update)# startup-script
https://company.com/startup/<MAC>
```

> **Note:** The advantage of using the Startup Script file over the CLI Script file for the initial configuration is that it overwrites **all** existing configuration on the device. Configuration settings not included in the file are set to default settings. Therefore, this provides more control over the device's configuration - the device's complete configuration settings are known, and "forgotten" or unwanted settings are eliminated.

   d. Once connection is established with the provisioning server, the device downloads the Startup Script file (in the 200 OK response from the provisioning server).
   e. Zero Configuration is now considered complete.

2. **Automatic Update:**
   a. Once the Startup Script file has been downloaded, the device applies the Automatic Update settings (and any other settings requiring a device reset) according to the file.
   b. The device resets (caused by the Startup Script file), triggering the Automatic Update mechanism.

c. The device attempts to download the files from the URLs (provisioning servers) according to the Automatic Update settings. The method for connecting to the provisioning server(s) and for determining whether the file(s) must be downloaded is described in 'Querying HTTP Provisioning Server for Updated Files' on page 637. The order of the files downloaded by the device is described in 'File Download Sequence' on page 639.

   Note that as the Startup Script file (above) was already downloaded (see Step 1.d), the file is not downloaded during this initial Automatic Update process. The Startup Script is downloaded only when the Automatic Update process is next triggered and the file on the provisioning server has been subsequently modified (determined by the HTTP If-Modified-Since header).

Notes:

- If the Automatic Update process succeeds, the device repeats the Zero Configuration process **only** if you reset the device to factory defaults. If the Automatic Update process fails, the device repeats the Zero Configuration process at the next device reset or power up.
- You can also use the Zero Configuration TLS Context (certificate) for the connection to the HTTPS provisioning server(s) for the Automatic Update process. This is configured using the CLI command, configure system > automatic-update > use-zero-conf-certs on.

47.4 Automatic Provisioning using USB Flash Drive

The device can be automatically provisioned using an external USB hard drive or flash drive (disk on key) connected to its USB port. In order to do this, you need to create a CLI script file named "ac_autorun.txt" that contains your desired configuration based on CLI commands, and then save it to your USB flash drive. Once you plug the USB flash drive into the device's USB port, the device automatically runs the commands in the "ac_autorun.txt" file, line-by-line similar to a Telnet connection (CLI session).

The CLI script file can contain any type of configuration - system, voice, and/or data-router. This can include, for example, automatic update settings such as URLs from where software and auxiliary files can be downloaded. URLs can also point to the USB flash drive itself, where the files to be downloaded are located. The CLI script file can also include commands that are related to device status and diagnostics such as show and debugging commands. The device provides you with the results (CLI output) of running these commands in a text file named ac_output.txt, which it sends to the USB flash drive upon completion of the process. Therefore, you can also use this tool for fast-and-easy troubleshooting. Note that the ac_output.txt file also includes the CLI output of the configuration commands, providing you feedback on the success or failure of each command.

As the device treats the commands in the ac_autorun.txt file as a regular console input, the CLI script must be written in the same format as if you are in a "live" CLI session with the device, but without the CLI prompts. In other words, you need to provide the following:

- Login credentials for accessing the CLI session and the "privileged" (enabled) mode.
- Full path to each command, including navigation between commands using the exit command for leaving command paths.

An example of a CLI script format in the ac_autorun.txt file is shown below. The example provides basic configuration of the device for the administrator to log on remotely. The configuration sets the WAN Gigabit Ethernet interface IP address to 100.0.10.10 and allows SSH connection from the WAN interface.

```
Admin
Admin
en
Admin
configure data
    interface GigabitEthernet 0/0
    ip address 100.0.10.10 255.255.0.0
    exit
exit
configure system
    cli-terminal
          set ssh on
          set wan-ssh-allow on
    exit
exit
reload now
```

The CLI output of the above example which the device sends to the USB flash drive in the ac_output file is shown below:

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

```
MSBR# configure data
MSBR(config-data)# interface GigabitEthernet 0/0
MSBR(conf-if-GE 0/0)# ip address 100.0.10.10 255.255.0.0
MSBR(conf-if-GE 0/0)# exit
MSBR(config-data)# exit
MSBR# configure system
MSBR(config-system)# cli-terminal
MSBR(cli-terminal)#
activate defaults exit help
history list pwd quit
set
MSBR(cli-terminal)# set ssh on
MSBR(cli-terminal)#
activate defaults exit help
history list pwd quit
set
MSBR(cli-terminal)# set wan-ssh-allow on
Note: Setting this parameter requires a reset.
MSBR(cli-terminal)*# exit
MSBR(config-system)*# exit
MSBR*# write
Writing configuration...done
MSBR*#
```

➢ **To automatically provision the device using a USB flash drive:**

1. Using a basic text-editing program such as Notepad, create a new text file.
2. Type the desired CLI commands in the file.
3. Save the file as "ac_autorun.txt".
4. Copy the file to a USB flash drive. The USB must be formatted to the FAT32 file system.
5. Power up the device.
6. Plug the USB flash drive into the device's USB port, located on the front panel; the device runs the "CLI session".
7. When the STATUS LED starts to flash red, remove the USB flash drive from the USB port. This indicates that the device has finished running the CLI script.
8. If required, view the results (output) of the "CLI session" in the ac_output.txt, which the device sent to the USB flash drive.

48 Restoring Factory Defaults

You can restore the device's configuration to factory defaults using one of the following methods:

- CLI (see "Restoring Defaults using CLI" on page 653)
- Hardware reset pinhole button (see Restoring Defaults using Hardware Reset Button on page 654)
- Loading an empty *ini* file (see "Restoring Defaults using an ini File" on page 654)

48.1 Restoring Defaults using CLI

The device can be restored to factory defaults using CLI, as described in the following procedure.

➢ To restore factory defaults using CLI:

1. Access the CLI:
   a. Connect the RS-232 serial port of the device to the communication port on your computer. For serial cabling, refer to the *Hardware Installation Manual*.
   b. Establish serial communication with the device using a serial communication program (such as HyperTerminal™) with the following communication port settings:
      - **Baud Rate:** 115,200 bps
      - **Data Bits:** 8
      - **Parity:** None
      - **Stop Bits:** 1
      - **Flow Control:** None
2. At the CLI prompt, type the username (default is "Admin" - case sensitive), and then press Enter:
   ```
   # Username: Admin
   ```
3. At the prompt, type the password (default is "Admin" - case sensitive), and then press Enter:
   ```
   # Password: Admin
   ```
4. At the prompt, type the following, and then press Enter:
   ```
   # enable
   ```
5. At the prompt, type the password again, and then press Enter:
   ```
   # Password: Admin
   ```
6. At the prompt, type the following to reset the device to default settings, and then press Enter:
   ```
   # write factory
   ```

48.2 Restoring Defaults using Hardware Reset Button

The device's hardware reset pinhole button can be used to reset the device to default settings.

➢ **To restore default settings using the hardware reset pinhole button:**

- With a paper clip or any other similar pointed object, press and hold down the reset pinhole button, located on the front panel for at least 12 seconds (but no more than 25 seconds).

48.3 Restoring Defaults using an ini File

You can restore the device to factory default settings by loading an empty *ini* file to the device. This is done using the Web interface's Configuration File page (see "Backing Up and Loading Configuration File" on page 629). 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.

49 Automatic Archiving of Configuration File

You can configure the device to automatically save a configuration file each time you modify the device's configuration. The archived file can be saved to a user-defined URL of a remote server (TFTP or HTTP/S), or to a USB storage device attached to the device. The device first saves the configuration to its flash memory and then sends the file to the remote server or USB. The configuration in the archived file is based only on CLI commands. Archiving configuration can be useful, for example, when you need to revert to a previously backed-up configuration (for whatever reason).

To configure configuration-file archiving, use the following CLI command (root level):

- **Archiving to a remote server:**

  ```
  # write-and-backup to <URL path with file name>
  ```

- Archiving to a USB:

  ```
  # write-and-backup to usb:///<file name>
  ```

This page is intentionally left blank.

50 USB Storage Capabilities

The device supports USB storage using an external USB hard drive or flash disk (disk on key) connected to its USB port. The storage capabilities are configured using the CLI and include the following:

- To save network captures to the USB:

  ```
  # debug capture data physical stop usb
  ```

- To update the device's firmware from the USB:

  ```
  # copy firmware from usb:///<cmp file name>
  ```

- To update the device's configuration from the USB:

  ```
  # copy voice-configuration from usb:///<ini configuration file
  name>
  ```

- To save the current configuration to the USB:

  ```
  # copy voice-configuration to usb:///<ini configuration file
  name>
  ```

Note: Only a single USB storage (formatted to FAT/FAT32) operation is supported at any given time.

This page is intentionally left blank.

Part X

Status, Performance Monitoring and Reporting

51 System Status

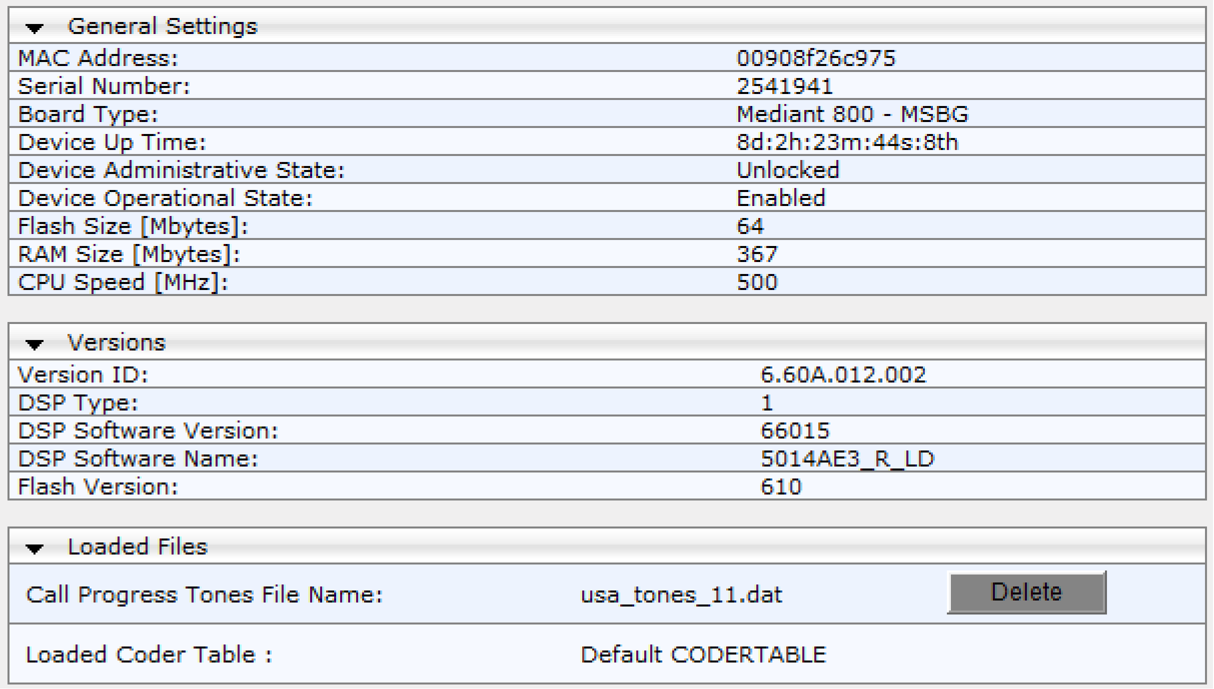
This section describes how to view various system statuses.

51.1 Viewing Device Information

The Device Information page displays hardware and software information about the device. This page also lists any Auxiliary files that have been installed on the device and allows you to remove them.

➢ **To access the Device Information page:**

- Open the Device Information page (**Status & Diagnostics** tab > **System Status** menu > **Device Information**).



▼ General Settings

MAC Address:	00908f26c975
Serial Number:	2541941
Board Type:	Mediant 800 - MSBG
Device Up Time:	8d:2h:23m:44s:8th
Device Administrative State:	Unlocked
Device Operational State:	Enabled
Flash Size [Mbytes]:	64
RAM Size [Mbytes]:	367
CPU Speed [MHz]:	500

▼ Versions

Version ID:	6.60A.012.002
DSP Type:	1
DSP Software Version:	66015
DSP Software Name:	5014AE3_R_LD
Flash Version:	610

▼ Loaded Files

Call Progress Tones File Name:	usa_tones_11.dat	Delete
Loaded Coder Table :	Default CODERTABLE	

➢ **To delete a loaded file:**

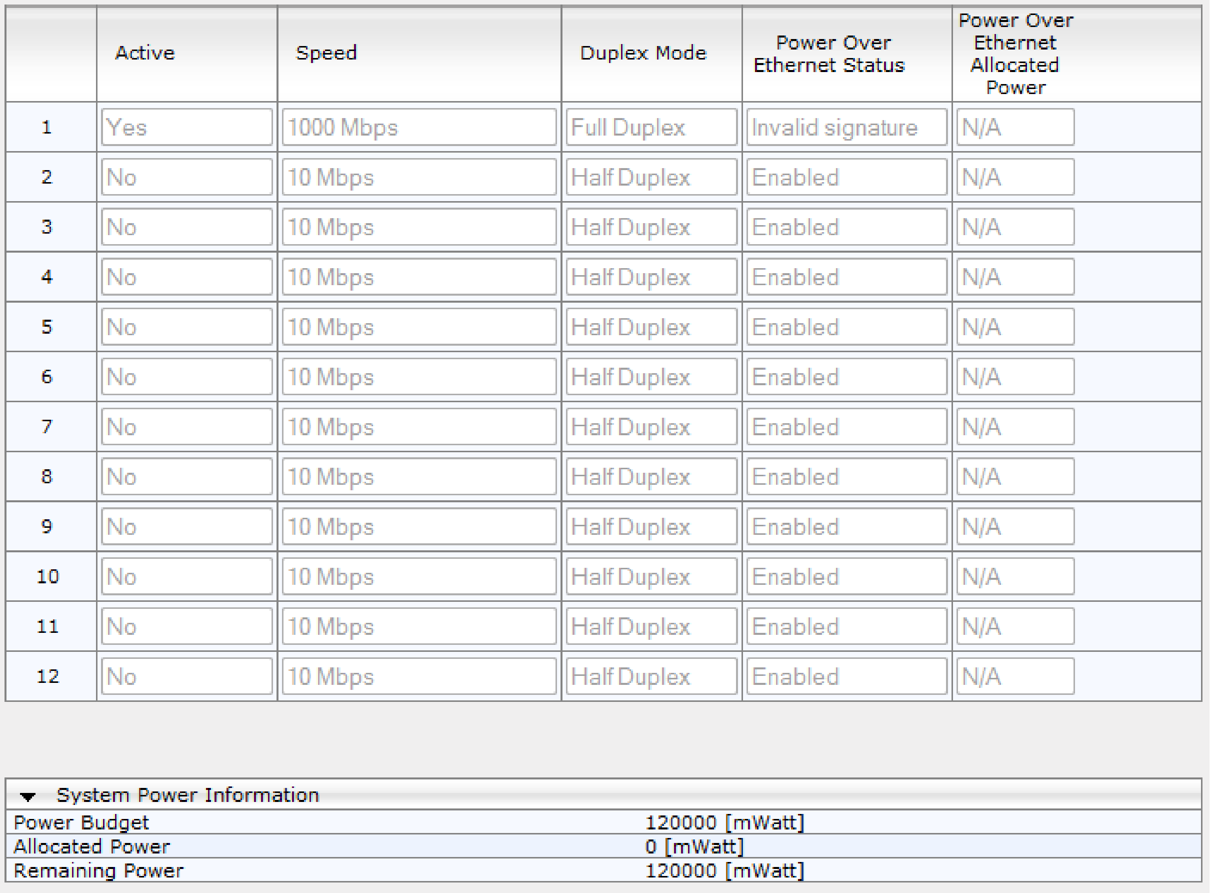
- Click the **Delete** button corresponding to the file that you want to delete. Deleting a file takes effect only after device reset (see "Resetting the Device" on page 589).

51.2 Viewing Ethernet Port Information

The Ethernet Port Information page displays read-only information about the Ethernet Port connections.

➢ **To view Ethernet port information:**

- Open the Ethernet Port Information page:
  - Navigation menu tree: **Status & Diagnostics** tab > **System Status** menu > **Ethernet Port Info**
  - On the Home page, click any Ethernet port on the graphical display of the device (see "Viewing the Home Page" on page 60)



	Active	Speed	Duplex Mode	Power Over Ethernet Status	Power Over Ethernet Allocated Power
1	Yes	1000 Mbps	Full Duplex	Invalid signature	N/A
2	No	10 Mbps	Half Duplex	Enabled	N/A
3	No	10 Mbps	Half Duplex	Enabled	N/A
4	No	10 Mbps	Half Duplex	Enabled	N/A
5	No	10 Mbps	Half Duplex	Enabled	N/A
6	No	10 Mbps	Half Duplex	Enabled	N/A
7	No	10 Mbps	Half Duplex	Enabled	N/A
8	No	10 Mbps	Half Duplex	Enabled	N/A
9	No	10 Mbps	Half Duplex	Enabled	N/A
10	No	10 Mbps	Half Duplex	Enabled	N/A
11	No	10 Mbps	Half Duplex	Enabled	N/A
12	No	10 Mbps	Half Duplex	Enabled	N/A

▼ System Power Information

Power Budget	120000 [mWatt]
Allocated Power	0 [mWatt]
Remaining Power	120000 [mWatt]

Table 51-1: 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

Parameter	Description
Power Over Ethernet Status	Displays whether Power over Ethernet (PoE) is active on the port.
Power Over Ethernet Allocated Power	Displays the power allocated to the port.
Power Budget	Displays the power budget.
Allocated Power	Displays the power allocated to the ports.
Remaining Power	Displays the power available for additional ports.

This page is intentionally left blank.

52 Carrier-Grade Alarms

This section describes how to view the following types of alarms:

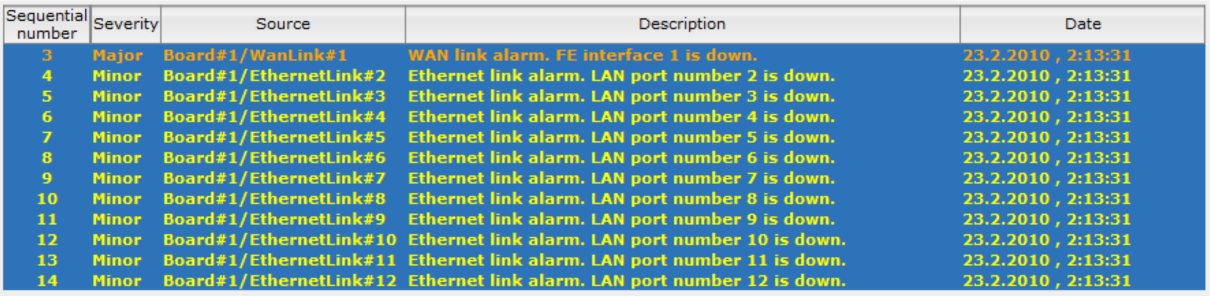
- Active alarms - see "Viewing Active Alarms" on page 665
- Alarm history - see "Viewing Alarm History" on page 666

52.1 Viewing Active Alarms

The Active Alarms page displays a list of currently active alarms. You can also access this page from the Home page (see "Viewing the Home Page" on page 60).

➢ **To view the list of active alarms:**

- Open the Active Alarms page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Active Alarms**).



Sequential number	Severity	Source	Description	Date
3	Major	Board#1/WanLink#1	WAN link alarm. FE interface 1 is down.	23.2.2010 , 2:13:31
4	Minor	Board#1/EthernetLink#2	Ethernet link alarm. LAN port number 2 is down.	23.2.2010 , 2:13:31
5	Minor	Board#1/EthernetLink#3	Ethernet link alarm. LAN port number 3 is down.	23.2.2010 , 2:13:31
6	Minor	Board#1/EthernetLink#4	Ethernet link alarm. LAN port number 4 is down.	23.2.2010 , 2:13:31
7	Minor	Board#1/EthernetLink#5	Ethernet link alarm. LAN port number 5 is down.	23.2.2010 , 2:13:31
8	Minor	Board#1/EthernetLink#6	Ethernet link alarm. LAN port number 6 is down.	23.2.2010 , 2:13:31
9	Minor	Board#1/EthernetLink#7	Ethernet link alarm. LAN port number 7 is down.	23.2.2010 , 2:13:31
10	Minor	Board#1/EthernetLink#8	Ethernet link alarm. LAN port number 8 is down.	23.2.2010 , 2:13:31
11	Minor	Board#1/EthernetLink#9	Ethernet link alarm. LAN port number 9 is down.	23.2.2010 , 2:13:31
12	Minor	Board#1/EthernetLink#10	Ethernet link alarm. LAN port number 10 is down.	23.2.2010 , 2:13:31
13	Minor	Board#1/EthernetLink#11	Ethernet link alarm. LAN port number 11 is down.	23.2.2010 , 2:13:31
14	Minor	Board#1/EthernetLink#12	Ethernet link alarm. LAN port number 12 is down.	23.2.2010 , 2:13:31

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical (red)
  - Major (orange)
  - Minor (yellow)
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

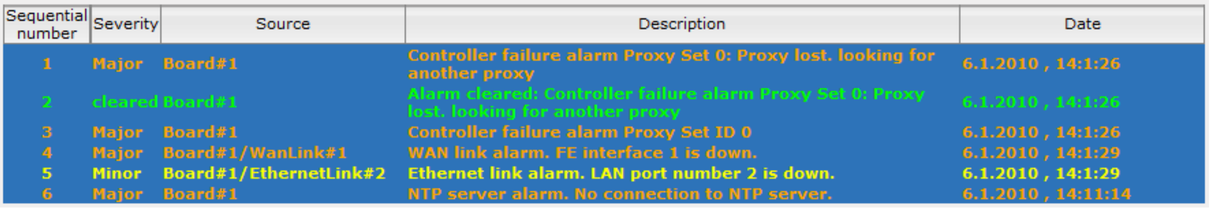
You can view the next 20 alarms (if exist), by clicking the **Go to page** button.

52.2 Viewing Alarm History

The Alarms History page displays a list of alarms that have been raised and traps that have been cleared.

➢ To view the list of history alarms:

- Open the Alarms History page (**Status & Diagnostics** tab > **System Status** menu > **Carrier-Grade Alarms** > **Alarms History**).



Sequential number	Severity	Source	Description	Date
1	Major	Board#1	Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
2	cleared	Board#1	Alarm cleared: Controller failure alarm Proxy Set 0: Proxy lost. looking for another proxy	6.1.2010 , 14:1:26
3	Major	Board#1	Controller failure alarm Proxy Set ID 0	6.1.2010 , 14:1:26
4	Major	Board#1/WanLink#1	WAN link alarm. FE interface 1 is down.	6.1.2010 , 14:1:29
5	Minor	Board#1/EthernetLink#2	Ethernet link alarm. LAN port number 2 is down.	6.1.2010 , 14:1:29
6	Major	Board#1	NTP server alarm. No connection to NTP server.	6.1.2010 , 14:11:14

For each alarm, the following information is provided:

- **Severity:** severity level of the alarm:
  - Critical (red)
  - Major (range)
  - Minor (yellow)
  - Cleared (green)
- **Source:** unit from which the alarm was raised
- **Description:** brief explanation of the alarm
- **Date:** date and time that the alarm was generated

To view the next 20 alarms (if exist), click the **Go to page** button.

➢ To delete all the alarms in the table:

1. Click the **Delete History Table** button; a confirmation message box appears.
2. Click **OK** to confirm.

53 Performance Monitoring

This section describes how to view performance monitoring.

53.1 Viewing MOS per Media Realm

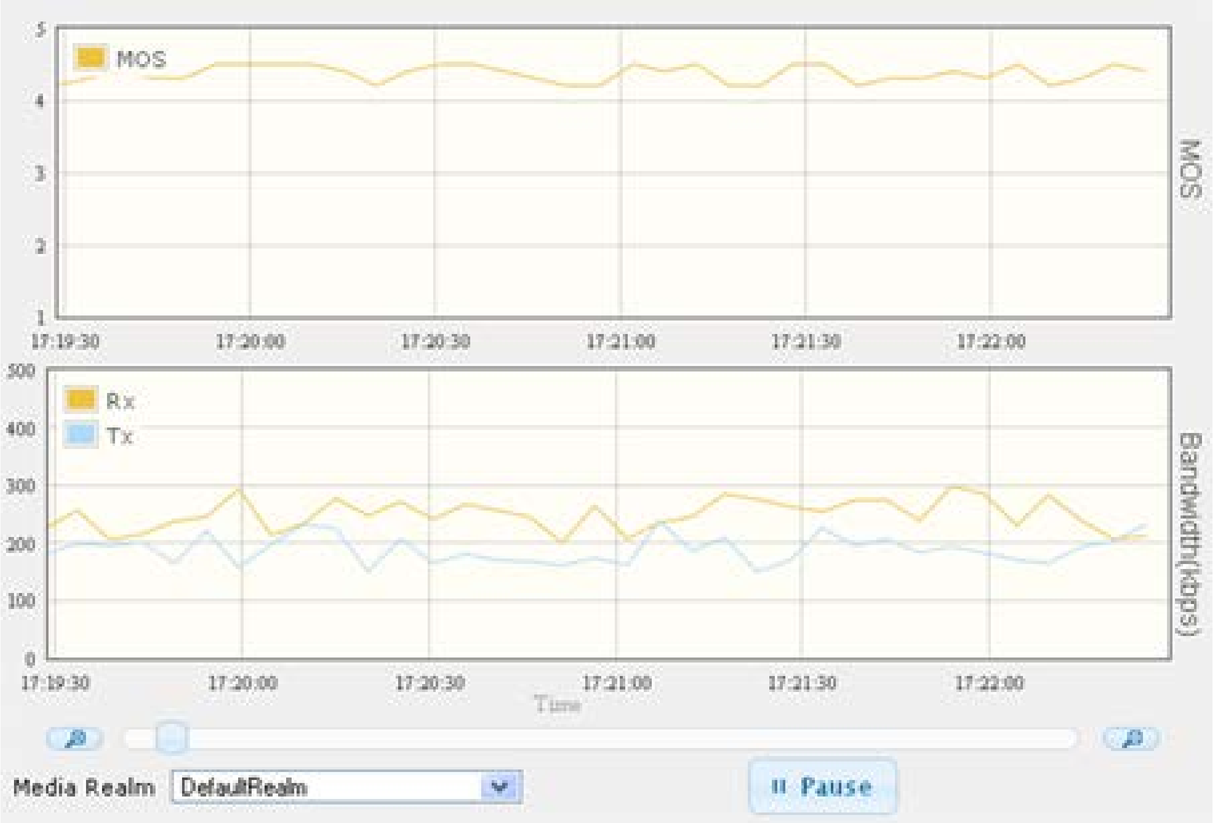
The MOS Per Media Realm page displays statistics on Media Realms (configured in "Configuring Media Realms" on page 259). This page provides two graphs:

- 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 53-1: MOS Per Media Realm Graph



Media Realm DefaultRealm

2. From the 'Media Realm' drop-down list, select the Media Realm for which you want to view.

Use the **Zoom In** 🔍 button to increase the displayed time resolution or the **Zoom Out** 🔍 button to decrease it. Instead of using these zoom buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

To pause the graph, click the **Pause** button; click **Play** to resume.

53.2 Viewing Trunk Utilization

The Trunk Utilization page provides an X-Y graph that displays the number of active channels per trunk over time. The x-axis indicates the time; the y-axis indicates the number of active trunk channels.

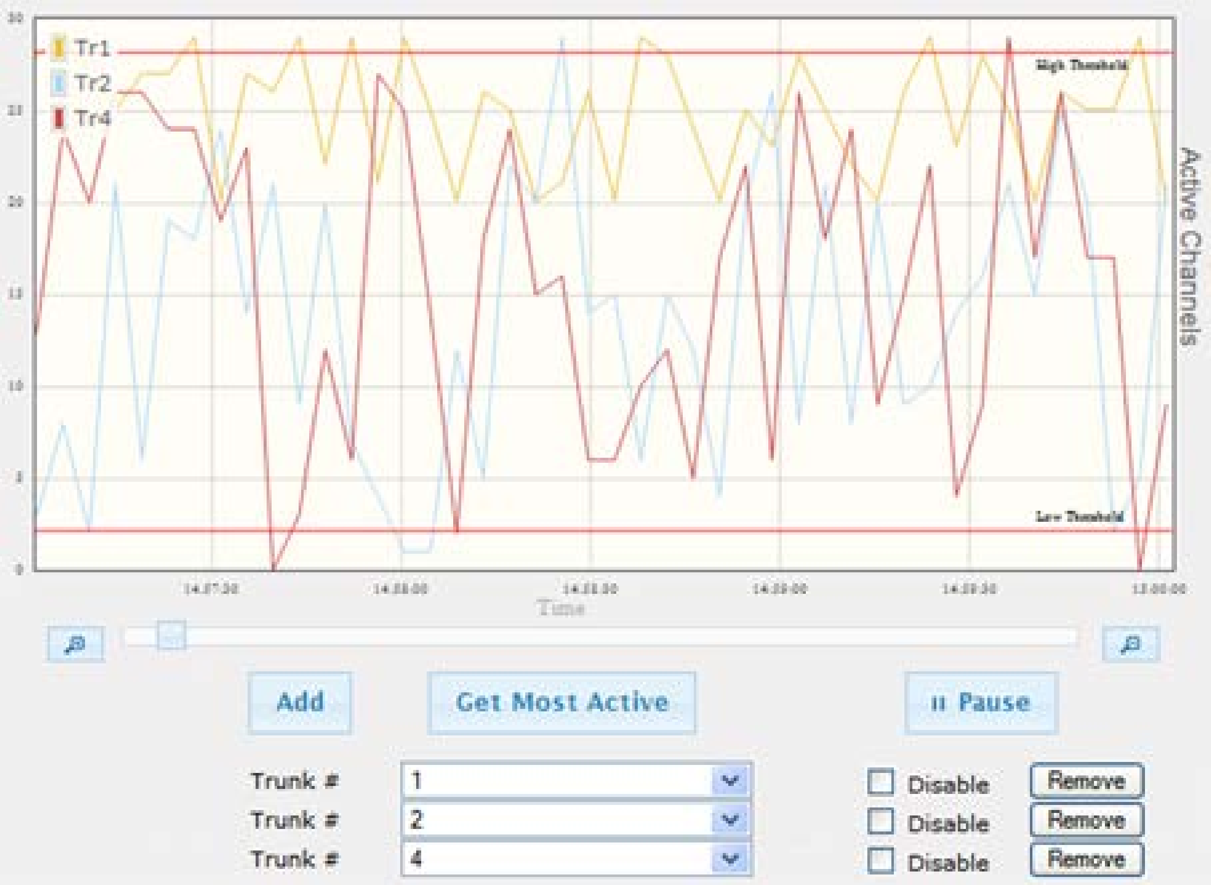
Notes:

- The Trunk Utilization page is available only if your device is equipped with have trunks and the SBC application is disabled.
- If you navigate to a different page, the data displayed in the graph and all its settings are cleared.

➢ To view the number of active trunk channels

1. Open the Trunk Utilization page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **Trunk Utilization**).

Figure 53-2: Trunk Utilization Page



Trunk #	1	☐ Disable	Remove
Trunk #	2	☐ Disable	Remove
Trunk #	4	☐ Disable	Remove

2. From the 'Trunk' drop-down list, select the trunk for which you want to view active channels.

For more graph functionality, see the following table:

Table 53-1: Additional Graph Functionality for Trunk Utilization

Button	Description
Add button	Displays additional trunks in the graph. Up to five trunks can be displayed simultaneously in the graph. To view another trunk, click this button and then from the new 'Trunk' drop-down list, select the required trunk. Each trunk is displayed in a different color, according to the legend shown in the top-left corner of the graph.
Remove button	Removes the selected trunk display from the graph.
Disable check box	Hides or shows an already selected trunk. Select this check box to temporarily hide the trunk display; clear this check box to show the trunk. This is useful if you do not want to remove the trunk entirely (using the **Remove** button).
Get Most Active button	Displays only the trunk with the most active channels (i.e., trunk with the most calls).
Pause button	Pauses the display in the graph.
Play button	Resumes the display in the graph.
Zoom slide ruler and buttons	Increases or reduces the trunk utilization display resolution concerning time. The **Zoom In** 🔍 button increases the time resolution; the **Zoom Out** 🔍 button decreases it. Instead of using the 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.

53.3 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).

Note: The Quality Of Experience page is applicable only to SBC calls.

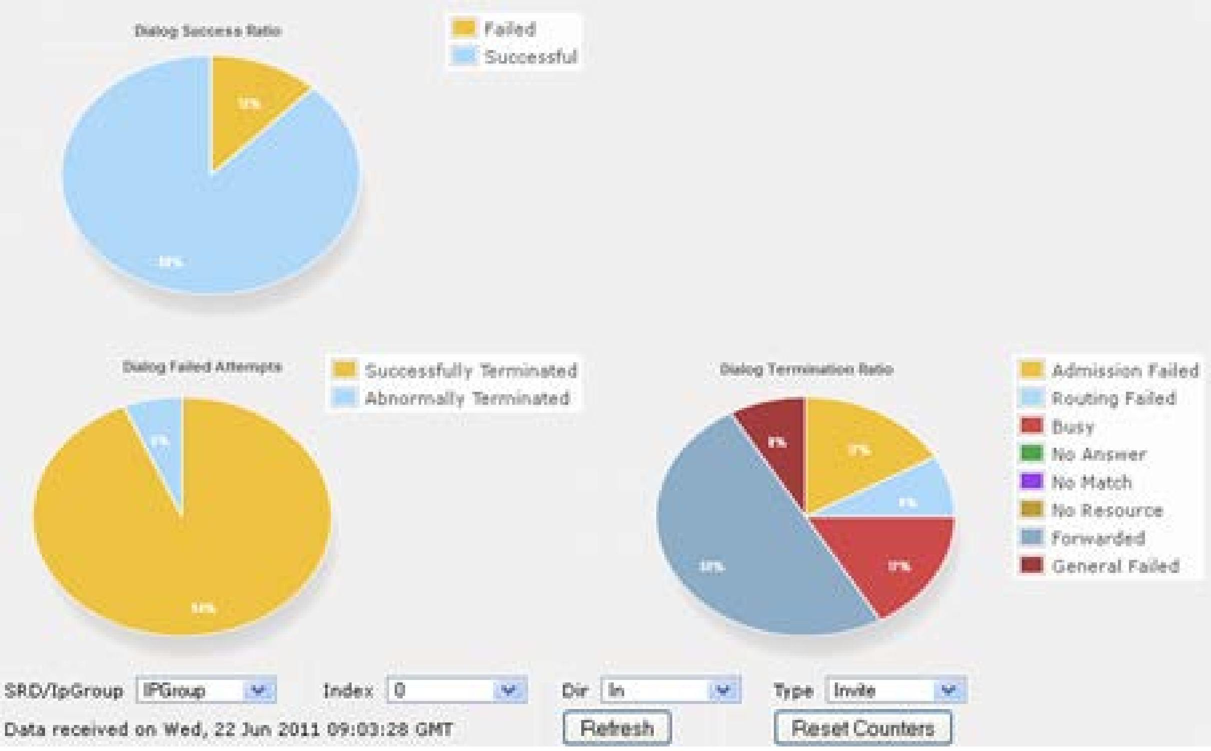
This page provides three pie charts:

- 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 53-3: Quality Of Experience Graph



SRD/IpGroup: IPGroup | Index: 0 | Dir: In | Type: Invite

Data received on Wed, 22 Jun 2011 09:03:28 GMT

2. From the 'SRD/IpGroup' drop-down list, select whether you want to view QoE for an SRD or IP Group.
3. From the 'Index' drop-down list, select the SRD or IP Group index.
4. From the 'Dir' drop-down list, select the call direction:
   - **In -** incoming calls
   - **Out** - outgoing calls
   - **Both -** incoming and outgoing calls
5. From the 'Type' drop-down list, select the SIP message type:
   - **Invite** - INVITE
   - **Subscribe** - SUBSCRIBE
   - **Other** - all SIP messages

To refresh the charts, click **Refresh**. To reset the counters, click **Reset Counters**.

53.4 Viewing Average Call Duration

The Average Call Duration page displays information about a specific SRD or IP Group. This page includes two graphs:

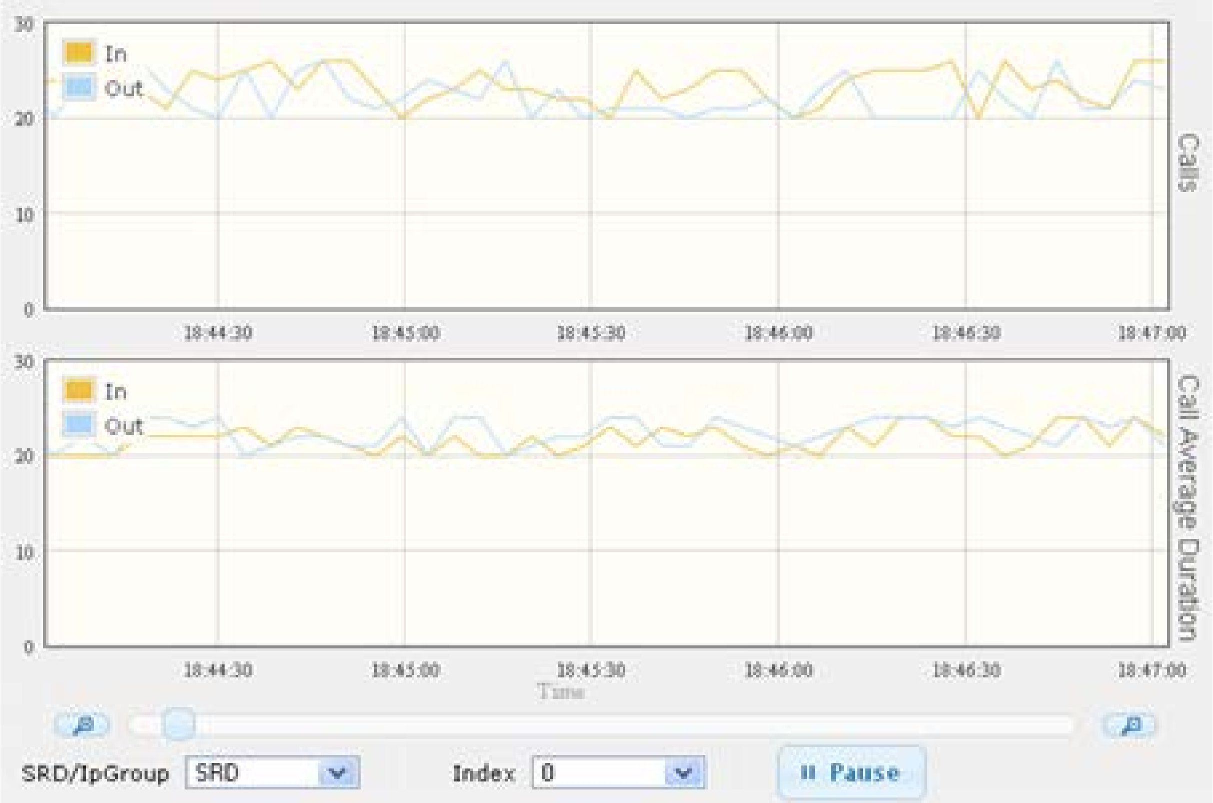
- Upper graph: displays the number of calls (INVITEs).
- Lower graph: displays the average call duration.

Note: The Quality Of Experience page is applicable only to SBC calls.

➢ **To view average call duration:**

1. Open the Average Call Duration page (**Status & Diagnostics** tab > **Performance Monitoring** menu > **Average Call Duration**).

Figure 53-4: Average Call Duration Graph



SRD/IpGroup SRD

Index 0

2. From the 'SRD/IpGroup' drop-down list, select whether you want to view information for an SRD or IP Group.
3. From the 'Index' drop-down list, select the SRD or IP Group index.

Use the **Zoom In** 🔍 button to increase the displayed time resolution or the **Zoom Out** 🔍 button to decrease it. Instead of using these zoom buttons, you can use the slide ruler. As you increase the resolution, more data is displayed on the graph. The minimum resolution is about 30 seconds; the maximum resolution is about an hour.

To pause the graph, click the **Pause** button; click **Play** to resume.

53.5 Network Monitoring (Probing) Two Devices

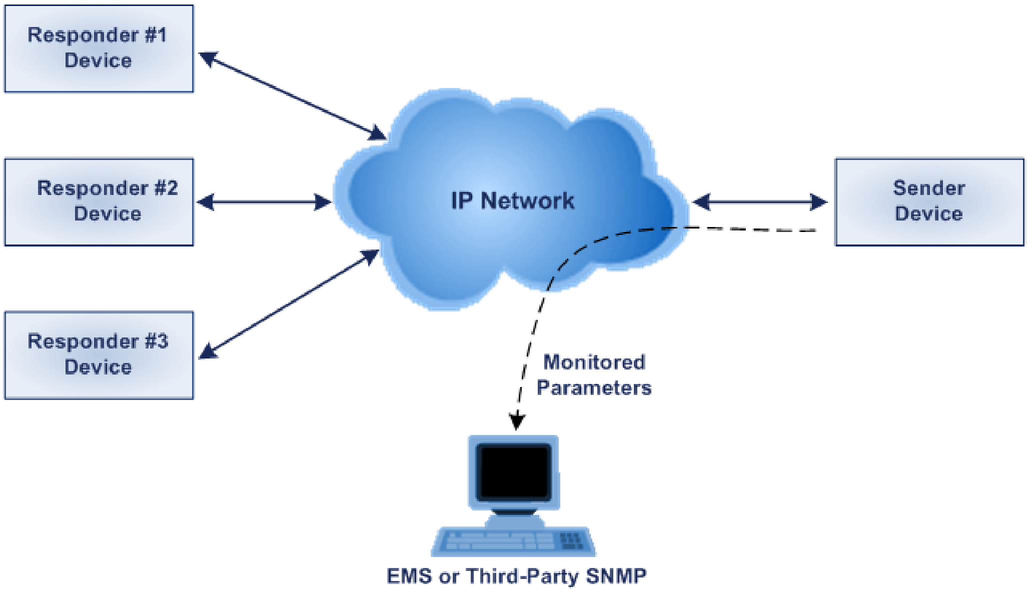
The device can be configured to monitor the quality of the network path (network quality monitoring - NQM) between it and other AudioCodes devices. The path monitoring is done by sending packets from a "sender" device to a "responder" device and then calculating the round-trip time (RTT), packet loss (PL), and jitter. Since both responder and sender nodes are AudioCodes devices, the monitoring is done by sending RTP/RTCP packets in a way that accurately predicts the WAN service-level agreement (SLA) granted for real VoIP calls by the network.

Note: If the packets sent mimic a G.711 or G.729 stream, the following quality measurements are also done:

- Listener quality MOS per ITU-T specification
- Conversation quality MOS per ITU-T specification

You can configure up to 10 network quality probing paths. For example, you can configure three probing paths, where your device is configured as a sender for three different responder devices, as shown in the figure below:

Figure 53-5: Network Quality Probing Example



You can periodically poll the device for the latest VoIP quality metrics and specify thresholds for the quality metrics mentioned above. If these thresholds are crossed, the device generates the following SNMP traps to Audiocodes EMS or third-party SNMP-based manager:

- NqmConnectivityAlarm: Connectivity with monitored probe destination is lost
- NqmRttAlarm: High RTP detected toward probe destination
- NqmJitterAlarm: High jitter detected toward probe destination
- NqmPacketLossAlarm: High packet loss detected toward probe destination

This feature is configured using the CLI. Below is a list of some of the CLI commands. For a full list and description of the related CLI commands, refer to the *CLI Reference Guide*.

- nqm responder-table – adds a responder (IP address and port)
- nqm probing-table – defines the polling attributes (duration and frequency)
- nqm sender-table - adds a sender (including RTT, PL, and jitter thresholds; associates probing definition; responder address; local interface)

The following procedure describes how to quickly configure network quality probing.

➢ To configure network quality probing:

1. Configure the "sender termination" side:

   a. Bind a WAN interface to the NQM service:

```
(config-system) bind GigabitEthernet 0/0 nqm
```

Note: The chosen WAN interface should be the interface on which the NQM packets are planned to flow bi-directionally and binding is necessary to create the corresponding static NAT rules. If the NQM session is planned to flow within the LAN then no binding is needed and this step can be skipped.

   b. Configure a row in the Probing table:

```
(config-system)# nqm probing-table 0
(probing-table-0)# set probe-name voip_probe_1     ;
identifies this line
(probing-table-0)# set start-time now  ; starting time of
this probe
(probing-table-0)# exit    ; activates the probe
```

   c. Configure a row in the Sender table to define a sender termination:

```
(config-system)# nqm sender-table 0
(sender-table-0)# set sender-name
main_office_voip_checker_1 ; identifies specific sender
(sender-table-0)# set target-ip 10.4.3.98    ; IP address
of responder termination
(sender-table-0)# set target-port 3900 ; listening port
number at responder termination
```

Note: A responder termination defined by the pair <target IP address, target port> can be defined only once for a single sender line; multipile senders can’t be defined to send packets to the same responder termination.

```
(sender-table-0)# set probe-name voip_probe_1      ; name
of probing row previously configured to be used by this
sender
```

Note: A single row in the Probing table may be shared by several senders, thereby sharing and simplifying common attributes.

```
(sender-table-0)# set source-interface-name NQM_WAN      ;
name of network interface to send packets from
```

Note: If you want to output packets to the WAN interface, simply set NQM_WAN as the source interface name; otherwise, set the interface name to a specific interface name listed in the network interface table.

```
(sender-table-0)# exit     ; activates the sender line
```

2. Configure the "responder termination" side:
   a. Bind a WAN interface to the NQM service:

```
(config-system) bind GigabitEthernet 0/0 nqm
```

Note: The chosen WAN interface should be the interface on which the NQM packets are planned to flow bi-directionally and binding is necessary to create the corresponding static NAT rules. If the NQM session is planned to flow within the LAN then no binding is needed and this step can be skipped.

   b. Configure a row in the Responder table:

```
(config-system)# nqm responder-table 0
(responder-table-0)# set responder-name
vmain_office_voip_responder_1    ; name tag to identify
this line
(responder-table-0)# set local-port 3900     ; listening
port number at responder termination
(responder-table-0)# exit  ; activates the probe line
```

Note: Ensure that the local-port value is the same as the target-port value set for the corresponding sender termination.

```
(responder-table-0)# set source-interface-name NQM_WAN   ;
name of network interface to send packets from
```

Notes:

- If you want to listen to the WAN interface, simply set NQM_WAN as the source interface name; otherwise, set the interface name to a specific interface name listed in the network interface table.
- Ensure that the network interface the responder termination is listening upon is in-sync with the target-ip value set for the corresponding sender termination.

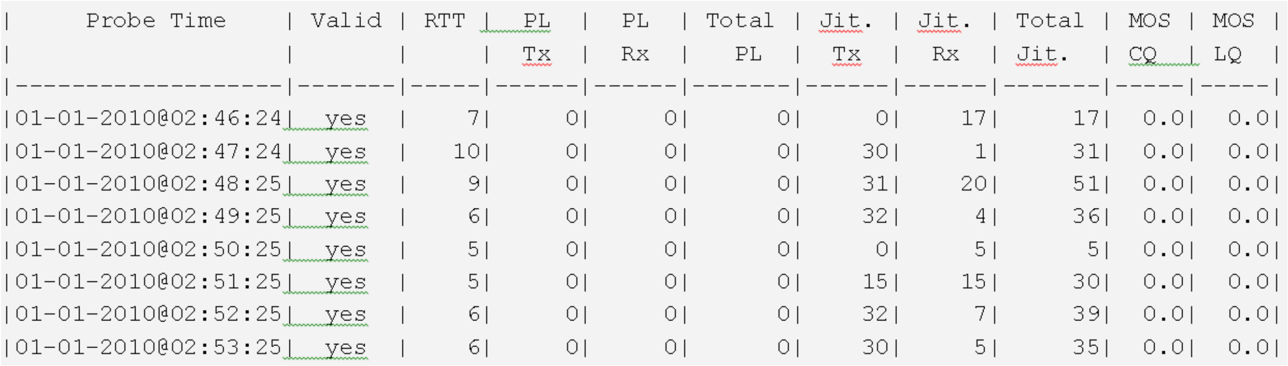
```
(responder-table-0)# exit  ; activates the responder line
```

➢ To view NQM results:

- On the sender termination device, type the following command to view eight result rows of sender "0":

```
# show system nqm 0 8
```

Figure 53-6: Example of NQM Resullts



```
|     Probe Time    | Valid | RTT |  PL  |  PL  | Total | Jit. | Jit. | Total | MOS | MOS |
|                   |       |     |  Tx  |  Rx  |  PL   |  Tx  |  Rx  | Jit.  | CQ  | LQ  |
|-------------------|-------|-----|------|------|-------|------|------|-------|-----|-----|
|01-01-2010@02:46:24|  yes  |    7|     0|     0|      0|     0|    17|     17|  0.0|  0.0|
|01-01-2010@02:47:24|  yes  |   10|     0|     0|      0|    30|     1|     31|  0.0|  0.0|
|01-01-2010@02:48:25|  yes  |    9|     0|     0|      0|    31|    20|     51|  0.0|  0.0|
|01-01-2010@02:49:25|  yes  |    6|     0|     0|      0|    32|     4|     36|  0.0|  0.0|
|01-01-2010@02:50:25|  yes  |    5|     0|     0|      0|     0|     5|      5|  0.0|  0.0|
|01-01-2010@02:51:25|  yes  |    5|     0|     0|      0|    15|    15|     30|  0.0|  0.0|
|01-01-2010@02:52:25|  yes  |    6|     0|     0|      0|    32|     7|     39|  0.0|  0.0|
|01-01-2010@02:53:25|  yes  |    6|     0|     0|      0|    30|     5|     35|  0.0|  0.0|
```

This page is intentionally left blank.

54 VoIP Status

This section describes how to view VoIP status and statistics.

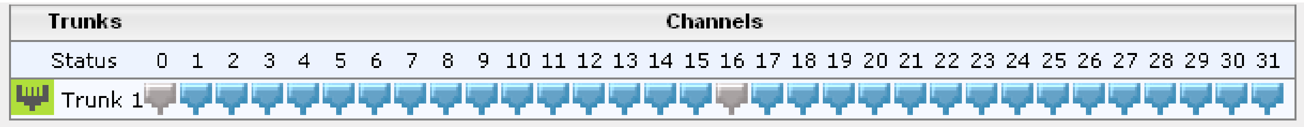
54.1 Viewing Trunks & Channels Status

The Trunks & Channels Status page displays the status of the device's trunks and corresponding channels. It also enables you to view trunk configuration and channel information.

➢ **To view the status of the device's trunks and channels:**

1. Open the Home page.
2. On the graphical display of the device, click the required trunk, and then from the shortcut menu, choose Port Settings; the Trunks & Channels Status page appears.

Figure 54-1: Trunks and Channels Status Screen



Trunks Channels

Status 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

Trunk 1

Note: The number of displayed trunks and channels depends on configuration.

The status of the trunks is depicted by color-coded icons, as described in the table below:

Table 54-1: Description of Color-Coded Icons for Trunk Status

Icon	Color	Trunk
		Label
⬜	Gray	**Disabled**
🟩	Green	**Active - OK**
🟨	Yellow	**RAI Alarm**
🟥	Red	**LOS / LOF Alarm**
🟦	Blue	**AIS Alarm**
🟧	Light Orange	**D-Channel Alarm**
🟧	Dark Orange	**NFAS Alarm**
🟪	Purple	**Lower Layer Down (DS3 physical layer is disabled)**

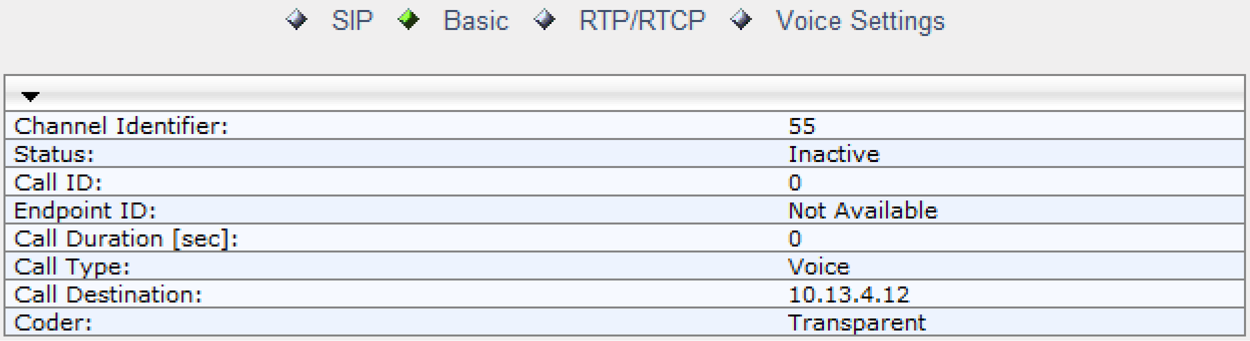
The status of the channels is depicted by color-coded icons, as described in the table below:

Table 54-2: Description of Color-Coded Icons for Channel Status

Icon	Color	Label	Description
🟦	Light blue	**Inactive**	Channel is configured, but currently has no calls
🟩	Green	**Active**	Call in progress (RTP traffic) and no alarms
🟪	Purple	**SS7**	Channel is configured for SS7 **Note:** Currently, SS7 is not supported.
⬜	Gray	**Non Voice**	Channel is not configured
🟦	Blue	**ISDN Signaling**	Channel is configured as a D-channel
🟨	Yellow	**CAS Blocked**	-
🟧	Dark Orange	**Maintenance**	B-channel has been intentionally taken out of service due to maintenance
🟥	Red	**Out Of Service**	B-channel is out of service

3. To view detailed information on a specific trunk's channel, click the required channel icon; the Basic Channel Information page appears, displaying information under the **Basic** tab (displayed in green):

Figure 54-2: Basic Channel Information Page



Channel Identifier:	55
Status:	Inactive
Call ID:	0
Endpoint ID:	Not Available
Call Duration [sec]:	0
Call Type:	Voice
Call Destination:	10.13.4.12
Coder:	Transparent

To view additional channel information, click the required tab (**SIP**, **RTP/RTCP**, and **Voice Settings**).

4. To view the settings of a specific trunk, click the required trunk icon, and then from the shortcut menu, choose **Port Settings**; the Trunk Settings page opens, displaying the trunk's settings. If needed, you can modify the settings (see "Configuring Trunk Settings" on page 347).

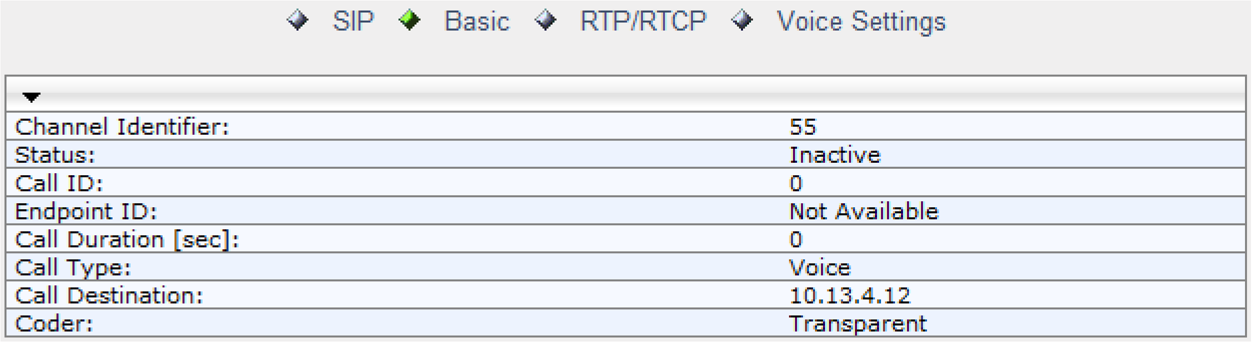
54.2 Viewing Analog Port Information

The Home page allows you to view detailed information on selected FXS and FXO analog ports such as RTP/RTCP and voice settings.

➢ **To view information on an analog port:**

1. Open the Home page.
2. On the graphical display of the device, click the required analog port; a shortcut menu appears.
3. From the shortcut menu, choose **Port Settings**; the Basic Channel Information page appears with the **Basic** tab selected (displayed in green):

Figure 54-3: Basic Channel Information Page



Channel Identifier:	55
Status:	Inactive
Call ID:	0
Endpoint ID:	Not Available
Call Duration [sec]:	0
Call Type:	Voice
Call Destination:	10.13.4.12
Coder:	Transparent

4. To view additional channel information, click the required tab - **SIP**, **RTP/RTCP**, and **Voice Settings**.

54.3 Viewing NFAS Groups and D-Channel Status

The NFAS Group & D-Channel Status page displays the status of the device's D-channels and NFAS groups. The status of a D-channel and NFAS group can be "In Service" or "Out of Service". This page also indicates whether the D-channel is a primary or backup D-channel.

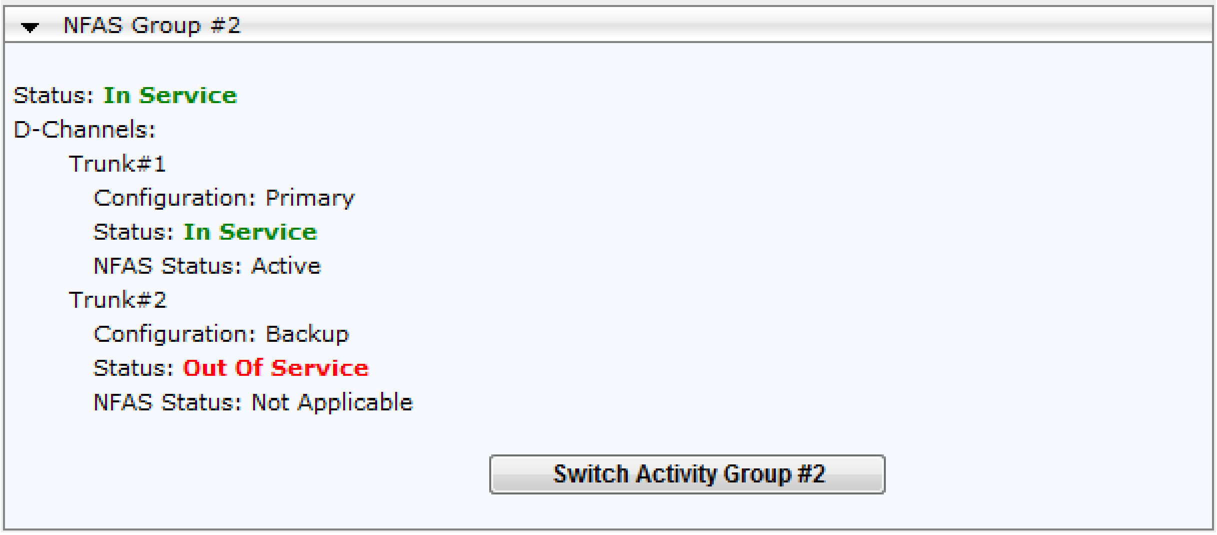
This page also enables you to manually switchover between active and standby D-channels belonging to the same NFAS group. This is done using the **Switch Activity** button. For more information, see "Performing Manual D-Channel Switchover in NFAS Group" on page 362.

Note: This page is applicable only to T1 ISDN protocols supporting NFAS, and only if the NFAS group is configured with two D-channels.

➢ **To view the status of the D-channels and NFAS groups:**

- Open the NFAS Group & D-Channel Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **NFAS Group & D-Channel Status**).

Figure 54-4: NFAS Group & D-Channel Status Page



▼ NFAS Group #2

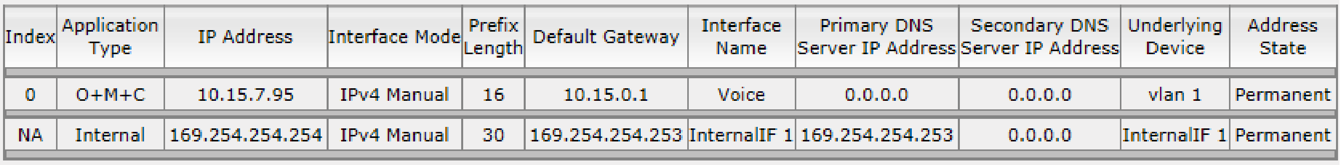
Status: In Service
D-Channels:
Trunk#1
Configuration: Primary
Status: In Service
NFAS Status: Active
Trunk#2
Configuration: Backup
Status: Out Of Service
NFAS Status: Not Applicable

54.4 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 130).

➢ **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	Address State
0	O+M+C	10.15.7.95	IPv4 Manual	16	10.15.0.1	Voice	0.0.0.0	0.0.0.0	vlan 1	Permanent
NA	Internal	169.254.254.254	IPv4 Manual	30	169.254.254.253	InternalIF 1	169.254.254.253	0.0.0.0	InternalIF 1	Permanent

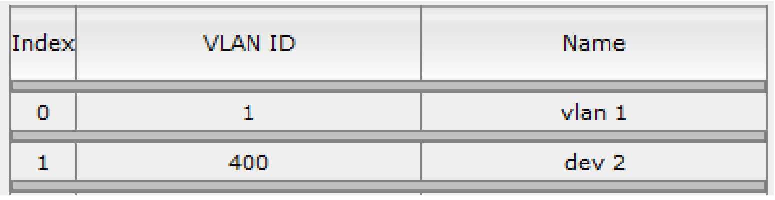
54.5 Viewing Ethernet Device Status

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 129.

➢ **To view the configured and applied Ethernet Devices:**

- Open the Ethernet Device Status page (**Status & Diagnostics** tab > **VoIP Status** menu >**Ethernet Device Status Table**).

Figure 54-5: Ethernet Device Status Page



Index	VLAN ID	Name
0	1	vlan 1
1	400	dev 2

54.6 Viewing Static Routes Status

The IP Routing Status Table page displays the status of the static routes. These are routes configured in the Static Route table (see "Configuring Static IP Routing" on page 139) and routes through the Default Gateway.

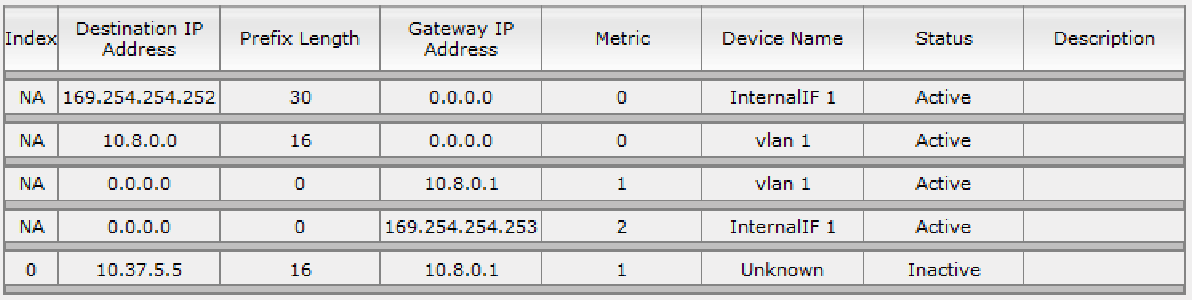
The status of the static routes can be one of the following:

- "Active": Static route is used by the device.
- "Inactive": Static route is not used. When the destination IP address is not on the same segment with the next hop, or the interface does not exist, the route state changes to "Inactive".

➢ To view the status of static IP routing:

- Open the IP Routing Status Table page (**Status & Diagnostics** tab > **VoIP Status** menu >**Static Route Status**).

Figure 54-6: 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	

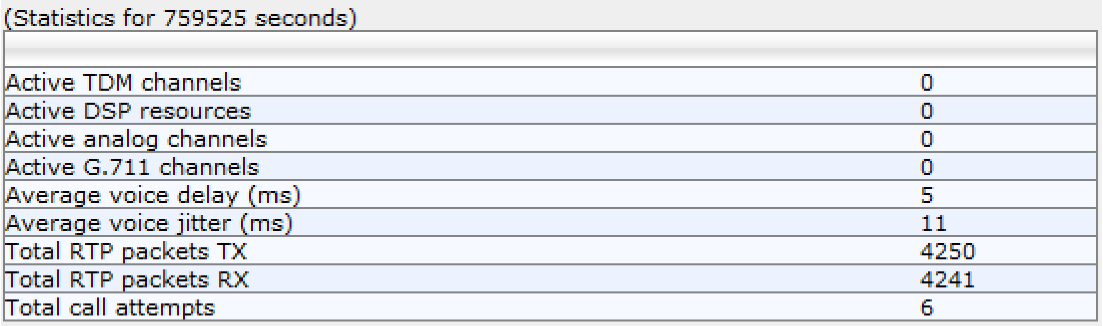
54.7 Viewing Performance Statistics

The Basic Statistics page provides read-only, device performance statistics. This page is refreshed every 60 seconds. The duration that the currently displayed statistics has been collected is displayed above the statistics table.

➢ To view performance statistics:

- Open the Basic Statistics page (**Status & Diagnostics** tab > **VoIP Status** menu > **Performance Statistics**).

Figure 54-7: Basic Statistics Page



(Statistics for 759525 seconds)

Active TDM channels	0
Active DSP resources	0
Active analog channels	0
Active G.711 channels	0
Average voice delay (ms)	5
Average voice jitter (ms)	11
Total RTP packets TX	4250
Total RTP packets RX	4241
Total call attempts	6

The duration that the displayed statistics were collected is displayed in seconds above the table. To reset the performance statistics to zero, click the **Reset Statistics** button.

54.8 Viewing Call Counters

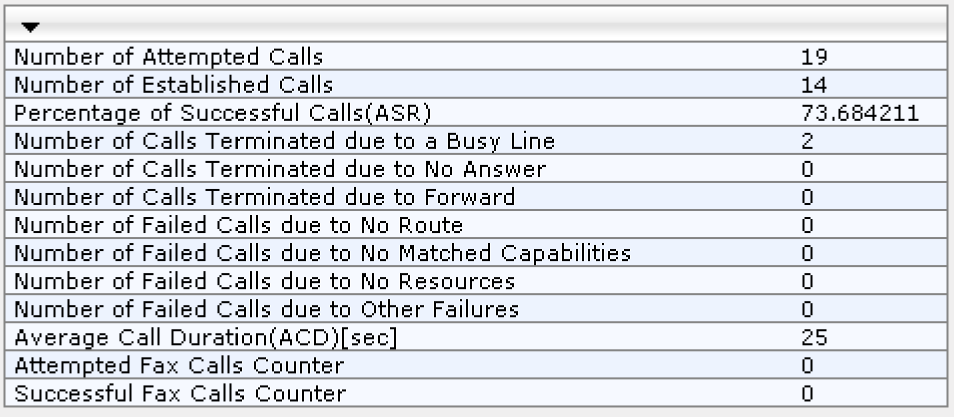
The IP to Tel Calls Count page and Tel to IP Calls Count page provide you with statistical information on incoming (IP-to-Tel) and outgoing (Tel-to-IP) calls. The statistical information is updated according to the release reason that is received after a call is terminated (during the same time as the end-of-call Call Detail Record or CDR message is sent). The release reason can be viewed in the 'Termination Reason' field in the CDR message.

You can reset the statistical data displayed on the page (i.e., refresh the display), by clicking the **Reset Counters** button located below the table.

➢ **To view IP-to-Tel and Tel-to-IP call counters:**

- Open the Call Counters page that you want to view (**Status & Diagnostics** tab > **VoIP Status** menu > **IP to Tel Calls Count** or **Tel to IP Calls Count**); the figure below shows the IP to Tel Calls Count page.

Figure 54-8: Calls Count Page



Number of Attempted Calls	19
Number of Established Calls	14
Percentage of Successful Calls(ASR)	73.684211
Number of Calls Terminated due to a Busy Line	2
Number of Calls Terminated due to No Answer	0
Number of Calls Terminated due to Forward	0
Number of Failed Calls due to No Route	0
Number of Failed Calls due to No Matched Capabilities	0
Number of Failed Calls due to No Resources	0
Number of Failed Calls due to Other Failures	0
Average Call Duration(ACD)[sec]	25
Attempted Fax Calls Counter	0
Successful Fax Calls Counter	0

The fields in this page are described in the following table:

Table 54-3: Call Counters Description

Counter	Description
Number of Attempted Calls	Indicates the number of attempted calls. It is composed of established and failed calls. The number of established calls is represented by the 'Number of Established Calls' counter. The number of failed calls is represented by the failed-call counters. Only one of the established / failed call counters is incremented every time.
Number of Established Calls	Indicates the number of established calls. It is incremented as a result of one of the following release reasons if the duration of the call is greater than zero: ▪ GWAPP_REASON_NOT_RELEVANT (0) ▪ GWAPP_NORMAL_CALL_CLEAR (16) ▪ GWAPP_NORMAL_UNSPECIFIED (31) And the internal reasons: ▪ RELEASE_BECAUSE_UNKNOWN_REASON ▪ RELEASE_BECAUSE_REMOTE_CANCEL_CALL ▪ RELEASE_BECAUSE_MANUAL_DISC ▪ RELEASE_BECAUSE_SILENCE_DISC ▪ RELEASE_BECAUSE_DISCONNECT_CODE **Note:** When the duration of the call is zero, the release reason GWAPP_NORMAL_CALL_CLEAR increments the 'Number of Failed Calls due to No Answer' counter. The rest of the release reasons

Counter	Description
	increment the 'Number of Failed Calls due to Other Failures' counter.
Percentage of Successful Calls (ASR)	The percentage of established calls from attempted calls.
Number of Calls Terminated due to a Busy Line	Indicates the number of calls that failed as a result of a busy line. It is incremented as a result of the following release reason: GWAPP_USER_BUSY (17)
Number of Calls Terminated due to No Answer	Indicates the number of calls that weren't answered. It's incremented as a result of one of the following release reasons: ▪ GWAPP_NO_USER_RESPONDING (18) ▪ GWAPP_NO_ANSWER_FROM_USER_ALERTED (19) ▪ GWAPP_NORMAL_CALL_CLEAR (16) (when the call duration is zero)
Number of Calls Terminated due to Forward	Indicates the number of calls that were terminated due to a call forward. The counter is incremented as a result of the following release reason: RELEASE_BECAUSE_FORWARD
Number of Failed Calls due to No Route	Indicates the number of calls whose destinations weren't found. It is incremented as a result of one of the following release reasons: ▪ GWAPP_UNASSIGNED_NUMBER (1) ▪ GWAPP_NO_ROUTE_TO_DESTINATION (3)
Number of Failed Calls due to No Matched Capabilities	Indicates the number of calls that failed due to mismatched device capabilities. It is incremented as a result of an internal identification of capability mismatch. This mismatch is reflected to CDR via the value of the parameter DefaultReleaseReason (default is GWAPP_NO_ROUTE_TO_DESTINATION (3)) or by the GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED (79) reason.
Number of Failed Calls due to No Resources	Indicates the number of calls that failed due to unavailable resources or a device lock. The counter is incremented as a result of one of the following release reasons: ▪ GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED ▪ RELEASE_BECAUSE_GW_LOCKED
Number of Failed Calls due to Other Failures	This counter is incremented as a result of calls that failed due to reasons not covered by the other counters.
Average Call Duration (ACD) [sec]	The average call duration (ACD) in seconds of established calls. The ACD value is refreshed every 15 minutes and therefore, this value reflects the average duration of all established calls made within a 15 minute period.
Attempted Fax Calls Counter	Indicates the number of attempted fax calls.
Successful Fax Calls Counter	Indicates the number of successful fax calls.

54.9 Viewing Registered Users

You can view SAS and 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:

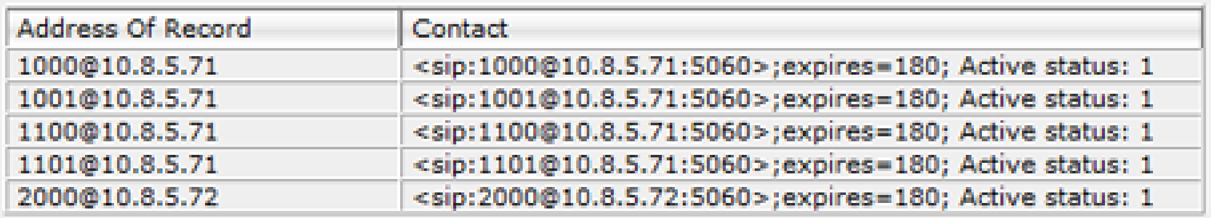
- "Active status:1" indicates that the contact has been successfully registered and thus, calls can be routed to it.
- "Active status:0" indicates that the device has recently received a REGISTER request from the contact, but the contact has yet to be registered. The device removes the contact from the 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 SAS/SBC users in the Users Registration database:**

- Web: SAS/SBC Registered Users page (**Status & Diagnostics** tab > **VoIP Status** menu > **SAS/SBC Registered Users**).

Figure 54-9: 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>
    ```

  - SAS users:

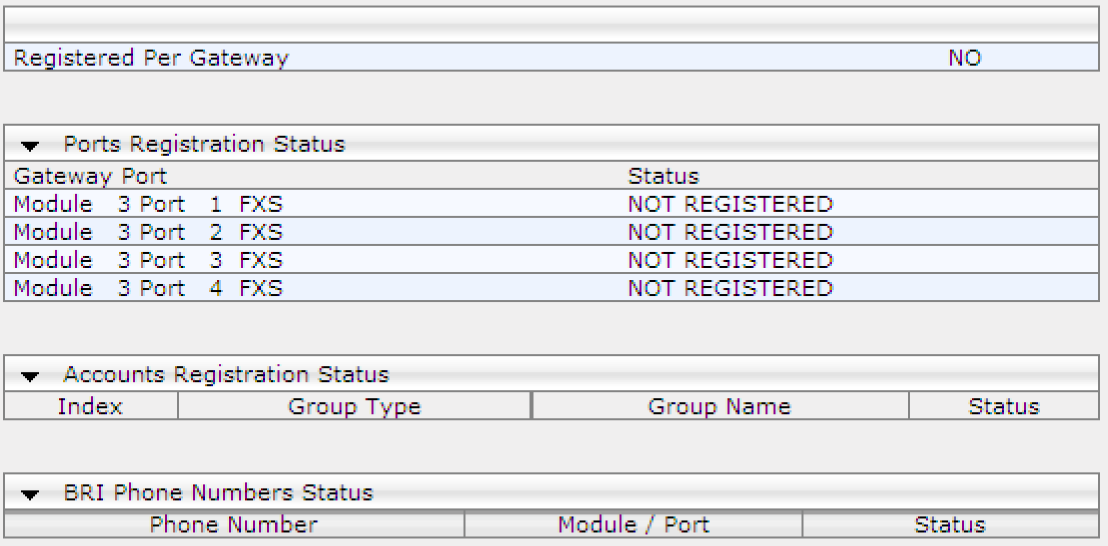
    ```
    # show voip register db sas list
    ```

54.10 Viewing Registration Status

The Registration Status page displays the registration status of the device's endpoints (such as FXS, FXO and BRI) and SIP Accounts, which are configured in the Accounts table (see "Configuring Registration Accounts" on page 291).

➢ **To view registration status:**

- Open the Registration Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Registration Status**).



Registered Per Gateway	NO

▼ Ports Registration Status

Gateway Port	Status
Module 3 Port 1 FXS	NOT REGISTERED
Module 3 Port 2 FXS	NOT REGISTERED
Module 3 Port 3 FXS	NOT REGISTERED
Module 3 Port 4 FXS	NOT REGISTERED

▼ Accounts Registration Status

Index	Group Type	Group Name	Status

▼ BRI Phone Numbers Status

Phone Number	Module / Port	Status

- Registered Per Gateway (applicable only to the Gateway application): Registration of device as one entity - "YES" or "NO"
- Ports Registration Status: "REGISTERED" or "NOT REGISTERED"
- **Accounts Registration Status:**
  - **Group Type:** served Trunk Group or IP Group
  - **Group Name:** name of served Trunk Group or IP Group, if applicable
  - **Status:** "Registered" or "Unregistered"
- BRI Phone Number Status:
  - Phone Number: phone number of BRI endpoint
  - Module/Port: module/port number of BRI endpoint
  - Status: "Registered" or "Unregistered"

Note: The registration mode (i.e., per device, endpoint, account. or no registration) is configured in the Hunt Group Settings table (see Configuring Hunt Group Settings on page 367) or using the TrunkGroupSettings ini file parameter.

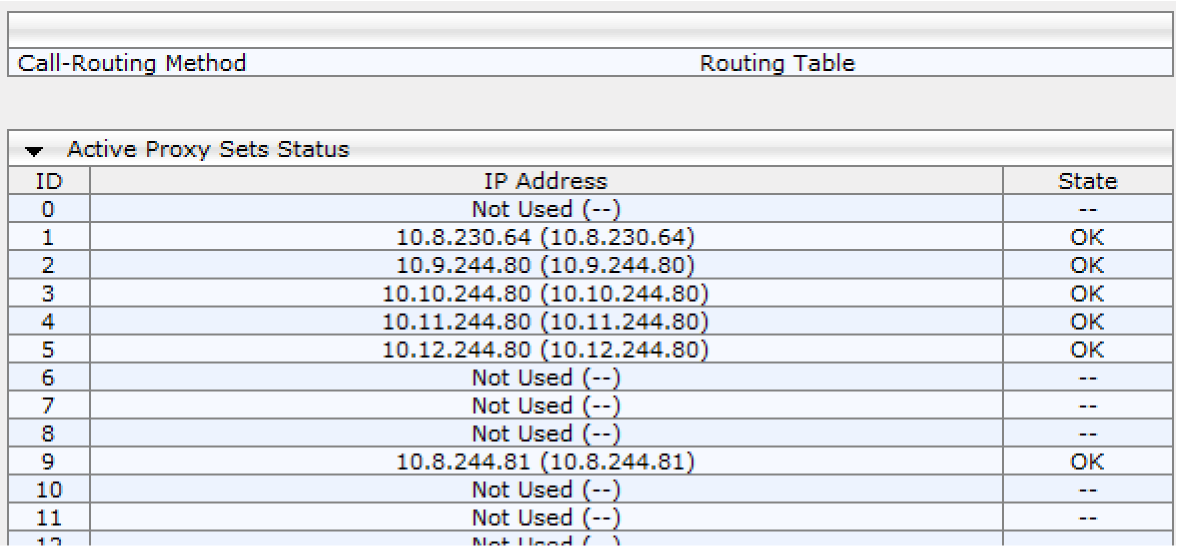
54.11 Viewing Call Routing Status

The Call Routing Status page provides you with information on the current routing method used by the device. This information includes the IP address and FQDN (if used) of the Proxy server with which the device currently operates.

➢ **To view call routing status:**

- Open the Call Routing Status page (**Status & Diagnostics** tab > **VoIP Status** menu > **Call Routing Status**).

Figure 54-10: Call Routing Status Page



Call-Routing Method	Routing Table

▼ Active Proxy Sets Status

ID	IP Address	State
0	Not Used (--)	--
1	10.8.230.64 (10.8.230.64)	OK
2	10.9.244.80 (10.9.244.80)	OK
3	10.10.244.80 (10.10.244.80)	OK
4	10.11.244.80 (10.11.244.80)	OK
5	10.12.244.80 (10.12.244.80)	OK
6	Not Used (--)	--
7	Not Used (--)	--
8	Not Used (--)	--
9	10.8.244.81 (10.8.244.81)	OK
10	Not Used (--)	--
11	Not Used (--)	--
12	Not Used (--)	

Table 54-4: Call Routing Status Parameters

Parameter	Description
Call-Routing Method	▪ Proxy/GK = Proxy server (Proxy Set) is used to route calls. For configuring Proxy Sets, see "Configuring Proxy Sets" on page 282. ▪ Routing Table = Calls are routed using the routing table: ✓ Gateway calls: Outbound IP Routing table (Configuring Outbound IP Routing Table on page 397) ✓ SBC calls: SBC IP-to-IP Routing table (Configuring SBC IP-to-IP Routing Rules on page 552)
IP Address	▪ Not Used = Proxy server isn't defined. ▪ IP address and FQDN (if exists) of the Proxy server with which the device currently operates.
State	▪ N/A = Proxy server isn't defined. ▪ OK = Communication with the Proxy server is in order. ▪ Fail = No response from any of the defined Proxies.

54.12 Viewing IP Connectivity

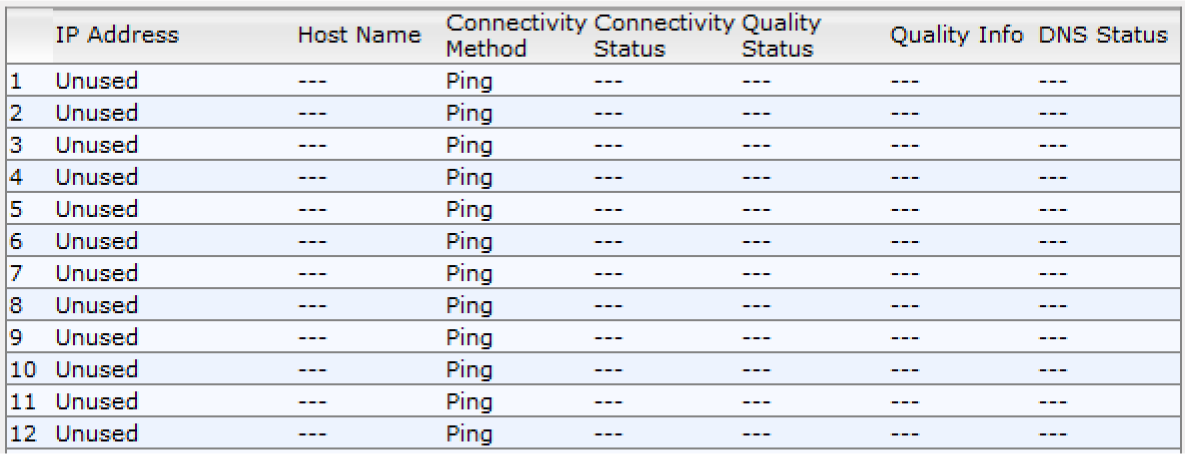
The IP Connectivity page displays on-line, read-only network diagnostic connectivity information on all destination IP addresses configured in the Outbound IP Routing table (see "Configuring Outbound IP Routing Table" on page 397).

Note: he information in columns 'Quality Status' and 'Quality Info' (per IP address) is reset if two minutes elapse without a call to that destination.

➢ **To view IP connectivity information:**

1. In the Routing General Parameters page, set the 'Enable Alt Routing Tel to IP' parameter (AltRoutingTel2IPMode) to **Enable** or **Status Only** (see "Configuring General Routing Parameters" on page 397).
2. Open the IP Connectivity page (**Status & Diagnostics** tab > **VoIP Status** menu > **IP Connectivity**).

Figure 54-11: IP Connectivity Page



	IP Address	Host Name	Connectivity Method	Connectivity Status	Quality Status	Quality Info	DNS Status
1	Unused	---	Ping	---	---	---	---
2	Unused	---	Ping	---	---	---	---
3	Unused	---	Ping	---	---	---	---
4	Unused	---	Ping	---	---	---	---
5	Unused	---	Ping	---	---	---	---
6	Unused	---	Ping	---	---	---	---
7	Unused	---	Ping	---	---	---	---
8	Unused	---	Ping	---	---	---	---
9	Unused	---	Ping	---	---	---	---
10	Unused	---	Ping	---	---	---	---
11	Unused	---	Ping	---	---	---	---
12	Unused	---	Ping	---	---	---	---

Table 54-5: IP Connectivity Parameters

Column Name	Description
IP Address	The IP address can be one of the following: ▪ IP address defined as the destination IP address in the Outbound IP Routing table. ▪ IP address resolved from the host name defined as the destination IP address in the Outbound IP Routing table.
Host Name	Host name (or IP address) as defined in the Outbound IP Routing table.
Connectivity Method	The method according to which the destination IP address is queried periodically (SIP OPTIONS request).
Connectivity Status	The status of the IP address' connectivity according to the method in the 'Connectivity Method' field. ▪ OK = Remote side responds to periodic connectivity queries. ▪ Lost = Remote side didn't respond for a short period. ▪ Fail = Remote side doesn't respond. ▪ Init = Connectivity queries not started (e.g., IP address not resolved).

Column Name	Description
	▪ Disable = The connectivity option is disabled, i.e., parameter 'Alt Routing Tel to IP Mode' (AltRoutingTel2IPMode *ini*) is set to 'None' or 'QoS'.
Quality Status	Determines the QoS (according to packet loss and delay) of the IP address. ▪ Unknown = Recent quality information isn't available. ▪ OK ▪ Poor **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
Quality Info.	Displays QoS information: delay and packet loss, calculated according to previous calls. **Notes:** ▪ This parameter is applicable only if the parameter 'Alt Routing Tel to IP Mode' is set to 'QoS' or 'Both' (AltRoutingTel2IPMode = 2 or 3). ▪ This parameter is reset if no QoS information is received for 2 minutes.
DNS Status	DNS status can be one of the following: ▪ DNS Disable ▪ DNS Resolved ▪ DNS Unresolved

55 Reporting Information to External Party

This section describes features for reporting various information to an external party.

55.1 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.

Note: The RTCP XR feature is available only if the device is installed with a Software License Key that includes this feature. For installing a Software License Key, see "Software License Key" on page 621.

RTCP XR messages containing key call-quality-related metrics are exchanged periodically (user-defined) between the device and the SIP UA. This allows an analyzer to monitor these metrics midstream, or a device to retrieve them using SNMP. The device sends RTCP XR reports to an Event State Compositor (ESC) server, using SIP PUBLISH messages. These reports can be sent at the end of each call and according to a user-defined interval between consecutive reports.

Table 55-1: 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
	Silence Suppression State
Jitter Buffer	Jitter Buffer Adaptive
	Jitter Buffer Rate
	Jitter Buffer Nominal

Group	Metric Name
	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

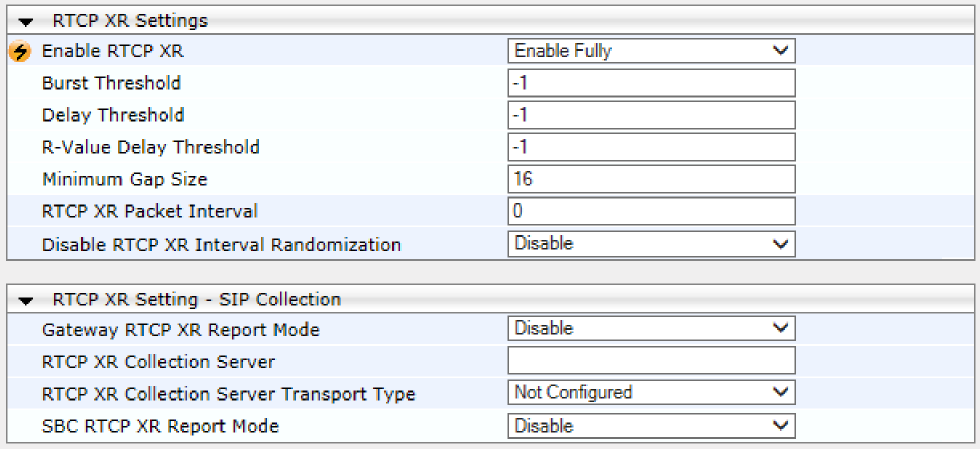
Below shows an example of a SIP PUBLISH message sent with RTCP XR and QoE information:

```
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=137461752128420121558350@10.8.61.16
    LocalID: <sip:12345@10.8.61.16>
    RemoteID: <sip:54321@10.8.61.18>
    OrigID: <sip:12345@10.8.61.16>
    LocalAddr: IP=10.8.61.16 Port=6110 SSRC=0xce110633
    RemoteAddr: IP=10.8.61.18 Port=6050 SSRC=0xffffffff
    LocalGroup:
    RemoteGroup:
    LocalMAC: 00:90:8f:2e:3c:67
    LocalMetrics:
    Timestamps: START=2012-04-28T15:58:36Z STOP=2012-04-
28T15:58:36Z
    SessionDesc: PT=8 PD=PCMA SR=8000 FD=20 PLC=3 SSUP=Off
    JitterBuffer: JBA=3 JBR=0 JBN=0 JBM=0 JBX=300
    PacketLoss: NLR=0.00 JDR=0.00
    BurstGapLoss: BLD=0.00 BD=0 GLD=0.00 GD=0 GMIN=16
    Delay: RTD=0 ESD=0
    QualityEst:
    DialogID:137461752128420121558350@10.8.61.16;to-
tag=1c252030485; from-tag=1c1374725246
```

➢ To configure RTCP XR:

1. Open the RTP/RTCP Settings page (**Configuration** tab > **VoIP** menu > **Media** > **RTP/RTCP Settings**). The RTCP XR parameters are listed under the RTCP XR Settings group:

Figure 55-1: RTCP XR Parameters in RTP/RTCP Settings Page



RTCP XR Settings

Enable RTCP XR	Enable Fully
Burst Threshold	-1
Delay Threshold	-1
R-Value Delay Threshold	-1
Minimum Gap Size	16
RTCP XR Packet Interval	0
Disable RTCP XR Interval Randomization	Disable

RTCP XR Setting - SIP Collection

Gateway RTCP XR Report Mode	Disable
RTCP XR Collection Server	
RTCP XR Collection Server Transport Type	Not Configured
SBC RTCP XR Report Mode	Disable

2. Under the RTCP XR Settings group, configure the following:
   - 'Enable RTCP XR' (*VQMonEnable*) - enables voice quality monitoring and RTCP XR.
   - 'Burst Threshold' (*VQMonBurstHR*) - defines the voice quality monitoring excessive burst alert threshold.
   - 'Delay Threshold' (*VQMonDelayTHR*) - defines the voice quality monitoring excessive delay alert threshold.
   - 'R-Value Delay Threshold' (*VQMonEOCRValTHR*) - defines the voice quality monitoring end of call low quality alert threshold.
   - 'Minimum Gap Size' (*VQMonGMin*) - defines the voice quality monitoring minimum gap size (number of frames).
   - 'RTCP XR Packet Interval' (*RTCPInterval*) - defines the time interval between adjacent RTCP reports.
   - 'Disable RTCP XR Interval Randomization' (*DisableRTCPRandomize*) - determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval.
3. Under the RTCP XR Setting - SIP Collection group, configure the following:
   - **(Gateway Application Only)** 'Gateway RTCP XR Report Mode' (*RTCPXRReportMode*) - determines whether RTCP XR reports are sent to the ESC server and defines the interval at which they are sent.
   - **(Gateway Application Only)** 'RTCP XR Collection Server' (*RTCPXREscIP*) - defines the IP address of the ESC server.
   - **(Gateway Application Only)** 'RTCP XR Collection Server Transport Type' (*RTCPXRESCTransportType*) - determines the transport layer for outgoing SIP dialogs initiated by the device to the ESC server.
   - **(SBC Application Only)** 'SBC RTCP XR Report Mode' (SBCRtcpXrReportMode) - enables the sending of RTCP XR reports of QoE metrics at the end of each call session (i.e., after a SIP BYE).
4. Click **Submit**, and then reset the device with a save ("burn") for your settings to take effect.

55.2 Generating Call Detail Records

The Call Detail Record (CDR) contains vital statistic information on calls made from the device. The device can be configured to generate and report CDRs for various stages of the call, including SIP messages and/or media. You can configure when CDRs for a call are generated, for example, only at the end of the call or only at the start and end of the call. Once generated, the device sends the CDRs to a user-defined Syslog server.

The CDR Syslog message complies with RFC 3161 and is identified by Facility 17 (local1) and Severity 6 (Informational).

For CDR in RADIUS format, see "Configuring RADIUS Accounting" on page 704.

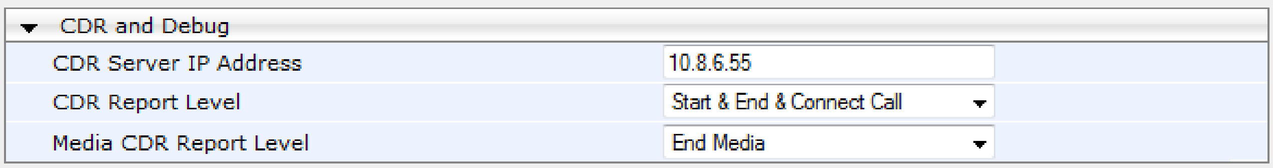
55.2.1 Configuring CDR Reporting

The following procedure describes how to configure CDR reporting.

➢ **To configure CDR reporting:**

1. Enable the Syslog feature for sending log messages generated by the device to a collecting log message server. For more information, see "Enabling Syslog" on page 719.
2. Open the Advanced Parameters page (**Configuration** tab > **VoIP** menu > **SIP Definitions** > **Advanced Parameters**). The CDR parameters appear under the 'CDR and Debug' group, as shown below:

Figure 55-2: CDR Parameters in Advanced Parameters Page



▾ CDR and Debug

CDR Server IP Address	10.8.6.55
CDR Report Level	Start & End & Connect Call
Media CDR Report Level	End Media

3. Configure the parameters as required. For a description of the parameters, see "Syslog, CDR and Debug Parameters" on page 788.
4. Click **Submit**.

Note: If the CDR server IP address is not configured, the CDRs are sent to the Syslog server configured in "Enabling Syslog" on page 719.

55.2.2 CDR Field Description

This section describes the CDR fields that are generated by the device.

55.2.2.1 CDR Fields for SBC Signaling

The CDR fields for SBC signaling are listed in the table below.

A typical SBC session consists of two SBC legs. Each leg generates its own signaling CDRs. Each leg generates three CDR types: at call start (SBCReportType=CALL_START), connect time (SBCReportType=CALL_CONNECT) and when the call ends (SBCReportType=CALL_END). CDRs belonging to the same SBC session (both legs) have the same Session ID (SessionId CDR field). CDRs belonging to the same SBC leg have the same SIP Call ID (SIPCallId CDR field).

For billing applications, the CDR that is sent when the call ends (CALL_END) is usually sufficient. Billing may be based on the following:

- Call ID (SIPCallId CDR field)
- Source URI (SrcURI CDR field)
- Destination URI (DstURI CDR field)
- Call originator (Orig CDR filed) - 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 55-2: CDR Fields for SBC Signaling

CDR Field Name	Description	Format
SBCReportType	Report Type: ▪ "CALL_START" ▪ "CALL_CONNECT" ▪ "CALL_END" ▪ "DIALOG_START" ▪ "DIALOG_END"	String
EPTyp	Endpoint type: ▪ "SBC"	String
SIPMethod	SIP message type	String of up to 10 characters
SIPCallId	Unique ID of call	String of up to 50 characters
SessionId	Unique Session ID	String of up to 10 characters
Orig	Call originator: ▪ "LCL" - local ▪ "RMT" **-** remote	String
SourceIp	Source IP address	String of up to 20 characters
SourcePort	Source UDP port	String of up to 10 characters
DestIp	Destination IP address	String of up to 20 characters
DestPort	Destination UDP port	String of up to 10 characters
TransportType	Transport type: ▪ "UDP" ▪ "TCP" ▪ "TLS"	String
SrcURI	Source URI	String of up to 41 characters
SrcURIBeforeMap	Source URI before manipulation	String of up to 41 characters
DstURI	Destination URI	String of up to 41 characters
DstURIBeforeMap	Destination URI before manipulation	String of up to 41 characters
Durat	Call duration (in seconds)	String of up to 5 characters
TrmSd	Termination side: ▪ "LCL" – local ▪ "RMT" **-** remote	String
TrmReason	Termination reason	String of up to 40 characters
TrmReasonCategory	Termination reason category: **Calls with duration 0 (i.e., not connected):** ▪ NO_ANSWER: ✓ "GWAPP_NORMAL_CALL_CLEAR" ✓ "GWAPP_NO_USER_RESPONDING" ✓ "GWAPP_NO_ANSWER_FROM_USER_ALERTED" ▪ BUSY:	String of up to 17 characters

CDR Field Name	Description	Format
	✓ " GWAPP_USER_BUSY" ▪ NO_RESOURCES: ✓ "GWAPP_RESOUUCE_UNAVAILABLE_UNSPECIFIED" ✓ "RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT" ✓ "RESOURCE_BECAUSE_NO_TRANSCODING_RESOURCES_LEFT" ✓ "RELEASE_BECAUSE_GW_LOCKED" ▪ NO_MATCH: ✓ "RELEASE_BECAUSE_UNMATCHED_CAPABILITIES" ▪ FORWARDED: ✓ "RELEASE_BECAUSE_FORWARD" ▪ GENERAL_FAILED: Any other reason **Calls with duration:** ▪ NORMAL_CALL_CLEAR: ✓ "GWAPP_NORMAL_CALL_CLEAR" ▪ ABNORMALLY_TERMINATED: Anything else **N/A:** Reasons not belonging to above categories	
SetupTime	Call setup time	String of up to 35 characters
ConnectTime	Call connect time	String of up to 35 characters
ReleaseTime	Call release time	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

Below shows an example of an SBC signaling CDR sent at the end of the call (call was not answered):

```
[S=1] |SBCReportType |EPTyp| SIPCallId| SessionId |Orig |SourceIp
|SourcePort |DestIp |DestPort |TransportType |SrcURI
|SrcURIBeforeMap |DstURI |DstURIBeforeMap |Durat |TrmSd |TrmReason
|TrmReasonCategory |SetupTime |ConnectTime |ReleaseTime
|RedirectReason |RedirectURINum |RedirectURINumBeforeMap
|TxSigIPDiffServ |IPGroup (description) |SrdId (name)
|SIPInterfaceId |ProxySetId |IpProfileId (name) |MediaRealmId
(name) |DirectMedia |SIPTrmReason

[S=3] |CALL_END |SBC |170369730753201211288@10.132.10.245 |0 |RMT
|10.132.10.245 |5060 |10.132.10.250 |5070 |UDP |103@audiocodes.com
|103@audiocodes.com |101@10.132.10.250 |101@10.132.10.250 |0 |RMT
|GWAPP_NORMAL_CALL_CLEAR |NO_ANSWER |06:13:54.950 UTC Thu Mar 02
2012 | |06:14:01.175 UTC Thu Mar 02 2012 |-1 | | |40 |2 () |0
(5070SRD) |2 |3 |0 () |0 (lanmedia) |no |CANCEL
```

55.2.2.2 CDR Fields for SBC Media

The CDR fields for SBC media are listed in the table below. The media CDRs are published for each active media stream, thereby allowing multiple media CDRs, where each media CDR has a unique call ID corresponding to the signaling CDR.

Table 55-3: CDR Fields for SBC Media

CDR Field Name	Description
MediaReportType	Report type (media start, update, or end)
SIPCallId	Unique call ID
Cid	Channel CID
MediaType	Media type (audio, video, or text)
Coder	Coder name
PacketInterval	Coder packet interval

CDR Field Name	Description
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
RemoteRFactor	Remote conversation quality
LocalMosCQ	Local MOS for conversation
RemoteMosCQ	Remote MOS for conversation
TxRTPIPDiffServ	Media IP DiffServ
LatchedRtpIp	Remote IP address of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal.
LatchedRtpPort	Remote RTP port of the incoming RTP stream that the device "latched" on to as a result of the RTP latching mechanism for NAT traversal.
LatchedT38Ip	Latching of a new T.38 stream - new IP address
LatchedT38Port	Latching of a new T.38 stream - new port

55.2.2.3 CDR Fields for Gateway/IP-to-IP Application

The CDR fields for the Gateway / IP-to-IP application are listed in the table below.

Table 55-4: CDR Fields for Gateway/IP-to-IP Application

Field Name	Description
GWReportType	Report type: ▪ CALL_START ▪ CALL_CONNECT ▪ CALL_END
Cid	Port number
SessionId	SIP session identifier

Field Name	Description
Trunk	Physical trunk number **Note:** This field is applicable only to the Gateway application.
BChan	Selected B-channel **Note:** This field is applicable only to the Gateway application.
ConId	SIP conference ID **Note:** This field is applicable only to the Gateway application.
TG	Trunk Group ID **Note:** This field is applicable only to the Gateway application.
EPTyp	Endpoint type: ▪ FXO ▪ FXS ▪ EANDM ▪ ISDN ▪ CAS ▪ DAA ▪ IPMEDIA ▪ NETANN ▪ STREAMING ▪ TRANSPARENT ▪ MSCML ▪ VXML ▪ IP2IP
Orig	Call originator: ▪ LCL (Tel side) ▪ RMT (IP side)
SourceIp	Source IP address
DestIp	Destination IP address
TON	Source phone number type **Note:** This field is applicable only to the Gateway application.
NPI	Source phone number plan **Note:** This field is applicable only to the Gateway application.
SrcPhoneNum	Source phone number
SrcNumBeforeMap	Source number before manipulation
TON	Destination phone number type **Note:** This field is applicable only to the Gateway application.
NPI	Destination phone number plan **Note:** This field is applicable only to the Gateway application.
DstPhoneNum	Destination phone number
DstNumBeforeMap	Destination number before manipulation
Durat	Call duration
Coder	Selected coder

Field Name	Description
Intrv	Packet interval
RtpIp	RTP IP address
Port	Remote RTP port
TrmSd	Initiator of call release (IP, Tel, or Unknown)
TrmReason	SIP call termination reason (see "Release Reasons in CDR" on page 701)
Fax	Fax transaction during call
InPackets	Number of incoming packets
OutPackets	Number of outgoing packets
PackLoss	Local packet loss
RemotePackLoss	Number of outgoing lost packets
SIPCallId	Unique SIP call ID
SetupTime	Call setup time
ConnectTime	Call connect time
ReleaseTime	Call release time
RTPdelay	RTP delay
RTPjitter	RTP jitter
RTPssrc	Local RTP SSRC
RemoteRTPssrc	Remote RTP SSRC
RedirectReason	Redirect reason
TON	Redirection phone number type **Note:** This field is applicable only to the Gateway application.
NPI	Redirection phone number plan **Note:** This field is applicable only to the Gateway application.
RedirectPhonNum	Redirection phone number
MeteringPulses	Number of generated metering pulses **Note:** This field is applicable only to the Gateway application.
SrcHost	Source host name
SrcHostBeforeMap	Source host name before manipulation
DstHost	Destination host name
DstHostBeforeMap	Destination host name before manipulation
IPG	IP Group description
LocalRtpIp	Remote RTP IP address
LocalRtpPort	Local RTP port

Field Name	Description
Amount	0-999999 Data is stored per call and sent in the syslog as follows: ▪ currency-type: amount multiplier for currency charge (euro or usd) ▪ recorded-units: for unit charge (1-999999)
Mult	0,001-1000 (in steps of 10) (See explanation above.)
TrmReasonCategory	Termination reason category: ▪ Calls with duration 0 (i.e., not connected): ✓ **NO_ANSWER -** GWAPP_NORMAL_CALL_CLEAR, GWAPP_NO_USER_RESPONDING, GWAPP_NO_ANSWER_FROM_USER_ALERTED ✓ **BUSY -** GWAPP_USER_BUSY ✓ **NO_RESOURCES -** GWAPP_RESOUUCE_UNAVAILABLE_UNSPECIFIED, RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT, RESOURCE_BECAUSE_NO_TRANSCODING_RESOURCES_LEFT, RELEASE_BECAUSE_GW_LOCKED ✓ **NO_MATCH -** RELEASE_BECAUSE_UNMATCHED_CAPABILITIES ✓ **FORWARDED -** RELEASE_BECAUSE_FORWARD ✓ **GENERAL_FAILED -** any other reason ▪ Calls with duration: ✓ **NORMAL_CALL_CLEAR -** GWAPP_NORMAL_CALL_CLEAR ✓ **ABNORMALLY_TERMINATED -** Anything else ▪ **N/A -** Reasons not belonging to above categories
RedirectNumBeforeMap	Redirect number before manipulation
SrdId	SRD ID name
SIPInterfaceId	SIP interface ID
ProxySetId	Proxy Set ID
IpProfileId	IP Profile ID name
MediaRealmId	Media Realm name
SigTransportType	SIP signaling transport type (UDP, TCP, or TLS)
TxRTPIPDiffServ	Media IP DiffServ
TxSigIPDiffServ	Signaling IP DiffServ
LocalRFactor	Local R-factor
RemoteRFactor	Remote R-factor
LocalMosCQ	Local MOS for conversation quality
RemoteMosCQ	Remote MOS for conversation quality
SigSourcePort	SIP source port
SigDestPort	SIP destination port
MediaType	Media type - audio, video, or text
SIPTrmReason	SIP call termination reason (BYE, CANCEL, or SIP error codes, e.g., 404)

Field Name	Description
SipTermDesc	Description of SIP termination reason: ▪ SIP Reason header, if exists, for example: SIP ;cause=200 ;text="Call completed elsewhere". ▪ If no SIP Reason header exists, the description is taken from the reason text, if exists, of the SIP response code, for example: "417 Unknown Resource-Priority". ▪ If no reason text exists in the SIP response code, the description is taken from an internal SIP response mapping mechanism. For example, if the device receives a SIP response "422", it sends in the CDR "422 Session Interval Too Small method" as the description.
PstnTermReason	Q.850 protocol termination reason (0-127).
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

55.2.2.4 Release Reasons in CDR

The possible reasons for call termination for the Gateway / IP-to-IP application which is represented in the CDR field **TrmReason** are listed below:

- "REASON N/A"
- "RELEASE_BECAUSE_NORMAL_CALL_DROP"
- "RELEASE_BECAUSE_DESTINATION_UNREACHABLE"
- "RELEASE_BECAUSE_DESTINATION_BUSY"
- "RELEASE_BECAUSE_NOANSWER"
- "RELEASE_BECAUSE_UNKNOWN_REASON"
- "RELEASE_BECAUSE_REMOTE_CANCEL_CALL"
- "RELEASE_BECAUSE_UNMATCHED_CAPABILITIES"
- "RELEASE_BECAUSE_UNMATCHED_CREDENTIALS"
- "RELEASE_BECAUSE_UNABLE_TO_HANDLE_REMOTE_REQUEST"
- "RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT"
- "RELEASE_BECAUSE_CONFERENCE_FULL"
- "RELEASE_BECAUSE_VOICE_PROMPT_PLAY_ENDED"
- "RELEASE_BECAUSE_VOICE_PROMPT_NOT_FOUND"
- "RELEASE_BECAUSE_TRUNK_DISCONNECTED"
- "RELEASE_BECAUSE_RSRC_PROBLEM"
- "RELEASE_BECAUSE_MANUAL_DISC"
- "RELEASE_BECAUSE_SILENCE_DISC"
- "RELEASE_BECAUSE_RTP_CONN_BROKEN"
- "RELEASE_BECAUSE_DISCONNECT_CODE"
- "RELEASE_BECAUSE_GW_LOCKED"

- "RELEASE_BECAUSE_NORTEL_XFER_SUCCESS"
- "RELEASE_BECAUSE_FAIL"
- "RELEASE_BECAUSE_FORWARD"
- "RELEASE_BECAUSE_ANONYMOUS_SOURCE"
- "RELEASE_BECAUSE_IP_PROFILE_CALL_LIMIT"
- "GWAPP_UNASSIGNED_NUMBER"
- "GWAPP_NO_ROUTE_TO_TRANSIT_NET"
- "GWAPP_NO_ROUTE_TO_DESTINATION"
- "GWAPP_CHANNEL_UNACCEPTABLE"
- "GWAPP_CALL_AWARDED_AND "
- "GWAPP_PREEMPTION"
- "PREEMPTION_CIRCUIT_RESERVED_FOR_REUSE"
- "GWAPP_NORMAL_CALL_CLEAR"
- "GWAPP_USER_BUSY"
- "GWAPP_NO_USER_RESPONDING"
- "GWAPP_NO_ANSWER_FROM_USER_ALERTED"
- "MFCR2_ACCEPT_CALL"
- "GWAPP_CALL_REJECTED"
- "GWAPP_NUMBER_CHANGED"
- "GWAPP_NON_SELECTED_USER_CLEARING"
- "GWAPP_INVALID_NUMBER_FORMAT"
- "GWAPP_FACILITY_REJECT"
- "GWAPP_RESPONSE_TO_STATUS_ENQUIRY"
- "GWAPP_NORMAL_UNSPECIFIED"
- "GWAPP_CIRCUIT_CONGESTION"
- "GWAPP_USER_CONGESTION"
- "GWAPP_NO_CIRCUIT_AVAILABLE"
- "GWAPP_NETWORK_OUT_OF_ORDER"
- "GWAPP_NETWORK_TEMPORARY_FAILURE"
- "GWAPP_NETWORK_CONGESTION"
- "GWAPP_ACCESS_INFORMATION_DISCARDED"
- "GWAPP_REQUESTED_CIRCUIT_NOT_AVAILABLE"
- "GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED"
- "GWAPP_PERM_FR_MODE_CONN_OUT_OF_S"
- "GWAPP_PERM_FR_MODE_CONN_OPERATIONAL"
- "GWAPP_PRECEDENCE_CALL_BLOCKED"
    - "RELEASE_BECAUSE_PREEMPTION_ANALOG_CIRCUIT_RESERVED_FOR_REUSE"
    - "RELEASE_BECAUSE_PRECEDENCE_CALL_BLOCKED"
- "GWAPP_QUALITY_OF_SERVICE_UNAVAILABLE"
- "GWAPP_REQUESTED_FAC_NOT_SUBSCRIBED"
- "GWAPP_BC_NOT_AUTHORIZED"
- "GWAPP_BC_NOT_PRESENTLY_AVAILABLE"
- "GWAPP_SERVICE_NOT_AVAILABLE"
- "GWAPP_CUG_OUT_CALLS_BARRED"

- "GWAPP_CUG_INC_CALLS_BARRED"
- "GWAPP_ACCES_INFO_SUBS_CLASS_INCONS"
- "GWAPP_BC_NOT_IMPLEMENTED"
- "GWAPP_CHANNEL_TYPE_NOT_IMPLEMENTED"
- "GWAPP_REQUESTED_FAC_NOT_IMPLEMENTED"
- "GWAPP_ONLY_RESTRICTED_INFO_BEARER"
- "GWAPP_SERVICE_NOT_IMPLEMENTED_UNSPECIFIED"
- "GWAPP_INVALID_CALL_REF"
- "GWAPP_IDENTIFIED_CHANNEL_NOT_EXIST"
- "GWAPP_SUSPENDED_CALL_BUT_CALL_ID_NOT_EXIST"
- "GWAPP_CALL_ID_IN_USE"
- "GWAPP_NO_CALL_SUSPENDED"
- "GWAPP_CALL_HAVING_CALL_ID_CLEARED"
- "GWAPP_INCOMPATIBLE_DESTINATION"
- "GWAPP_INVALID_TRANSIT_NETWORK_SELECTION"
- "GWAPP_INVALID_MESSAGE_UNSPECIFIED"
- "GWAPP_NOT_CUG_MEMBER"
- "GWAPP_CUG_NON_EXISTENT"
- "GWAPP_MANDATORY_IE_MISSING"
- "GWAPP_MESSAGE_TYPE_NON_EXISTENT"
- "GWAPP_MESSAGE_STATE_INCONSISTENCY"
- "GWAPP_NON_EXISTENT_IE"
- "GWAPP_INVALID_IE_CONTENT"
- "GWAPP_MESSAGE_NOT_COMPATIBLE"
- "GWAPP_RECOVERY_ON_TIMER_EXPIRY"
- "GWAPP_PROTOCOL_ERROR_UNSPECIFIED"
- "GWAPP_INTERWORKING_UNSPECIFIED"
- "GWAPP_UKNOWN_ERROR"
- "RELEASE_BECAUSE_HELD_TIMEOUT"

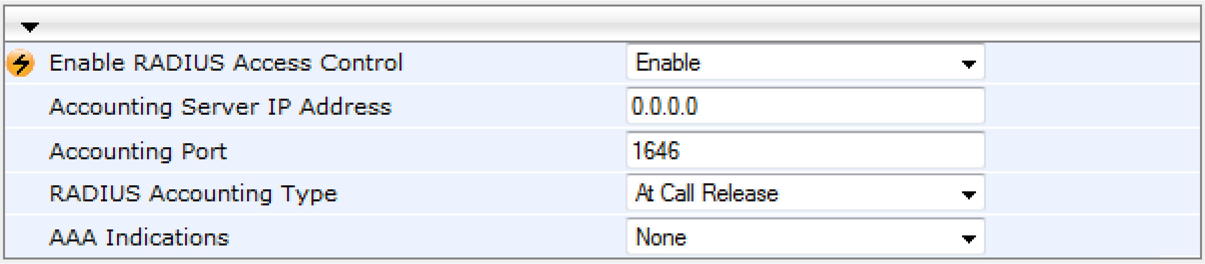
55.3 Configuring RADIUS Accounting

The device can send accounting data of SIP calls as call detail records (CDR) to a RADIUS Accounting server. The device can send the accounting messages to the RADIUS server upon call release, call connection and release, or call setup and release. For a list of the CDR attributes, see the table following the procedure below.

➢ **To configure RADIUS accounting:**

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

Figure 55-3: RADIUS Accounting Parameters Page



Enable RADIUS Access Control	Enable
Accounting Server IP Address	0.0.0.0
Accounting Port	1646
RADIUS Accounting Type	At Call Release
AAA Indications	None

2. Set the 'Enable RADIUS Access Control' parameter to **Enable**.
3. Configure the remaining parameters as required. For a description of these parameters, see "RADIUS Parameters" on page 1007.
4. Click **Submit**.
5. For your settings to take effect, reset the device with a flash burn.

The table below lists the RADIUS Accounting CDR attributes included in the communication packets transmitted between the device and a RADIUS server.

Table 55-5: Supported RADIUS Accounting CDR Attributes

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
Request Attributes						
1	user-name	-	Account number or calling party number or blank	String up to 15 digits long	5421385747	Start Acc Stop Acc
4	nas-ip-address	-	IP address of the requesting device	Numeric	192.168.14.43	Start Acc Stop Acc
6	service-type	-	Type of service requested	Numeric	1: login	Start Acc Stop Acc
26	h323-incoming-conf-id	1	SIP call identifier	Up to 32 octets	-	Start Acc Stop Acc
26	h323-remote-address	23	IP address of the remote gateway	Numeric	-	Stop Acc
26	h323-conf-id	24	H.323/SIP call identifier	Up to 32 octets	-	Start Acc Stop Acc
26	h323-setup-	25	Setup time in NTP	String	-	Start Acc

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
	time		format 1			Stop Acc
26	h323-call-origin	26	The call's originator: Answering (IP) or Originator (PSTN)	String	Answer, Originate etc	Start Acc Stop Acc
26	h323-call-type	27	Protocol type or family used on this leg of the call	String	VoIP	Start Acc Stop Acc
26	h323-connect-time	28	Connect time in NTP format	String	-	Stop Acc
26	h323-disconnect-time	29	Disconnect time in NTP format	String	-	Stop Acc
26	H323-Disconnect-Cause	30	Q.931 disconnect cause code	Numeric	-	Stop Acc
26	h323-gw-id	33	Name of the gateway	String	SIPIDString	Start Acc Stop Acc
26	sip-call-id	34	SIP Call ID	String	abcde@ac.com	Start Acc Stop Acc
26	call-terminator	35	The call's terminator: PSTN-terminated call (Yes); IP-terminated call (No).	String	Yes, No	Stop Acc
30	called-station-id	-	Destination phone number (Gateway / IP-to-IP call) or Destination URI (SBC call)	String	8004567145	Start Acc
31	calling-station-id	-	Calling Party Number (ANI) (Gateway call) or Source URI (SBC call)	String	5135672127	Start Acc Stop Acc
40	acct-status-type	-	Account Request Type (start or stop) **Note:** 'start' isn't supported on the Calling Card application.	Numeric	1: start, 2: stop	Start Acc Stop Acc
41	acct-delay-time	-	No. of seconds tried in sending a particular record	Numeric	5	Start Acc Stop Acc
42	acct-input-octets	-	Number of octets received for that call duration (Gateway call)	Numeric	-	Stop Acc

Attribute Number	Attribute Name	Vendor Specific Attribute (VSA) No.	Purpose	Value Format	Example	AAA
43	acct-output-octets	-	Number of octets sent for that call duration (Gateway call)	Numeric	-	Stop Acc
44	acct-session-id	-	A unique accounting identifier - match start & stop	String	34832	Start Acc Stop Acc
46	acct-session-time	-	For how many seconds the user received the service	Numeric	-	Stop Acc
47	acct-input-packets	-	Number of packets received during the call	Numeric	-	Stop Acc
48	acct-oputput-packets	-	Number of packets sent during the call	Numeric	-	Stop Acc
61	nas-port-type	-	Physical port type of device on which the call is active	String	0: Asynchronous	Start Acc Stop Acc
Response Attributes						
26	h323-return-code	103	The reason for failing authentication (0 = ok, other number failed)	Numeric	0 Request accepted	Stop Acc
44	acct-session-id	-	A unique accounting identifier – match start & stop	String	-	Stop Acc

Below is an example of RADIUS Accounting, where the non-standard parameters are preceded with brackets:

```
Accounting-Request (361)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2
acct-input-octets = 4841
acct-output-octets = 8800
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
```

55.4 Event Notification using X-Detect Header

The device can notify a remote party of the occurrence (or detection) of certain events in the media stream. The device detects events and notifies their occurrence, using the SIP X-Detect message header and only when establishing a SIP dialog.

Note: This feature is applicable only to the Gateway application (not SBC).

The table below lists the supported event types (and subtypes) and the corresponding device configurations that you need to do:

Table 55-6: Supported X-Detect Event Types

Events Type	Subtype	Required Configuration

CPT	SIT-NC
	SIT-IC
	SIT-VC
	SIT-RO
	Busy
	Reorder
	Ringtone
		SITDetectorEnable = 1
		UserDefinedToneDetectorEnable = 1
		Note: Ensure that the CPT file is configured with the required tone type.

FAX	CED	(IsFaxUsed ≠ 0) or (IsFaxUsed = 0, and FaxTransportMode ≠ 0)
	modem	VxxModemTransportType = 3
PTT	voice-start voice-end	EnableDSPIPMDetectors = 1

The device can detect and report the following Special Information Tones (SIT) types from the PSTN:

- SIT-NC (No Circuit found)
- SIT-IC (Operator Intercept)
- SIT-VC (Vacant Circuit - non-registered number)
- SIT-RO (Reorder - System Busy)

There are additional three SIT tones that are detected as one of the above SIT tones:

- The NC* SIT tone is detected as NC
- The RO* SIT tone is detected as RO
- The IO* SIT tone is detected as VC

The device can map these SIT tones to a Q.850 cause and then map them to SIP 5xx/4xx responses, using the parameters SITQ850Cause, SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO.

Table 55-7: Special Information Tones (SITs) Reported by the device

Special Information Tones (SITs) Name	Description	First Tone Frequency Duration		Second Tone Frequency Duration		Third Tone Frequency Duration	
		(Hz)	(ms)	(Hz)	(ms)	(Hz)	(ms)
NC1	No circuit found	985.2	380	1428.5	380	1776.7	380
IC	Operator intercept	913.8	274	1370.6	274	1776.7	380
VC	Vacant circuit (non registered number)	985.2	380	1370.6	274	1776.7	380
RO1	Reorder (system busy)	913.8	274	1428.5	380	1776.7	380
NC*	-	913.8	380	1370.6	380	1776.7	380
RO*	-	985.2	274	1370.6	380	1776.7	380
IO*	-	913.8	380	1428.5	274	1776.7	380

For example:

```
INFO sip:5001@10.33.2.36 SIP/2.0
Via: SIP/2.0/UDP 10.33.45.65;branch=z9hG4bKac2042168670
Max-Forwards: 70
From: <sip:5000@10.33.45.65;user=phone>;tag=1c1915542705
To: <sip:5001@10.33.2.36;user=phone>;tag=WQJNIDDPCOKAPIDSCOTG
Call-ID: AIFHPETLLMVVFWPDXUHD@10.33.2.36
CSeq: 1 INFO
Contact: <sip:2206@10.33.45.65>
Supported: em,timer,replaces,path,resource-priority
Content-Type: application/x-detect
Content-Length: 28
Type= CPT
SubType= SIT-IC
```

The X-Detect event notification process is as follows:

1. For IP-to-Tel or Tel-to-IP calls, the device receives a SIP request message (using the X-Detect header) that the remote party wishes to detect events on the media stream. For incoming (IP-to-Tel) calls, the request must be indicated in the initial INVITE and responded to either in the 183 response (for early dialogs) or in the 200 OK response (for confirmed dialogs).
2. Once the device receives such a request, it sends a SIP response message (using the X-Detect header) to the remote party, listing all supported events that can be detected. The absence of the X-Detect header indicates that no detections are available.
3. Each time the device detects a supported event, the event is notified to the remote party by sending an INFO message with the following message body:
   - Content-Type: application/X-DETECT
   - Type = [CPT | FAX | PTT…]
   - Subtype = xxx (according to the defined subtypes of each type)

Below is an example of SIP messages using the X-Detect header:

```
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=CPT,FAX
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=CPT,FAX
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=CPT,FAX
Content-Type: Application/X-Detect
Content-Length: xxx
Type = CPT
Subtype = SIT
```

55.5 Querying Device Channel Resources using SIP OPTIONS

The device reports its maximum and available channel resources in SIP 200 OK responses upon receipt of SIP OPTIONS messages. The device sends this information in the SIP X-Resources header with the following parameters:

- **telchs:** Specifies the total telephone channels and the number of free (available) telephone channels.
- **mediachs:** Not applicable.

Below is an example of the X-Resources:

```
X-Resources: telchs= 12/4;mediachs=0/0
```

In the example above, "telchs" specifies the number of available channels and the number of occupied channels (4 channels are occupied and 12 channels are available).

Note: This feature is applicable only to the Gateway / IP-to-IP application.

56 Obtaining Status and Performance using a USB Flash Drive

You can use a USB flash drive to obtain status and performance information of the device. For more information, see 'Automatic Provisioning using USB Flash Drive' on page 651.

This page is intentionally left blank.

Part XI

Diagnostics

57 Syslog and Debug Recordings

Syslog is an event notification protocol that enables a device to send event notification messages across IP networks to event message collectors, also known as Syslog servers. The device contains an embedded Syslog client, which sends error reports / events that it generates to a remote Syslog server using the IP / UDP protocol. This information is a collection of error, warning, and system messages that records every internal operation of the device.

For receiving Syslog messages generated by the device, you can use any of the following Syslog servers:

- **Device's embedded Syslog server:** The device provides an embedded Syslog server, which is accessed through the Web interface. This provides limited Syslog server functionality.
- **Wireshark:** Third-party network protocol analyzer (http://www.wireshark.org).
- **Third-party, Syslog server:** Any third-party Syslog server program that enables filtering of messages according to parameters such as priority, IP sender address, time, and date.

57.1 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 719).

Below is an example of a Syslog message:

```
13:10:57.811 : 10.13.4.12 : NOTICE  : [S=235][SID:1034099026] (
lgr_flow)(63        )  UdpTransportObject#0- Adding socket event
for address 10.33.2.42:5060 [Time: 04-19-2012@18:29:39]
```

Table 57-1: Syslog Message Format Description

Message Item	Description
Message Types	Syslog generates the following types of messages: ▪ ERROR: Indicates that a problem has been identified that requires immediate handling. ▪ WARNING: Indicates an error that might occur if measures are not taken to prevent it. ▪ NOTICE: Indicates that an unusual event has occurred. ▪ INFO: Indicates an operational message. ▪ DEBUG: Messages used for debugging. **Notes:** ▪ The INFO and DEBUG messages are required only for advanced debugging. Therefore, by default, they are not sent by the device. ▪ When viewing Syslog messages in the Web interface, these message types are color coded.
Message Sequence Number [S=<number>]	By default, Syslog messages are sequentially numbered in the format [S=<number>], for example, "[S=643]". A skip in the number sequence of messages indicates a loss of message packets. For example, in the below Syslog message, messages 238 through 300 were not received. In other words, 63 Syslog messages were lost (the sequential

Message Item	Description
	numbers are indicated below in bold font): 18:38:14. 52 : 10.33.45.72 : NOTICE: [**S=235**][SID:1034099026] (lgr_psbrdex)(619) recv <-- DIGIT(0) Ch:0 OnTime:0 InterTime:100 Direction:0 System:1 [File: Line:-1] 18:38:14. 83 : 10.33.45.72 : NOTICE: [**S=236**][SID:1034099026] (lgr_flow)(620) #0:DIGIT_EV [File: Line:-1] 18:38:14. 83 : 10.33.45.72 : NOTICE: [**S=237**][SID:1034099026] (lgr_flow)(621) | #0:DIGIT_EV [File: Line:-1] 18:38:14.958 : 10.33.45.72 : NOTICE: [**S=301**][SID:1034099026] (lgr_flow)(625) | #0:DIGIT_EV [File: Line:-1] You can disable the inclusion of the message sequence number in Syslog messages, by setting the 'CDR Session ID' parameter to **Disable** (see "Configuring Syslog" on page 719).
Log Number (lgr)(number)	Ignore this number; it has been replaced by the Message Sequence Number (described previously).
Session ID	Automatically assigned (random), unique session identifier (session-id / SID) number per call in the CDR of sent Syslog messages and debug recording packets. This enables you to filter the information (such as SIP, Syslog, and media) according to the SID. ▪ Gateway application: A call session is considered either as a Tel-to-IP leg or an IP-to-Tel leg, where each leg is assigned a unique SID. ▪ SBC application: A session is considered as both the outgoing and incoming legs, where both legs share the same SID. The benefit of this unique numbering is that it enables you to filter the information (such as SIP, Syslog, and media) according to a specific SID. **Note:** Forked legs and alternative legs share the same SID.
Message Body	Describes the message.
Timestamp	When the Network Time Protocol (NTP) is enabled, a timestamp string [hour:minutes:seconds] is added to all Syslog messages.

57.1.1 Event Representation in Syslog Messages

The Syslog message events that the device sends are represented by unique abbreviations. An example of an abbreviated event in a Syslog message indicating packet loss (PL) is shown below:

```
Apr  4 12:00:12 172.30.1.14 PL:5  [Code:3a002] [CID:3294] [Time:
20:17:00]
```

The table below lists these unique event abbreviations:

Table 57-2: Syslog Error Name Descriptions

Error Abbreviation	Error Name Description
AA	Invalid Accumulated Packets Counter
AC	Invalid Channel ID
AL	Invalid Header Length
AO	Invalid Codec Type
AP	Unknown Aggregation Payload Type
AR	Invalid Routing Flag Received
AT	Simple Aggregation Packets Lost
CC	Command Checksum Error
CE	Invalid Cell Coder Code
CS	Command Sequence Error
ES	8 sec Timeout Before Disconnect
HO	Host Received Overrun
IA	Invalid AMR Payload
IC	Invalid CID Error
IG	Invalid G723 Code
IP	Invalid payload length
IR	Invalid RTCP Packet
IS	Invalid SID Length
LC	Transmitter Received Illegal Command
LF	Lost Fax Frames In High Speed Mode
LM	Lost Modem Frames In High Speed Mode
MI	Misalignment Error
MR	Modem Relay Is Not Supported
OR	DSP JB Overrun
PH	Packet Header Error
PL	RTP Packet Loss
RB	Counts the number of BFI Frames Received From The Host
RD	No Available Release Descriptor
RO	RTP Reorder
RP	Unknown RTP Payload Type
RS	RTP SSRC Error
UF	Unrecognized Fax Relay Command
AA	Invalid Accumulated Packets Counter
AC	Invalid Channel ID

Error Abbreviation	Error Name Description
AL	Invalid Header Length
AO	Invalid Codec Type
AP	Unknown Aggregation Payload Type
AR	Invalid Routing Flag Received

57.1.2 Unique Device Identification in Syslog Messages

Syslog messages include the following unique string for the device:

- Syslog messages relating to VoIP functionality are marked with "host"; those relating to Data Routing are marked with "DATA", for example:

```
12/12 12:46:40.921 : 10.8.5.70 : NOTICE  : host:  10.8.5.78 (sip_stack)(24)
Resource SIPMessage deleted - #267
11/24 08:14:09.311 : 10.3.2.100 : WARNING : DATA: Failed to set device eth0
netmask: Cannot assign requested address
```

57.1.3 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 57-3: Syslog Facility Levels

Numerical Value	Facility Level
16 (default)	local use 0 (local0)
17	local use 1 (local1)
18	local use 2 (local2)
19	local use 3 (local3)
20	local use 4 (local4)
21	local use 5 (local5)
22	local use 6 (local6)
23	local use 7 (local7)

Syslog messages begin with a less-than ("**<**") character, followed by a number, which is followed by a greater-than ("**>**") character. This is optionally followed by a single ASCII space. The number is known as the *Priority* and represents both the Facility level and the Severity level. A Syslog message with Facility level 16 is shown below:

```
Facility: LOCAL0 - reserved for local use (16)
```

57.1.4 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 57-4: 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:

57.2 Enabling Syslog

The following procedure describes how to enable and configure Syslog.

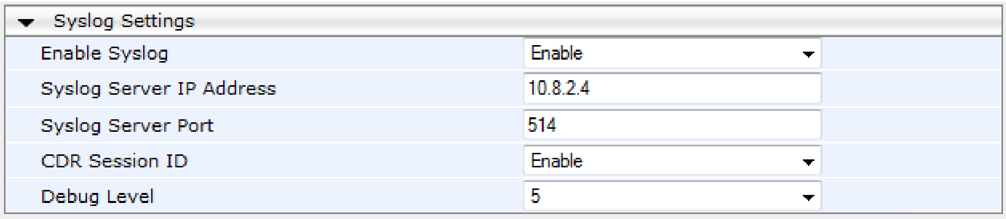
Notes:

- For configuring CDR reporting, see "Configuring CDR Reporting" on page 693.
- For viewing Syslog messages in the Web interface, see "Viewing Syslog Messages" on page 725.
- For a detailed description on the Syslog parameters, see "Syslog, CDR and Debug Parameters" on page 788.
- To configure the network interface (WAN or VoIP LAN OAMP) from where the device sends Syslog messages to a Syslog server, use the OampDefaultNetworkSource parameter.

➢ **To enable Syslog:**

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

Figure 57-1: Syslog Settings Page



Syslog Settings

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

2. Enable the Syslog feature by setting 'Enable Syslog' to **Enable**.
3. Define the Syslog server using the 'Syslog Server IP Address' and 'Syslog Server Port' parameters. If communication with the Syslog server is through the device's WAN interface, the Syslog server can be configured with an IPv6 address.
4. (Optional) Disable the inclusion of the Sequence Number in Syslog messages, by setting the 'CDR Session ID' parameter to **Disable**.
5. Configure the debug level using the 'Debug Level' parameter. This determines the level of messages that the device sends to the Syslog server. If set to Basic or Detailed, you can also configure related features using the following parameters:
   - SyslogOptimization: Enables the device to accumulate and bundle multiple debug messages into a single UDP packet and then send it to a Syslog server. The benefit of this feature is that it reduces the number of UDP Syslog packets, thereby improving (optimizing) CPU utilization. The size of the bundled message is configured by the MaxBundleSyslogLength parameter.
   - SyslogCpuProtection: Enables the protection of the device's CPU resources during debug reporting, ensuring voice traffic is unaffected. If CPU resources drop (i.e., high CPU usage) to a critical level (threshold), the device automatically lowers the debug level to free up CPU resources that were required for the previous debug-level functionality. When CPU resources become available again, the device increases the debug level. The threshold is configured by the DebugLevelHighThreshold parameter (see below).
   - DebugLevelHighThreshold: Defines the threshold (in percentage) for automatically switching to a different debug level, depending on CPU usage. For more information about this functionality, refer to the parameter's description in "Syslog, CDR and Debug Parameters" on page 788.
6. Click **Submit**.

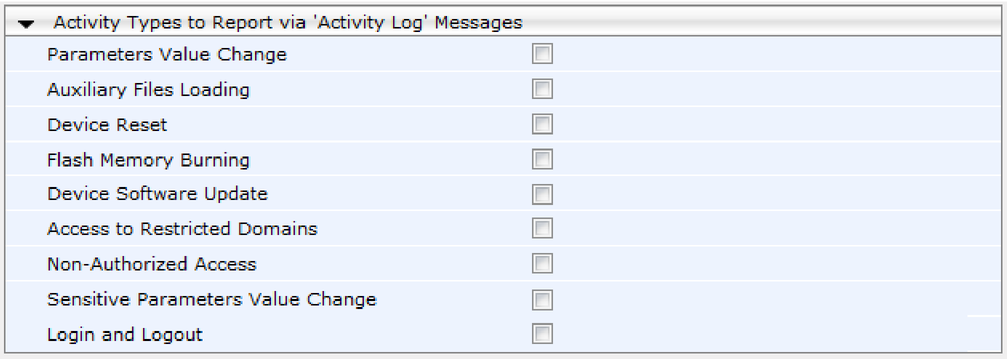
57.3 Configuring Web Operations to Report to Syslog

You can define the operations (activities) in the Web interface that must be reported to the Syslog server. The following procedure describes how to configure this in the Web interface. You can also configure this using the ini file parameter, ActivityListToLog or CLI command, config-system > logging > activity-log.

➢ To define Web activities to report to Syslog server:

1. Open the Syslog Settings page (**Configuration** tab > **System** menu > **Syslog Settings**).
2. Under the Activity Types to Report via Activity Log Messages group, select the Web actions to report to the Syslog server. For more information, see ''Syslog, CDR and Debug Parameters'' on page 788.

Figure 57-2: Web Activities to Report to Syslog



Activity Types to Report via 'Activity Log' Messages	
Parameters Value Change	☐
Auxiliary Files Loading	☐
Device Reset	☐
Flash Memory Burning	☐
Device Software Update	☐
Access to Restricted Domains	☐
Non-Authorized Access	☐
Sensitive Parameters Value Change	☐
Login and Logout	☐

3. Click **Submit**.

57.4 Configuring Debug Recording

The device enables you to activate debug recording and send debug recording packets to a defined capturing server. When the debug recording is activated, the device duplicates all messages that are sent and/or received by it and then sends them to an external IP address. The debug recording can be done for different types of traffic for example, RTP/RTCP, T.38, ISDN, CAS, 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.

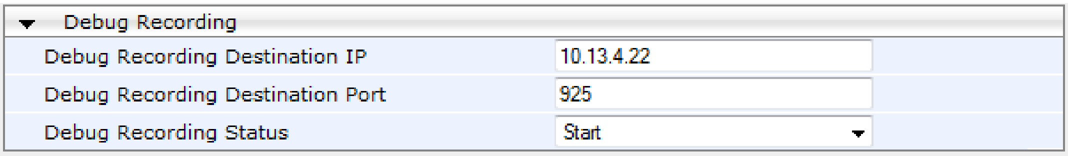
Notes:

- Debug recording is collected only on the device's OAMP interface.
- You can also save debug recordings to an external USB hard drive that is connected to the device's USB port. For more information, see USB Storage Capabilities on page 657.

➢ To configure and activate debug recording:

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

Figure 57-3: Logging Settings Page



Debug Recording	
Debug Recording Destination IP	10.13.4.22
Debug Recording Destination Port	925
Debug Recording Status	Start

2. Configure the debug capturing server using the 'Debug Recording Destination IP' and 'Debug Recording Destination Port' parameters.

3. From the 'Debug Recording Status' drop-down list, select **Start** to start the debug recording or **Stop** to end the recording.
4. Click **Submit**.

For a detailed description of these parameters, see "Syslog, CDR and Debug Parameters" on page 788.

57.5 Filtering Syslog Messages and Debug Recordings

The device can filter Syslog messages and debug recording (DR) packets, which are sent to a Syslog server and packet capturing application (such as Wireshark), respectively. Filtering can be useful to reduce CPU consumption and minimize negative impact on VoIP performance.

You can configure up to 30 filtering rules, each based on a selected filtering criteria (e.g., an IP Group). Each filtering criteria can be configured with a range. For example, you can filter Syslog messages for IP Groups 1 through 4. For each filter criteria, you can enable or disable Syslog messages and debug recording.

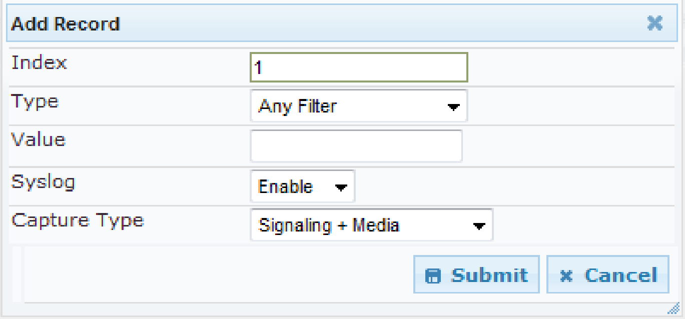
Debug recording can also be filtered using various filtering criteria such as SIP signaling or signaling and media.

The following procedure describes how to configure Logging Filter rules in the Web interface. You can also configure Logging Filter rules using the table ini file parameter, LoggingFilters or the CLI command configure system > logging > logging-filters.

➢ **To configure a logging filtering rule:**

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

Figure 57-4: Logging Filters Table - Add Record Dialog Box



Add Record

Index	1
Type	Any Filter
Value	
Syslog	Enable
Capture Type	Signaling + Media

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

Note: To configure the Syslog debug level, use the 'Debug Level' parameter (see "Enabling Syslog" on page 719).

Table 57-5: Logging Filters Table Parameter Descriptions

Parameter	Description
Index **[LoggingFilters_Index]**	Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.
Filter Type	Defines the filter type criteria.

Parameter	Description
CLI: filter-type **[LoggingFilters_FilterType]**	▪ **[1]** Any (default) ▪ [2] Trunk ID = Filters according to a specified Trunk ID (applicable only to the Gateway application) ▪ [3] Trunk Group ID = Filters according to a specified Trunk Group ID (applicable only to the Gateway application) ▪ [4] Trunk & B-channel = Filters according to a specified Trunk and B-channel (applicable only to the Gateway application) ▪ [5] FXS or FXO = Filters according to a specified FXS or FXO port. ▪ [6] Tel-to-IP = Filters according to a specified Tel-to-IP routing rule listed in the Outbound IP Routing table (applicable only to the Gateway application) ▪ [7] IP-to-Tel = Filters according to a specified IP-to-Tel routing rule listed in the Inbound IP Routing table (applicable only to the Gateway application) ▪ **[8]** IP Group = Filters according to a specified IP Group ID listed in the IP Group table ▪ [9] SRD = Filters according to a specified SRD ID listed in the SRD table ▪ [10] Classification = Filters according to a specified Classification rule listed in the Classification table (applicable only to the SBC application) ▪ [11] IP-to-IP Routing = Filters according to a specified SBC IP-to-IP routing rule listed in the IP-to-IP Routing table (applicable only to the SBC application) ▪ **[12]** User = Filters according to a specified user defined by username or user@host. ▪ **[13]** IP Trace = Filters according to a specified IP network trace wireshark-like expression. For a detailed description on configuring IP traces, see "Filtering IP Network Traces" on page 724.
Value CLI: value **[LoggingFilters_Value]**	Defines the value of the selected filtering type in the 'Filter Type' parameter. The value can be the following: ▪ A single value ▪ A range, using a hyphen "-" between the two values, e.g., "1-3" ▪ Multiple, non-contiguous values, using commas "," between each value, e.g., "1,3,9" ▪ Trunks/FXO/FXS pertaining to a module, using the syntax module number/port or port, for example: ✓ "1/2", means module 1, port 2 ✓ "1/[2-4]", means module 1, ports 2 through 4 ▪ **Any** to indicate all ▪ For IP trace expressions, see "Filtering IP Network Traces" on page 724
Syslog CLI: syslog **[LoggingFilters_Syslog]**	Enables Syslog messages for the defined logging filter: ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter is not applicable when 'Filter Type' is set to **IP Trace**.

Parameter	Description
Capture Type CLI: capture-type **[LoggingFilters_CaptureType]**	Enables debug recordings for the defined logging filter and defines what to record: ▪ **[0]** None (default) ▪ **[1]** Signaling = Information related to signaling such as SIP signaling messages, Syslog, CDR, and the device's internal processing messages. ▪ **[2]** Signaling & Media = Signaling and media (RTP/RTCP/T.38). ▪ **[3]** Signaling & Media & PCM = Signaling, media, and PCM (voice signals from and to TDM). ▪ [4] PSTN trace = ISDN and CAS traces - applicable only for Trunk-related filters. **Note:** This parameter is not applicable when 'Filter Type' is set to **IP Trace**.

57.5.1 Filtering IP Network Traces

You can filter Syslog and debug recording messages for IP network traces, by setting the 'Filter Type' parameter to **IP Trace** in the Logging Filters table. IP traces are used to record any IP stream, according to destination and/or source IP address, or port and Layer-4 protocol (UDP, TCP or any other IP type as defined by http://www.iana.com). Network traces are typically used to record HTTP.

When the **IP Trace** option is selected, only the ‘Value’ parameter is applicable; the 'Syslog' and 'Capture Type' parameters are not relevant. The 'Value' parameter configures the Wireshark-like filtering expressions for your IP trace. The following Wireshark-like expressions are supported:

Table 57-6: Supported Wireshark-like Expressions for 'Value' Parameter

Expression	Description
ip.src, ip.dst	Source and destination IP address
ip.addr	IP address - up to two IP addresses can be entered
ip.proto	IP protocol type (PDU) entered as an enumeration value (e.g., 1 is ICMP, 6 is TCP, 17 is UDP)
udp, tcp, icmp, sip, ldap, http, https	Single expressions for protocol type
udp.port, tcp.port	Transport layer
udp.srcport, tcp.srcport	Transport layer for source port
udp.dstport, tcp.dstport	Transport layer for destination port
and, &&, ==, <, >	Between expressions

Below are examples of configured expressions for the 'Value' parameter:

- udp && ip.addr==10.8.6.55
- ip.src==10.8.6.55 && udp.port>=5000 and udp.port<6000
- ip.dst==10.8.0.1/16
- ip.addr==10.8.6.40

For conditions requiring the "or" / "||" expression, add multiple table rows. For example, the Wireshark condition "(ip.src == 1.1.1.1 or ip.src == 2.2.2.2) and ip.dst == 3.3.3.3" can be configured using the following two table row entries:

1. ip.src == 1.1.1.1 and ip.dst == 3.3.3.3
2. ip.src == 2.2.2.2 and ip.dst == 3.3.3.3

> **Note:** If the 'Value' field is not defined, the device records all IP traffic types.

57.6 Viewing Syslog Messages

You can use the following tools to view the Syslog messages sent by the device:

- Web interface's Message Log page (see below).
- CLI -The device sends the error messages (e.g. Syslog messages) to the CLI console as well as to the original configured destination. Use the following commands:

  ```
  debug log              ; Starts the debug
  no debug log           ; Stops the debug
  no debug log all       ; Stops all debug process
  ```

- Any third-party Syslog server (e.g., Wireshark).

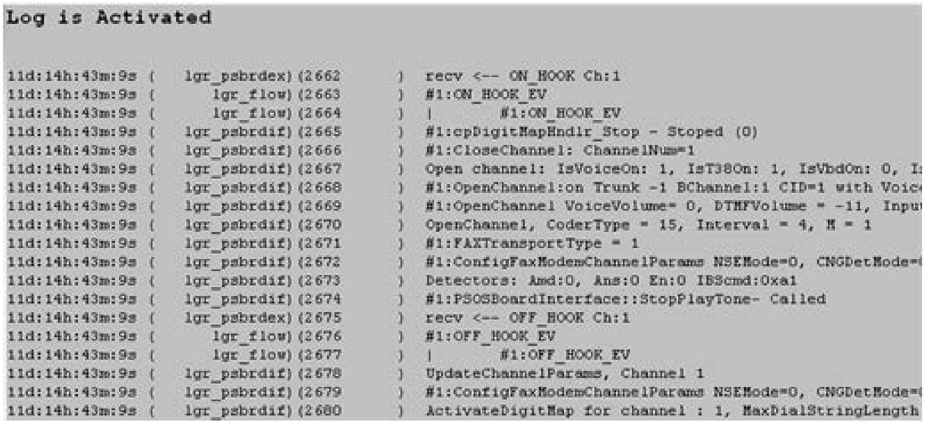
The following procedure describes how to view Syslog messages in the Web interface.

> **Notes:**
>
> - It's not recommended to keep a Message Log session open for a prolonged period. This may cause the device to overload. For prolonged (and detailed) debugging, use an external Syslog server.
> - You can select the Syslog messages in this page, and copy and paste them into a text editor such as Notepad. This text file (*txt*) can then be sent to AudioCodes Technical Support for diagnosis and troubleshooting.

➢ To activate the Web interface's Message Log:

1. Enable Syslog (see "Enabling Syslog" on page 719).
2. Open the Message Log page (**Status & Diagnostics** tab > **System Status** menu > **Message Log**); the Message Log page is displayed and the log is activated.

Figure 57-5: Message Log Page



```
Log is Activated

11d:14h:43m:9s (   lgr_psbrdex)(2662       )  recv <-- ON_HOOK Ch:1
11d:14h:43m:9s (      lgr_flow)(2663       )  #1:ON_HOOK_EV
11d:14h:43m:9s (      lgr_flow)(2664       )  |       #1:ON_HOOK_EV
11d:14h:43m:9s (   lgr_psbrdif)(2665       )  #1:cpDigitMapHndlr_Stop - Stoped (0)
11d:14h:43m:9s (   lgr_psbrdif)(2666       )  #1:CloseChannel: ChannelNum=1
11d:14h:43m:9s (   lgr_psbrdif)(2667       )  Open channel: IsVoiceOn: 1, IsT38On: 1, IsVbdOn: 0, I
11d:14h:43m:9s (   lgr_psbrdif)(2668       )  #1:OpenChannel:on Trunk -1 BChannel:1 CID=1 with Voic
11d:14h:43m:9s (   lgr_psbrdif)(2669       )  #1:OpenChannel VoiceVolume= 0, DTMFVolume = -11, Inpu
11d:14h:43m:9s (   lgr_psbrdif)(2670       )  OpenChannel, CoderType = 15, Interval = 4, M = 1
11d:14h:43m:9s (   lgr_psbrdif)(2671       )  #1:FAXTransportType = 1
11d:14h:43m:9s (   lgr_psbrdif)(2672       )  #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (   lgr_psbrdif)(2673       )  Detectors: Amd:0, Ans:0 En:0 IBScmd:0xa1
11d:14h:43m:9s (   lgr_psbrdif)(2674       )  #1:PSOSBoardInterface::StopPlayTone- Called
11d:14h:43m:9s (   lgr_psbrdex)(2675       )  recv <-- OFF_HOOK Ch:1
11d:14h:43m:9s (      lgr_flow)(2676       )  #1:OFF_HOOK_EV
11d:14h:43m:9s (      lgr_flow)(2677       )  |       #1:OFF_HOOK_EV
11d:14h:43m:9s (   lgr_psbrdif)(2678       )  UpdateChannelParams, Channel 1
11d:14h:43m:9s (   lgr_psbrdif)(2679       )  #1:ConfigFaxModemChannelParams NSEMode=0, CNGDetMode=
11d:14h:43m:9s (   lgr_psbrdif)(2680       )  ActivateDigitMap for channel : 1, MaxDialStringLength
```

The displayed logged messages are color-coded as follows:

- Yellow - fatal error message
- Blue - recoverable error message (i.e., non-fatal error)
- Black - notice message

➢ **To stop and clear the Message Log:**

- Close the Message Log page by accessing any another page in the Web interface.

57.7 Collecting Debug Recording Messages

To collect debug recording packets, use the open source program Wireshark. AudioCodes proprietary plug-in files for Wireshark are required.

Notes:

- The default debug recording port is 925. You can change the port in Wireshark (**Edit** menu > **Preferences** > **Protocols** > **AC DR**).
- The plug-in files are per major software release of Wireshark. For more information, contact your AudioCodes sales representative.
- The plug-in files are applicable only to Wireshark 32-bit for Windows.

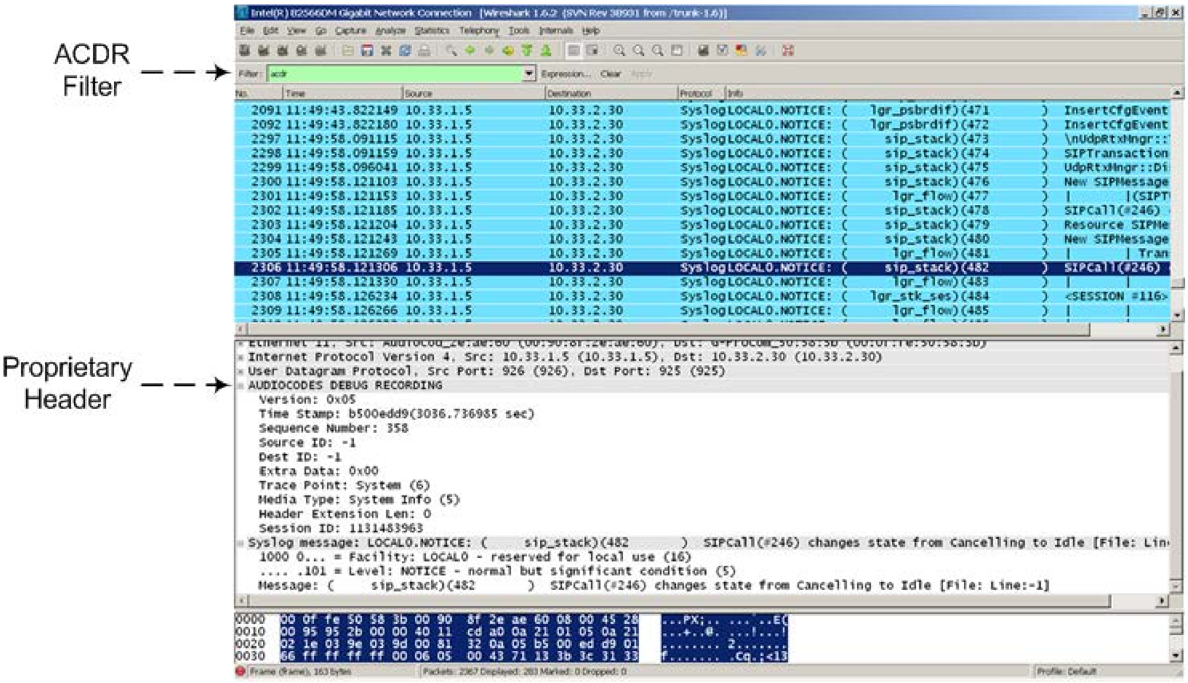
➢ **To install Wireshark and the plug-ins for debug recording:**

1. Install Wireshark on your computer. The Wireshark program can be downloaded from http://www.wireshark.org.
2. Download the proprietary plug-in files from 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
Intel(R) 82566DM Gigabit Network Connection [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: acdr
Expression... Clear Apply
Frame (frame), 163 bytes
Packets: 2367 Displayed: 283 Marked: 0 Dropped: 0
Profile: Default

```
No.  Time            Source       Destination  Protocol Info
2091 11:49:43.822149 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (   lgr_psbrdif)(471      )  InsertCfgEvent
2092 11:49:43.822180 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (   lgr_psbrdif)(472      )  InsertCfgEvent
2297 11:49:58.091115 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(473      )  \nUdpRtxMngr::
2298 11:49:58.091159 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(474      )  SIPTransaction
2299 11:49:58.096041 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(475      )  UdpRtxMngr::Di
2300 11:49:58.121103 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(476      )  New SIPMessage
2301 11:49:58.121153 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(477      )  |         |(SIPT
2302 11:49:58.121185 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(478      )  SIPCall(#246)
2303 11:49:58.121204 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(479      )  Resource SIPMe
2304 11:49:58.121243 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(480      )  New SIPMessage
2305 11:49:58.121269 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(481      )  |         | Tran
2306 11:49:58.121306 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (     sip_stack)(482      )  SIPCall(#246)
2307 11:49:58.121330 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(483      )  |         |
2308 11:49:58.126234 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (   lgr_stk_ses)(484      )  <SESSION #116>
2309 11:49:58.126266 10.33.1.5    10.33.2.30   Syslog LOCAL0.NOTICE: (      lgr_flow)(485      )  |         |

Ethernet II, Src: AudioCod_2e:ae:60 (00:90:8f:2e:ae:60), Dst: G-ProCom_50:58:3b (00:0f:fe:50:58:3b)
Internet Protocol Version 4, Src: 10.33.1.5 (10.33.1.5), Dst: 10.33.2.30 (10.33.2.30)
User Datagram Protocol, Src Port: 926 (926), Dst Port: 925 (925)
AUDIOCODES DEBUG RECORDING
  Version: 0x05
  Time Stamp: b500edd9(3036.736985 sec)
  Sequence Number: 358
  Source ID: -1
  Dest ID: -1
  Extra Data: 0x00
  Trace Point: System (6)
  Media Type: System Info (5)
  Header Extension Len: 0
  Session ID: 1131483963
Syslog message: LOCAL0.NOTICE: (     sip_stack)(482        )  SIPCall(#246) changes state from Cancelling to Idle [File: Lin
  1000 0... = Facility: LOCAL0 - reserved for local use (16)
  .... .101 = Level: NOTICE - normal but significant condition (5)
  Message: (     sip_stack)(482        )  SIPCall(#246) changes state from Cancelling to Idle [File: Line:-1]

0000  00 0f fe 50 58 3b 00 90  8f 2e ae 60 08 00 45 28   ...PX;.. ...`..E(
0010  00 95 95 2b 00 00 40 11  cd a0 0a 21 01 05 0a 21   ...+..@. ...!...!
0020  02 1e 03 9e 03 9d 00 81  32 0a 05 b5 00 ed d9 01   ........ 2.......
0030  66 ff ff ff ff 00 06 05  00 43 71 13 3b 3c 31 33   f....... .Cq.;<13
```

For ISDN trace messages, the additional header, "NetBricks Trace" is added below the "AUDIOCODES DEBUG RECORDING" header, as shown in the example below:

```
AUDIOCODES DEBUG RECORDING
NetBricks Trace
System time: 3559
     Direction: Message received from internal server queue (73)
From (Entity origination ID): DL_D (DL LAPD Q.921) (100)
     To (Entity destination ID): PH_D (D channel physical) (68)
     Primitive code: 67
     NAI (Network Access ID): 0 -> number of trunk
     SAPI: 1
     Connection ID: 0
     Congestion flag: 0
     Allocated message: 2
     Allocated buffer: 3
     Allocated timer cell: 141
     IT Message stack counter: 120
     IT Buffer stack counter: 120
     Message congestion counter: 0
     Buffer congestion counter: 0
     IT Stack message congestion counter: 0
     IT Stack buffer congestion counter: 0
     Pointer to message: 689
     Pointer to buffer: 0
     Data size: 33
     Link Access Procedure, Channel D (LAPD)
     Q.931
     Protocol discriminator: Q.931
     Call reference value length: 2
     Call reference flag: Message sent from originating side
     Call reference value: 0300 - > can be used as a filter to  identify entire ISDN call
     Message type: SETUP (0x05)
     Bearer capability
```

```
    Channel identification
    Calling party number: '201'
    Called party number: '102'
    Sending complete
```

For CAS trace messages, the additional header "CAS Trace" is added below the "AUDIOCODES DEBUG RECORDING" header, as shown in the example below:

```
AUDIOCODES DEBUG RECORDING
CAS Trace
    Timer: 1145504439
    From: DSP (0)
    Current State: 7
    Event: EV_DIAL_ENDED (15)
    Next State: -1
    Function Use: Unknown (-1)
    Parameter 1: -1
  Parameter 2: -1
    Parameter 3: -1
    Trunk Number: 3
    BChannel Number: 23
    Call Handle: 0
```

57.8 Debug Capturing VoIP and Data-Router Traffic

You can capture VoIP and data-router network traffic by sending traces to the CLI or to a file that is later sent to a FTP or TFTP server. The traffic can be captured using the following CLI commands to start the debug capture process:

- Data-router related debug capturing:

  ```
  # debug capture data interface <name> <slot/port> proto
  <protocol>  host <host> port <port> [ftp-server|tftp-server]
  <IP address>
  ```

  For example:

  ```
  # debug capture data interface GigabitEthernet 0/0 proto udp
  host any port any ftp-server 192.168.0.15
  ```

- Voice-related debug capturing:

  ```
  # debug capture voip interface <name> <slot/port> proto
  <protocol>  host <host> port <port> [ftp-server|tftp-server]
  <IP address>
  ```

  For example:

  ```
  # debug capture voip interface vlan 1 proto all host any port
  any ftp-server 10.4.2.58
  ```

57.9 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 capture can also be saved to a USB device. The maximum file size of debug captures that can be saved to the device is 20 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.

57.10 Configuring Termination of Debug Capture Upon Event

You can configure the device to stop a debug-traffic capture on the device's physical network interfaces upon a user-defined event. This event can be a Syslog message or an interface state-change. This feature supports all physical targets (TFTP, FTP, and USB), and SSH retrieval, as well as regular and cyclic-buffer modes. When combined with cyclic-buffer mode, this feature makes diagnosis of network problems easier.

To configure termination of debug capture upon a user-defined event or state change, use the following CLI commands:

- Defines the Syslog message event upon which the device stops the debug capture:

  ```
  # debug capture data physical auto-stop event syslog
  "<message>"
  ```

- Defines a state change on a specific interface upon which the device stops the debug capture:

  ```
  # debug capture data physical auto-stop event state-change
  <interface, e.g., GigabitEthernet 0/0>
  ```

- Defines a state change on any interface upon which the device stops the debug capture:

  ```
  # debug capture data physical auto-stop event state-change any
  ```

- Defines what to do with the debug capture when it is automatically stopped:

  ```
  # debug capture data physical auto-stop [send <IP
  address>|keep]
  ```

  Where:

  - send - sends the capture to the defined IP address
  - keep - saves the capture on the device for later retrieval

- Disables the automatic stopping feature for debug captures:

  ```
  # no debug capture data physical auto-stop
  ```

58 Self-Testing

The device features the following self-testing modes to identify faulty hardware components:

- **Detailed Test (Configurable):** This test verifies the correct functioning of the different hardware components on the device. This test is done when the device is taken out of service (i.e., not in regular service for processing calls). The test is performed on startup when initialization of the device completes.

  To enable this test, set the ini file parameter, EnableDiagnostics to 1 or 2, and then reset the device. Upon completion of the test and if the test fails, the device sends information on the test results of each hardware component to the Syslog server.

  The following hardware components are tested:

  - Analog interfaces - when EnableDiagnostics = 1 or 2

Notes:

- To return the device to regular operation and service, disable the test by setting the ini file parameter, EnableDiagnostics to 0, and then reset the device.
- While the test is enabled, ignore errors sent to the Syslog server.

- **Startup Test (automatic):** This hardware test has minor impact in real-time. While this test is executed, the regular operation of the device is disabled. If an error is detected, an error message is sent to the Syslog.

This page is intentionally left blank.

59 Creating Core Dump and Debug Files upon Device Crash

For debugging purposes, you can create a core dump file and/or debug file. These files may help you easily identify the cause of the crash. The core dump can either be included in or excluded from the debug file, or alternatively, sent separately to a TFTP server. The files can then be sent to AudioCodes support team for troubleshooting.

- **Core Dump File:** You can enable the device to send a core dump file to a remote destination upon a device crash. The core dump is a copy of the memory image at the time of the crash. It provides a powerful tool for determining the root cause of the crash. When enabled, the core dump file is sent to a user-defined TFTP server (IP address). If no address is configured, the core dump file is saved to the device's flash memory (if it has sufficient memory). The core dump file is saved as a binary file in the following name format: "core_<device name>_ver_<firmware version>_mac_<MAC address>_<date>_<time>", for example, core_acMediant_ver_680-8-4_mac_00908F099096_1-11-2014_3-29-29.
- **Debug File:** You can manually retrieve the debug file from the device and save it to a folder on your local PC. The debug file contains the following information:
  - Exception information, indicating the specific point in the code where the crash occurred.
  - Latest log messages that were recorded prior to the crash.
  - Core dump (only if enabled, no IP address has been defined, and the device has sufficient memory on its flash).
  - May include additional application-proprietary debug information.

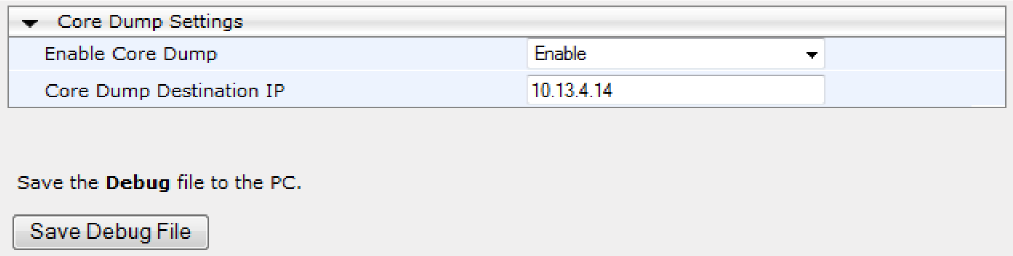
  The debug file is saved as a zipped file in the following name format: "debug_<device name>_ver_<firmware version>_mac_<MAC address>_<date>_<time>", for example, debug_acMediant_ver_680-8-4_mac_00908F099096_1-11-2014_3-29-29.

The following procedure describes how to configure core dump file creation in the Web interface.

➢ **To enable core dump creation:**

1. Open the Debug Utilities page (**Maintenance** tab > **Maintenance** menu > **Debug Utilities**).

Figure 59-1: 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.

2. From the 'Enable Core Dump' drop-down list, select **Enable**.
3. In the 'Core Dump Destination IP' field, enter an IP address of the remote server to where you want the file to be sent (optional).
4. Click **Submit**.

The following procedure describes how to retrieve the debug file from the device in the Web interface.

➢ **To save the debug file from the device:**

- In the Debug Utilities page, click the **Save Debug File** button.

60 Re-initializing Device with "Purified" Configuration

You can apply a "purified" version of the current configuration to enable proper functioning of the device. This is useful when the device has correct configuration, but for some or other reason it does not function properly. This may be attributed to accumulated "mess" due to lengthy and numerous configurations. Thus, this feature enables the device to do a "fresh-and-clean" start with the current configuration.

➢ **To re-initialize the device with a "purified" configuration:**

- In the CLI, type the following command:

  ```
  # copy startup-script from running-config
  ```

  When this command is run, the device 1) creates a CLI script file of the current configuration, 2) restores to factory defaults, 3) undergoes a reset, 4) applies (loads) the script file, and then 5) resets again (if required) for configuration settings to take effect.

This page is intentionally left blank.

61 Analog Line Testing

61.1 FXO Line Testing

The device can test the telephone lines connected to its FXO ports, using the SNMP acAnalogFxoLineTestTable table. These tests provide various line measurements. In addition to these tests, a keep-alive test is also done every 100 msec on each of the analog ports to detect communication problems with the analog equipment:

- Line Current (mA)
- Line Voltage (V)
- Hook (0 = on-hook; 1 = off-hook)
- Ring (0 - Off; 1 - On)
- Line Connected (0 = Disconnected; 1 = Connected)
- Polarity state (0 = Normal; 1 = Reversed, 2 = N\A)
- Line polarity (0 = Positive; 1 = Negative)
- Message Waiting Indication (0 = Off; 1 = On)

Note: Use the Analog Line testing mechanism only for monitoring and never when there are calls in progress.

61.2 FXS Line Testing

The device can test the telephone lines that are connected to its FXS ports, using the SNMP acAnalogFxsLineTestTable table. These tests provide various line measurements. In addition to these tests, a keep-alive test is also done every 100 msec on each of the analog ports to detect communication problems with the analog equipment and overheating of the FXS ports.

- Hardware revision number
- Temperature (above or below limit, only if a thermometer is installed)
- Hook state
- Coefficients checksum
- Message waiting indication status
- Ring state
- Reversal polarity state

Note: Use the Analog Line testing mechanism only for monitoring, and never when there are calls in progress.

This page is intentionally left blank.

62 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).

62.1 Configuring Test Call Endpoints

The Test Call table lets you test the SIP signaling (setup and registration) and media (DTMF signals) of calls between a simulated phone on the device and a remote endpoint. These tests involve both incoming and outgoing calls, where the test endpoint can be configured as the caller or called party. Test calls can be dialed automatically at a user-defined interval and/or manually when required. The simulated phone and remote endpoints are defined as SIP URIs (user@host) and the remote destination can be defined as an IP Group, IP address, or according to an Outbound IP Routing rule. You can also enable automatic registration of the endpoint.

When a SIP test call is initiated, the device generates a SIP INVITE towards the remote endpoint (e.g., a SIP proxy server or softswitch). It simulates the SIP call setup process, managing SIP 1xx responses and completing the SIP transaction with a 200 OK.

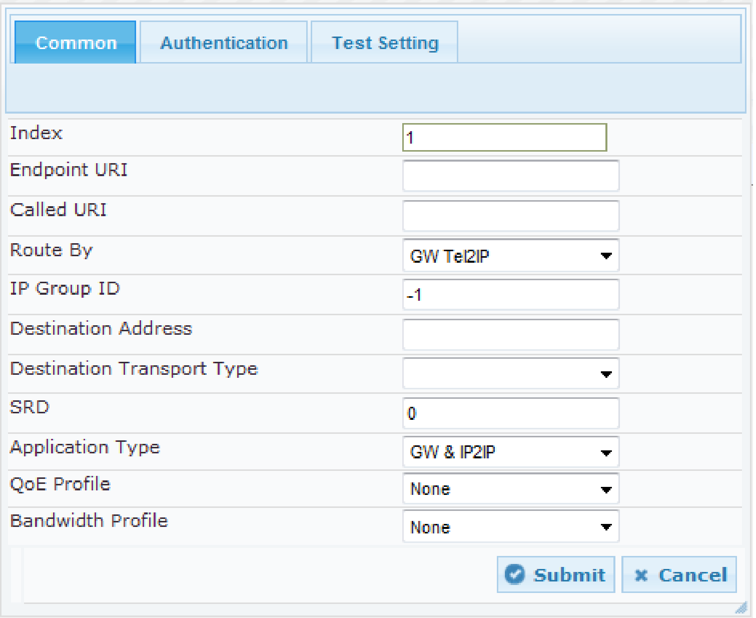
Note: By default, you can configure up to five test calls. However, this number can be increased by installing the relevant Software License Key. For more information, contact your AudioCodes sales representative.

The following procedure describes how to configure test calls in the Web interface. You can also configure this using the table ini file parameter, Test_Call or CLI command, configure system > test-call > test-call-table.

➢ To configure a test call:

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

Figure 62-1: General Tab of Test Call Table



Index	1
Endpoint URI	
Called URI	
Route By	GW Tel2IP
IP Group ID	-1
Destination Address	
Destination Transport Type	
SRD	0
Application Type	GW & IP2IP
QoE Profile	None
Bandwidth Profile	None

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

Table 62-1: Test Call Table Parameter Descriptions

Parameter	Description
General Tab	
Endpoint URI CLI: endpoint-uri **[Test_Call_EndpointURI]**	Defines the endpoint's URI. This can be defined as a user or user@host. The device identifies this endpoint only by the URI's user part. The URI's host part is used in the SIP From header in REGISTER requests. The valid value is a string of up to 150 characters. By default, this parameter is not configured.
Called URI CLI: called-uri **[Test_Call_CalledURI]**	Defines the destination (called) URI (user@host). The valid value is a string of up to 150 characters. By default, this parameter is not configured.

Parameter	Description
Route By CLI: route-by **[Test_Call_RouteBy]**	Defines the type of routing method. This applies to incoming and outgoing calls. ▪ **[0]** GW Tel2IP = (Default) Calls are matched by (or routed to) an SRD and Application type (defined in the SRD and Application Type parameters below). ▪ **[1]** IP Group = Calls are matched by (or routed to) an IP Group ID. ▪ **[2]** Dest Address = Calls are matched by (or routed to) an SRD and application type. **Notes:** ▪ For REGISTER messages, the option [0] cannot be used as the routing method. ▪ For REGISTER messages, if option [1] is used, only Server-type IP Groups can be used.
IP Group ID CLI: ip-group-id **[Test_Call_IPGroupID]**	Defines the IP Group ID to which the test call is sent or from which it is received. **Notes:** ▪ This parameter is applicable only if option [1] is configured for the 'Route By' parameter. ▪ This IP Group is used for incoming and outgoing calls.
Destination Address CLI: dst-address **[Test_Call_DestAddress]**	Defines the destination host. This can be defined as an IP address[:port] or DNS name[:port]. **Note:** This parameter is applicable only if the 'Route By' parameter is set to [2] (Dest Address).
Destination Transport Type CLI: dst-transport **[Test_Call_DestTransportType]**	Defines the transport type for outgoing calls. ▪ **[-1]** = Not configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** This parameter is applicable only if the 'Route By' parameter is set to [2] (Dest Address).
SRD CLI: srd [Test_Call_SRD]	Defines the SRD for the endpoint. The default is SRD 0. **Note:** This parameter is applicable only if the 'Route By' parameter is set any option except [1] (IP Group).
Application Type CLI: application-type **[Test_Call_ApplicationType]**	Defines the application type for the endpoint. This, in effect, associates the IP Group and SRD to a specific SIP interface. For example, assume two SIP Interfaces are configured in the SIP Interface table where one is set to "GW & IP2IP" and one to "SBC" for the 'Application Type'. If this parameter is set ot "SBC", the device uses the SIP Interface set to "SBC". ▪ [0] GW & IP2IP (default) ▪ [2] SBC
QoE Profile CLI: qoe-profile **[Test_Call_QOEProfile]**	Assigns a QoE Profile to the test call. To configure QoE Profiles, see "Configuring Quality of Experience Profiles" on page 248.

Parameter	Description
Bandwidth Profile CLI: bandwidth-profile **[Test_Call_BWProfile]**	Assigns a Bandwidth Profile to the test call. To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 252.
Authentication Tab **Note:** These parameters are applicable only if the Call Party parameter is set to **Caller.**	
Auto Register CLI: auto-register **[Test_Call_AutoRegister]**	Enables automatic registration of the endpoint. The endpoint can register to the device itself or to the 'Destination Address' or 'IP Group ID' parameter settings (see above). ▪ **[0]** Disable (default) ▪ **[1]** Enable
Username CLI: user-name **[Test_Call_UserName]**	Defines the authentication username. By default, no username is defined.
Password CLI: password **[Test_Call_Password]**	Defines the authentication password. By default, no password is defined.
Test Settings Tab	
Call Party CLI: call-party **[Test_Call_CallParty]**	Defines whether the test endpoint is the initiator or receiving side of the test call. ▪ **[0]** Caller (default) ▪ **[1]** Called
Maximum Channels for Session CLI: max-channels **[Test_Call_MaxChannels]**	Defines the maximum number of concurrent channels for the test session. For example, if you have configured an endpoint "101" and you set this parameter to "3", the device automatically creates three simulated endpoints - "101", "102" and "103" (i.e., consecutive endpoint URIs are assigned). The default is 1.
Call Duration CLI: call-duration **[Test_Call_CallDuration]**	Defines the call duration (in seconds). The valid value is -1 to 100000. The default is 20. A value of 0 means infinite. A value of -1 means that the parameter value is automatically calculated according to the values of the 'Calls per Second' and 'Maximum Channels for Session' parameters. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Calls per Second CLI: calls-per-second **[Test_Call_CallsPerSecond]**	Defines the number of calls per second. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.

Parameter	Description
Test Mode CLI: test-mode **[Test_Call_TestMode]**	Defines the test session mode. ▪ **[0]** Once = (Default) The test runs until the lowest value between the following is reached: ✓ Maximum channels is reached for the test session, configured by 'Maximum Channels for Session'. ✓ Call duration ('Call Duration') multiplied by calls per second ('Calls per Second'). ✓ Test duration expires, configured by 'Test Duration'. ▪ **[1]** Continuous = The test runs until the configured test duration is reached. If it reaches the maximum channels configured for the test session (in the 'Maximum Channels for Session'), it waits until the configured call duration of a currently established tested call expires before making the next test call. In this way, the test session stays within the configured maximum channels. **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Test Duration CLI: test-duration **[Test_Call_TestDuration]**	Defines the test duration (in minutes). The valid value is 0 to 100000. The default is 0 (i.e., unlimited). **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.
Play CLI: play **[Test_Call_Play]**	Enables and defines the playing of a tone to the answered side of the call. ▪ **[0]** Disable ▪ **[1]** DTMF (default) = Plays a user-defined DTMF string, configured in "Configuring DTMF Tones for Test Calls" on page 746. ▪ **[2]** PRT = Plays a non-DTMF tone from the PRT file (Dial Tone 2). For this option, a PRT file must be loaded to the device (see "Prerecorded Tones File" on page 604). **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" on page 192). ▪ This parameter is applicable only if 'Call Party' is set to **Caller**.
Schedule Interval CLI: schedule-interval **[Test_Call_ScheduleInterval]**	Defines the interval (in minutes) between automatic outgoing test calls. The valid value range is 0 to 100000. The default is 0 (i.e., scheduling is disabled). **Note:** This parameter is applicable only if 'Call Party' is set to **Caller**.

62.2 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 744).

62.3 Viewing Test Call Statistics

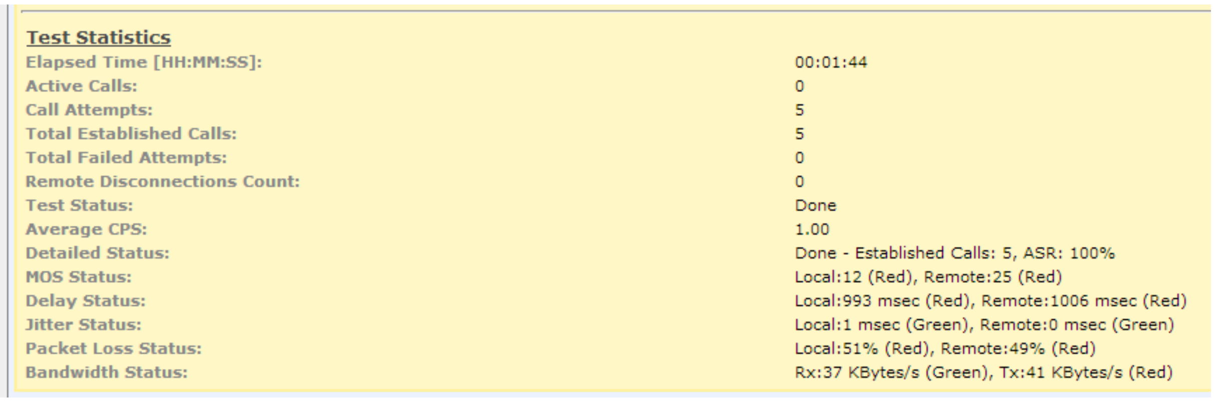
In addition to viewing a brief status description of the test call in the 'Test Status' field (as described in "Starting, Stopping and Restarting Test Calls" on page 744), you can also view a more detailed status description which includes test call statistics.

➢ **To view statistics of a test call:**

1. Open the Test Call Table page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Table**).
2. Select the test call table entry whose call statistics you want to view.

3. Click the **Show/Hide** button; the call statistics are displayed in the **Test Statistics** pane located below the table, as shown below:

Figure 62-2: 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 744).
- **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.

62.4 Configuring DTMF Tones for Test Calls

By default, no DTMF signal is played to an answered test call (incoming or outgoing). However, you can enable this per configured test call in the Test Call table (see "Configuring Test Call Endpoints" on page 739). If enabled, the default DTMF signal that is played is "3212333". You can change this as described below.

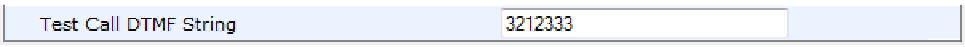
Notes:

- The DTMF signaling type (e.g., out-of-band or in-band) can be configured using the 'DTMF Transport Type' parameter. For more information, see "Dual-Tone Multi-Frequency Signaling" on page 192.
- 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 62-3: DTMF in Test Call Settings Page



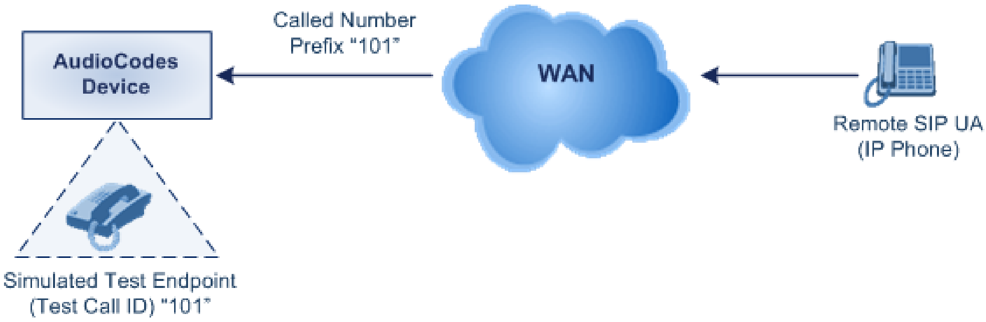
Test Call DTMF String 3212333

2. In the 'Test Call DTMF String' field, enter the DTMF string (up to 15 digits).
3. Click **Submit**.

62.5 Configuring Basic Test Call

The Basic Test Call feature tests incoming Gateway calls from a remote SIP endpoint 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. All incoming calls with this called (destination) prefix number is identified as a test call and sent to the simulated endpoint. The figure below displays a basic test call example.

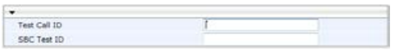
Figure 62-4: Incoming Test Call Example


AudioCodes Device
Called Number Prefix "101"
WAN
Remote SIP UA (IP Phone)
Simulated Test Endpoint (Test Call ID) "101"

➢ To configure basic call testing:

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

Figure 62-5: Test Call Settings Page



Test Call ID	
SBC Test ID	

2. In the 'Test Call ID' field, enter a prefix for the simulated endpoint.
3. Click **Submit**.

Notes:

- The Basic Test Call feature tests incoming calls only and is initiated only upon receipt of incoming calls with the configured prefix.
- For a full description of this parameter, see "SIP Test Call Parameters" on page 787.
- This call test is done on all SIP interfaces.
- This call test is applicable only to the Gateway application.

62.6 Configuring SBC Test Call with External Proxy

The SBC Test Call feature tests incoming SBC SIP call flow between a simulated test endpoint on the device and a remote SIP endpoint, when registration and routing is done through an external proxy/registrar server such as a hosted IP PBX in the WAN. In other words, the complete SIP flow, including the path to/from the external proxy/registrar can be tested.

Notes:

- The SBC Test Call feature is initiated only upon receipt of incoming calls and with the configured prefix.
- This call test is done on all SIP interfaces.

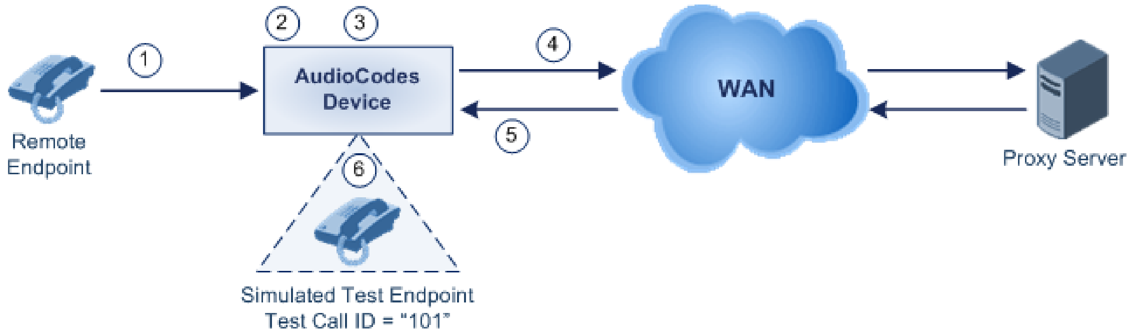
As this test call type involves an SBC call, you need to configure regular SBC rules such as classification and IP-to-IP routing. Therefore, this test call also allows you to verify correct SBC configuration.

For this test call, you also need to configure the following call IDs:

- Test Call ID - prefix number of the simulated endpoint on the device.
- SBC Test ID - prefix number of called number for identifying incoming call as SBC test call. The device removes this prefix, enabling it to route the call according to the IP-to-IP Routing rules to the external proxy/registrar, instead of directly to the simulated endpoint. Only when the device receives the call from the proxy/registrar, does it route the call to the simulated endpoint.

The figure below displays an example of an SBC test call:

Figure 62-6: SBC Test Call Example

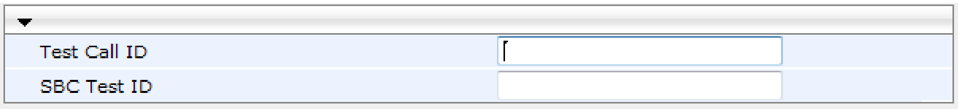


1. The call is received from the remote endpoint with the called number prefix "8101".
2. As the 'SBC Test ID' parameter is set to "8", the device identifies this call as a test call and removes the digit "8" from the called number prefix, leaving it as "101".
3. The device performs the regular SBC processing such as classification and manipulation.
4. The device routes the call, according to the configured SBC IP-to-IP routing rules, to the proxy server.
5. The device receives the call from the proxy server.
6. As the 'Test Call ID' parameter is set to "101", the device identifies the incoming call as a test call and sends it directly to the simulated test endpoint "101".

➢ To configure SBC call testing:

1. Configure the test call parameters (for a full description, see "SIP Test Call Parameters" on page 787):
   a. Open the Test Call Settings page (**Configuration** tab > **System** menu > **Test Call** > **Test Call Settings**).

Figure 62-7: Test Call Settings Page



Test Call ID	
SBC Test ID	

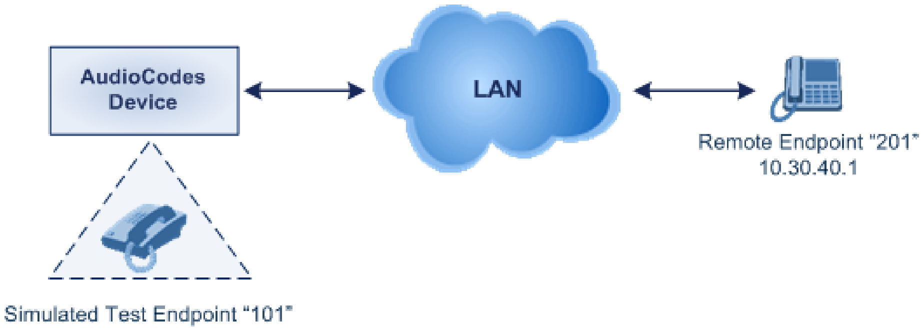
   b. In the 'Test Call ID' field, enter a prefix number for the simulated test endpoint on the device.
   c. In the 'SBC Test ID' field, enter a called prefix number for identifying the call as an SBC test call.
   d. Click **Submit**.
2. Configure regular SBC call processing rules for called number prefix "101", such as classification and IP-to-IP routing through a proxy server.

62.7 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 62-8: Single Test Call Example



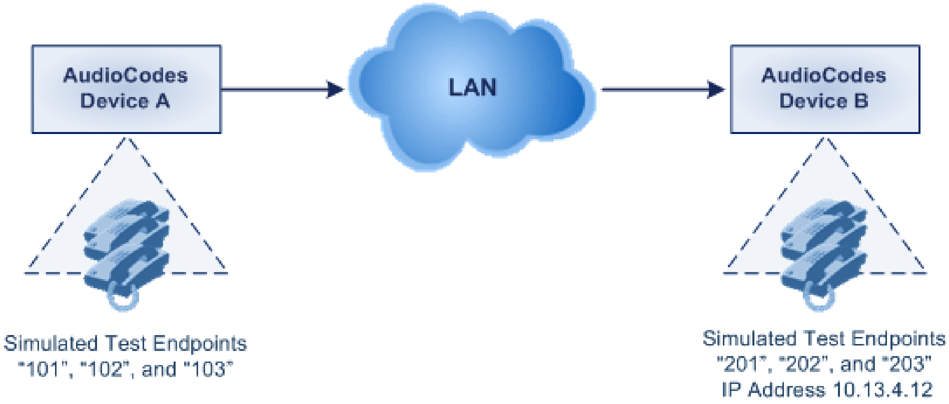
- Test Call table configuration:
  - Endpoint URI: "101"
  - Called URI: "201"
  - Route By: **Dest Address**
  - Destination Address: "10.30.40.01"
  - Call Party: **Caller**
  - Test Mode: **Once**

Alternatively, if you want to route the test call using the Outbound IP Routing table for the Gateway / IP-to-IP application, configure the following:

- Test Call table configuration:
  - Endpoint URI: 101@10.0.0.1
  - Route By: GW Tel2IP
  - Called URI: 201@10.30.40.1

    - Call Party: Caller
  - Outbound IP Routing table configuration:
    - Dest. Phone Prefix: 201 (i.e., the Called URI user-part)
    - Source Phone Prefix: 101 (i.e., the Endpoint URI user-part)
    - Dest. IP Address: 10.30.40.1
- **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.

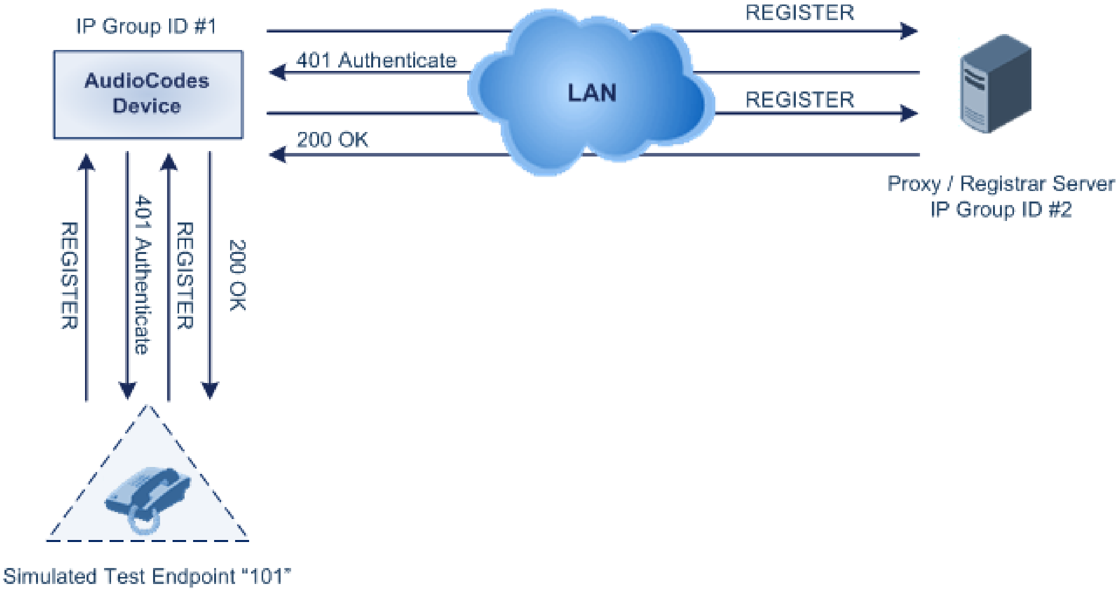
Figure 62-9: Batch Test Call Example


AudioCodes Device A
LAN
AudioCodes Device B
Simulated Test Endpoints "101", "102", and "103"
Simulated Test Endpoints "201", "202", and "203" IP Address 10.13.4.12

  - Test Call table configuration at Device A:
    - Endpoint URI: "101"
    - Called URI: "201"
    - Route By: **Dest Address**
    - Destination Address: "10.13.4.12"
    - Call Party: **Caller**
    - Maximum Channels for Session: "3" (configures three endpoints - "101", "102" and "103)
    - Call Duration: "5" (seconds)
    - Calls per Sec: "1"
    - Test Mode: **Continuous**
    - Test Duration: "3" (minutes)
    - Schedule Interval: "180" (minutes)
  - Test Call table configuration at Device B:
    - Endpoint URI: "201"
    - Maximum Channels for Session: "3" (configures three endpoints - "201", "202" and "203)

- **Registration Test Call Scenario:** This example describes the configuration for testing the registration and authentication (i.e., username and pas,sword) process of a simulated test endpoint on the device with an external proxy/registrar server. This is useful, for example, for verifying that endpoints located in the LAN can register with an external proxy and subsequently, communicate with one another.

Figure 62-10: Test Call Registration Example



This example assumes that you have configured your device for communication between LAN phone users such as IP Groups to represent the device (10.13.4.12) and the proxy server, and IP-to-IP routing rules to route calls between these IP Groups.

- Test Call table configuration:
  - Endpoint URI: "101"
  - Called URI: "itsp"
  - Route By: **Dest Address**
  - Destination Address: "10.13.4.12" (this is the IP address of the device itself)
  - Auto Register: **Enable**
  - User Name: "testuser"
  - Password: "12345"
  - Call Party: **Caller**

This page is intentionally left blank.

63 Data-Router Debugging

# ping <ip address> source voip interface [interface name] repeat size

[YL] in ipv4 you can ping from both voip or data interfaces

63.1 Loopback on WAN Interface Debugging

You can perform loopback testing on the WAN interface for debugging purposes. Loopback debugging can be activated on any WAN interface (name or type).

To perform loopback testing, use the following CLI command:

```
# debug ethernet loopback interface <interface name / type>
```

For example:

```
# debug ethernet loopback interface GigabitEthernet 0/0
```

The no debug command disables the loopback test.

> **Note:** All communication through the loopback WAN interface stops when this test is enabled.

63.2 Performing a Traceroute

You can use traceroute as a diagnostic tool for displaying the route (path) and measuring transit delays of packets across an IP network. The traceroute sends three requests to each hop on the way to the destination. Traceroute is done using the following CLI command:

```
# traceroute ipv6 <X:X::X:X> [vrf <vrf name>]
# traceroute <A.B.C.D or hostname> [vrf <vrf name>]
```

Examples:

```
# traceroute ipv6 2014:6666::dddd
1  2014:7777::aa55 (2014:7777::aa55)  2.421 ms  2.022 ms  2.155 ms
2  2014:6666::dddd (2014:6666::dddd)  2.633 ms  2.481 ms  2.568 ms
Traceroute: Destination reached
# traceroute 10.3.0.2
1  1 (10.4.0.1)  2.037 ms  3.665 ms  1.267 ms
2  1 (10.3.0.2)  1.068 ms  0.796 ms  1.070 ms
Traceroute: Destination reached
```

64 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 or data-router 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
  ```

- IPv6: The ping to an IPv6 address can be done only from a data-router interface and that is configured with an IPv6 address. The ping is done using the following CLI command:

  ```
  # ping ipv6 <IPv6 address or host name> source data [vrf | source-address interface|
  interface] [size <max. IP packet size>] [repeat <1-300>]
  ```

For a complete description of the ping command, refer to the *CLI Reference Guide*.

Note: IPv6 ping is currently only supported on Ethernet and Fiber interfaces.

This page is intentionally left blank.

65 Troubleshooting using a USB Flash Drive

You can use a USB flash drive for quick-and-easy troubleshooting of the device. For more information, see Automatic Provisioning using USB Flash Drive on page 651.

This page is intentionally left blank.

Part XII

Appendix

66 Dialing Plan Notation for Routing and Manipulation

The device supports flexible dialing plan notations for denoting the prefix and/or suffix source and/or destination numbers and SIP URI user names in the routing and manipulation tables.

Table 66-1: 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: ✓ **[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.

Notation	Description
	✓ 03(5xx): for any number that starts with 03 and ends with 5xx. ✓ 03(400,401,405): for any number that starts with 03 and ends with 400 or 401 or 405. **Notes:** ▪ The value *n* must be less than the value *m*. ▪ Only numerical ranges are supported (not alphabetical letters). ▪ For suffix ranges, the starting (*n*) and ending (*m*) numbers in the range must include the same number of digits. For example, (23-34) is correct, but (3-12) is not.
[n,m,...] or **(n,m,...)**	Represents multiple numbers. The value can include digits or characters. Examples: ▪ To depict a one-digit number starting with 2, 3, 4, 5, or 6: **[2,3,4,5,6]** ▪ To depict a one-digit number ending with 7, 8, or 9: **(7,8,9)** ▪ Prefix with Suffix: **[2,3,4,5,6](7,8,9) -** prefix is denoted in square brackets; suffix in parenthesis For **prefix only**, the notations *d[n,m]e* and *d[n-m]e* can also be used: ▪ To depict a five-digit number that starts with 11, 22, or 33: **[11,22,33]xxx#** ▪ To depict a six-digit number that starts with 111 or 222: **[111,222]xxx#**
[n1-m1,n2-m2,a,b,c,n3-m3] or **(n1-m1,n2-m2,a,b,c,n3-m3)**	Represents a mixed notation of single numbers and multiple ranges. For example, to depict numbers 123 to 130, 455, 766, and 780 to 790: ▪ Prefix: **[123-130,455,766,780-790]** ▪ Suffix: **(123-130,455,766,780-790)** **Note:** The ranges and the single numbers used in the dial plan must have the same number of digits. For example, each number range and single number in the dialing plan example above consists of three digits.
Special ASCII Characters	The device does not support the use of ASCII characters in manipulation rules and therefore, for LDAP-based queries, the device can use the hexadecimal (HEX) format of the ASCII characters for phone numbers instead. The HEX value must be preceded by a backslash "\". For example, you can configure a manipulation rule that changes the number +49 (7303) 165-xxxxx to +49 \287303\29 165-xxxxx, where *\28* is the ASCII HEX value for "(" and *\29* is the ASCII HEX value for ")". The manipulation rule in this example would denote the parenthesis in the destination number prefix using "x" wildcards (e.g., xx165xxxxx#); the prefix to add to the number would include the HEX values (e.g., +49 \287303\29 165-). Below is a list of common ASCII characters and their corresponding HEX values: **ASCII Character** **HEX Value** * \2a (\28) \29 \ \5c / \2f

Note: When configuring phone numbers or prefixes in the Web interface, enter them only as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

67 Configuration Parameters Reference

The device's VoIP functionality (not data-routing functionality) 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.

67.1 Management Parameters

This section describes the device's management-related parameters.

67.1.1 General Parameters

The general management parameters are described in the table below.

Table 67-1: General Management Parameters

Parameter	Description
WAN OAMP Interface CLI: bind GigabitEthernet <slot/port.vlanId> oamp [OAMPWanInterfaceName]	Binds the OAMP interface to a WAN interface, which can later be associated with a Virtual Routing and Forwarding (VRF).
Web: Allow WAN access to HTTP CLI: wan-http-allow [AllowWanHTTP]	Enables WAN access to the management interface through HTTP. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to HTTPS CLI: wan-https-allow [AllowWanHTTPS]	Enables WAN access to the management interface through HTTPS. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to SNMP CLI: wan-snmp-allow [AllowWanSNMP]	Enables WAN access to the management interface through SNMP. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to Telnet CLI: wan-telnet-allow [AllowWanTelnet	Enables WAN access to the management interface through Telnet. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Allow WAN access to SSH CLI: wan-ssh-allow [AllowWanSSH]	Enables WAN access to the management interface through SSH. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: Web and Telnet Access List Table EMS: Web Access Addresses **[WebAccessList_x]**	This table configures up to ten IP addresses that are permitted to access the device's Web interface and Telnet interfaces. Access from an undefined IP address is denied. When no IP addresses are defined in this table, this security feature is inactive (i.e., the device can be accessed from any IP address). The default is 0.0.0.0 (i.e., the device can be accessed from any IP address). For example: WebAccessList_0 = 10.13.2.66 WebAccessList_1 = 10.13.77.7 For a description of this parameter, see "Configuring Web and Telnet Access List" on page 73.

67.1.2 Web Parameters

The Web parameters are described in the table below.

Table 67-2: Web Parameters

Parameter	Description
Web: Password Change Interval **[WebUserPassChangeInterval]**	Defines the duration (in minutes) of the validity of Web login passwords. When this duration expires, the password of the Web user must be changed. The valid value is 0 to 100000, where 0 means that the password is always valid. The default is 1140. **Note:** This parameter is applicable only when using the Web Users table, where the default value of the 'Password Age' parameter in the Web Users table inherits this parameter's value.
Web: User Inactivity Timer **[UserInactivityTimer]**	Defines the duration (in days) for which a user has not logged in to the Web interface, after which the status of the user becomes inactive and can no longer access the Web interface. These users can only log in to the Web interface if their status is changed (to New or Valid) by a System Administrator or Master user. The valid value is 0 to 10000, where 0 means inactive. The default is 90. **Note:** This parameter is applicable only when using the Web Users table.
Web: Session Timeout **[WebSessionTimeout]**	Defines the duration (in minutes) of Web inactivity of a logged-in user, after which the user is automatically logged off the Web interface. The valid value is 0-100000, where 0 means no timeout. The default is 15. **Note:** This parameter can apply to all users, or per user when set in the Web Users table.
Web: Deny Access On Fail Count **[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.

Parameter	Description
Web: Deny Authentication Timer EMS: WEB Deny Authentication Timer **[DenyAuthenticationTimer]**	Defines the duration (in seconds) for which login to the Web interface is denied from a specific IP address (for all users) when the number of failed login attempts has exceeded the maximum. This maximum is defined by the DenyAccessOnFailCount parameter. Only after this time expires can users attempt to login from this same IP address. The valid value is 0 to 100000, where 0 means that login is not denied regardless of number of failed login attempts. The default is 60.
Web: Display Login Information **[DisplayLoginInformation]**	Enables display of user's login information on each successful login attempt. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
[EnableMgmtTwoFactorAuthentication]	Enables Web login authentication using a third-party, smart card. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable When enabled, the device retrieves the Web user's login username from the smart card, which is automatically displayed (read-only) in the Web Login screen; the user is then required to provide only the login password. Typically, a TLS connection is established between the smart card and the device's Web interface, and a RADIUS server is implemented to authenticate the password with the username. Thus, this feature implements a two-factor authentication - what the user has (the physical card) and what the user knows (i.e., the login password).
EMS: HTTPS Port CLI: http-port **[HTTPport]**	Defines the LAN HTTP port for Web management (default is 80). To enable Web management from the LAN, configure the desired port. **Note:** For this parameter to take effect, a device reset is required.
EMS: Disable WEB Config **[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'). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪

Parameter	Description
[ResetWebPassword]	Resets the username and password of the primary ("Admin") and secondary ("User") accounts to their default settings ("Admin" and "Admin" respectively), and deletes all other users that may have been configured. ▪ **[0]** = (Default) Password and username retain their values. ▪ **[1]** = Password and username are reset. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ You cannot reset the username and password through the Web interface (by loading an ini file or on the AdminPage). To reset the username and password: ✓ SNMP: 1) Set acSysGenericINILine to WEBPasswordControlViaSNMP = 1, and reset the device with a flash burn (set acSysActionSetResetControl to 1 and acSysActionSetReset to 1). 2) Change the username and password in the acSysWEBAccessEntry table. Use the following format: Username acSysWEBAccessUserName: old/pass/new Password acSysWEBAccessUserCode: username/old/new
[WelcomeMessage]	Enables and defines a Welcome message that appears on the Web Login page for logging in to the Web interface. The format of this parameter is as follows: [WelcomeMessage] FORMAT WelcomeMessage_Index = WelcomeMessage_Text [\WelcomeMessage] For Example: FORMAT WelcomeMessage_Index = WelcomeMessage_Text WelcomeMessage 1 = "**********************************" ; WelcomeMessage 2 = "********* This is a Welcome message ***" ; WelcomeMessage 3 = "**********************************" ; **Notes:** ▪ Each index row represents a line of text in the Welcome message box. Up to 20 lines (or rows) of text can be defined. ▪ The configured text message must be enclosed in double quotation marks (i.e., "..."). ▪ If this parameter is not configured, no Welcome message is displayed.

67.1.3 Telnet Parameters

The Telnet parameters are described in the table below.

Table 67-3: Telnet Parameters

Parameter	Description
Web: Embedded Telnet Server EMS: Server Enable CLI: telnet **[TelnetServerEnable]**	Enables the device's embedded Telnet server. Telnet is disabled by default for security. ▪ **[0]** Disable ▪ **[1]** Enable Unsecured (default) ▪ **[2]** Enable Secured **Note:** Only the primary Web User Account (which has Security Administration access level) can access the device using Telnet (see "Configuring Web User Accounts" on page 64).
Web: Telnet Server TCP Port EMS: Server Port CLI: telnet-port **[TelnetServerPort]**	Defines the port number for the embedded Telnet server. The valid range is all valid port numbers. The default port is 23.
Web: Telnet Server Idle Timeout EMS: Server Idle Disconnect CLI: idle-timeout **[TelnetServerIdleDisconnect]**	Defines the timeout (in minutes) for disconnection of an idle Telnet session. When set to zero, idle sessions are not disconnected. The valid range is any value. The default is 0. **Note:** For this parameter to take effect, a device reset is required.
Web: Maximum Telnet Sessions CLI: telnet-max-sessions **[TelnetMaxSessions]**	Defines the maximum number of permitted, concurrent Telnet/SSH sessions. The valid range is 1 to 5 sessions. The default is 2. **Note:** Before changing the value, make sure that not more than this number of sessions are currently active; otherwise, the new setting 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". **Notes:** ▪ The password is case-sensitive. ▪ The password is required only by Admin and Monitor user access levels; Security Administrator and Master user access levels automatically enter the Enable mode upon initial login.

67.1.4 ini File Parameters

The parameters relating to ini-file management are described in the table below.

Table 67-4: 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

Parameter	Description
	password> (e.g., $1$S3p+fno=). - **[1]** Enable = All passwords are hidden and replaced by an asterisk (*).

67.1.5 SNMP Parameters

The SNMP parameters are described in the table below.

Table 67-5: SNMP Parameters

Parameter	Description
Web: Enable SNMP CLI: disable **[DisableSNMP]**	Enables SNMP. - **[0]** Enable (default) - **[1]** Disable = SNMP is disabled and no traps are sent.
CLI: port **[SNMPPort]**	Defines the device's local (LAN) UDP port used for SNMP Get/Set commands. The range is 100 to 3999. The default port is 161. **Note:** For this parameter to take effect, a device reset is required.
[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.
EMS: Keep Alive Trap Port **[KeepAliveTrapPort]**	Defines the port to which keep-alive traps are sent. The valid range is 0 - 65534. The default is port 162.
[SendKeepAliveTrap]	Enables keep-alive traps and sends them every 9/10 of the time as defined by the NATBindingDefaultTimeout parameter. - **[0]** = Disable - **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
CLI: sys-oid **[SNMPSysOid]**	Defines the base product system OID. The default is eSNMP_AC_PRODUCT_BASE_OID_D. **Note:** For this parameter to take effect, a device reset is required.
[SNMPTrapEnterpriseOid]	Defines the Trap Enterprise OID. The default is eSNMP_AC_ENTERPRISE_OID. The inner shift of the trap in the AcTrap subtree is added to the end of the OID in this parameter. **Note:** For this parameter to take effect, a device reset is

Parameter	Description
	required.
[acUserInputAlarmDescription]	Defines the description of the input alarm.
[acUserInputAlarmSeverity]	Defines the severity of the input alarm.
[AlarmHistoryTableMaxSize]	Defines the maximum number of rows in the Alarm History table. This parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB). The valid range is 50 to 1000. The default is 500. **Note:** For this parameter to take effect, a device reset is required.
CLI: engine-id **[SNMPEngineIDString]**	Defines the SNMP engine ID for SNMPv2/SNMPv3 agents. This is used for authenticating a user attempting to access the SNMP agent on the device. The ID can be a string of up to 36 characters. The default is 00:00:00:00:00:00:00:00:00:00:00:00 (12 Hex octets characters). The provided key must be set with 12 Hex values delimited by a colon (":") in the format xx:xx:...:xx. For example, 00:11:22:33:44:55:66:77:88:99:aa:bb **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Before setting this parameter, all SNMPv3 users must be deleted; otherwise, the parameter setting is ignored. ▪ If the supplied key does not pass validation of the 12 Hex values input or it is set with the default value, the engine ID is generated according to RFC 3411.
Web: SNMP Trap Destination Parameters EMS: Network > SNMP Managers Table CLI: configure system/snmp trap destination **Note:** Up to five SNMP trap managers can be defined.	
SNMP Manager **[SNMPManagerIsUsed_x]**	Determines the validity of the parameters (IP address and port number) of the corresponding SNMP Manager used to receive SNMP traps. ▪ **[0]** (Check box cleared) = Disabled (default) ▪ **[1]** (Check box selected) = Enabled
Web: IP Address EMS: Address CLI: ip-address **[SNMPManagerTableIP_x]**	Defines the IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to this IP address. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255.
Web: Trap Port EMS: Port CLI: port **[SNMPManagerTrapPort_x]**	Defines the port number of the remote SNMP Manager. The device sends SNMP traps to this port. The valid SNMP trap port range is 100 to 4000. The default port is 162.
Web: Trap Enable CLI: send-trap **[SNMPManagerTrapSendingEnable_x]**	Enables the sending of traps to the corresponding SNMP manager. ▪ **[0]** Disable = Sending is disabled. ▪ **[1]** Enable = (Default) Sending is enabled.

Parameter	Description
Web: Trap User CLI: trap-user **[SNMPManagerTrapUser_x]**	Defines the SNMPv3 USM user or SNMPv2 user to associate with the trap destination. This determines the trap format, authentication level, and encryption level. By default, it is associated with the SNMPv2 user (SNMP trap community string). The valid value is a string.
Web: Trap Manager Host Name CLI: manager-host-name **[SNMPTrapManagerHostName]**	Defines an FQDN of the remote host used as an SNMP manager. The resolved IP address replaces the last entry in the Trap Manager table (defined by the SNMPManagerTableIP parameter) and the last trap manager entry of snmpTargetAddrTable in the snmpTargetMIB. For example: 'mngr.corp.mycompany.com'. The valid range is a string of up to 99 characters.
SNMP Community String Parameters	
Community String CLI: ro-community-string **[SNMPReadOnlyCommunityString_x]**	Defines up to five read-only SNMP community strings (up to 19 characters each). The default string is 'public'.
Community String CLI: rw-community-string **[SNMPReadWriteCommunityString_x]**	Defines up to five read/write SNMP community strings (up to 19 characters each). The default string is 'private'.
Trap Community String CLI: community-string **[SNMPTrapCommunityString]**	Defines the Community string used in traps (up to 19 characters). The default string is 'trapuser'.
SNMP Trusted Managers Table	
Web: SNMP Trusted Managers CLI: configure system > snmp > trusted-managers **[SNMPTrustedMgr_x]**	Defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. **Notes:** ▪ By default, the SNMP agent accepts SNMP Get and Set requests from any IP address, as long as the correct community string is used in the request. Security can be enhanced by using Trusted Managers, which is an IP address from which the SNMP agent accepts and processes SNMP requests. ▪ If no values are assigned to these parameters any manager can access the device. ▪ Trusted managers can work with all community strings.
SNMP V3 Users Table	
Web/EMS: SNMP V3 Users CLI: configure system > snmp v3-users **[SNMPUsers]**	This *parameter* table defines SNMP v3 users. The format of this parameter is as follows: [SNMPUsers] FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group; [\SNMPUsers] For example: SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1; The example above configures user 'v3admin1' with

Parameter	Description
	security level authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2. For a description of this table, see "Configuring SNMP V3 Users" on page 91.

67.1.6 TR-069 Parameters

The TR-069 parameters are described in the table below.

Table 67-6: TR-069 Parameters

Parameter	Description
Web: TR069 CLI: service **[TR069ServiceEnable]**	Enables device management using TR-069. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Interface Name CLI: interface-name **[TR069NetworkSource]**	Defines the device's network interface used for the TR-069 connection. ▪ **[0]** LAN ▪ **[1]** WAN Ethernet (default)
Web: Protocol CLI: protocol **[TR069Protocol]**	Defines the protocol used for the TR-069 connection. ▪ **[0]** HTTP (default) ▪ **[1]** HTTPS
Web: Port CLI: port **[TR069HTTPPort]**	Defines the local HTTP/S port used for TR-069. The valid range is 0 to 65535. The default is 82. **Note:** For this parameter to take effect, a device reset is required.
Web: URL Provisioning Mode CLI: acs-url-provisioning-mode **[Tr069AcsUrlProvisioningMode]**	Defines the method for configuring the URL of the TR-069 ACS. ▪ **[0]** Manual (default) = URL must be configured manually on the device. The URL is configured using the TR069ConnectionRequestUrl parameter. ▪ **[1]** Automatic = Device uses DHCP Option 43 to obtain URL address of ACS.
Web: URL CLI: acl-url **[TR069AcsUrl]**	Defines the URL address of the Auto Configuration Servers (ACS) to which the device connects. For example, http://10.4.2.1:10301/acs/. By default, no URL is defined. **Note:** This parameter is applicable only if the 'URL Provisioning Mode' parameter is set to **Manual**.
Web: Username CLI: acs-user-name **[TR069AcsUsername]**	Defines the login username that the device uses for authenticated access to the ACS. The valid value is a string of up to 256 characters. By default, no username is defined.

Parameter	Description
Web: Password CLI: acs-password **[TR069AcsPassword]**	Defines the login password that the device uses for authenticated access to the ACS. The valid value is a string of up to 256 characters. By default, no password is defined.
Web: URL CLI: connection-request-url **[TR069ConnectionRequestUrl**	Defines the URL for the ACS connection request. For example, http://10.31.4.115:82/tr069/.
Web: Username CLI: connection-request-user-name **[TR069ConnectionRequestUsername]**	Defines the connection request username used by the ACS to connect to the device. The valid value is a string of up to 256 characters. By default, no username is defined.
Web: Password CLI: connection-request-password **[TR069ConnectionRequestPassword]**	Defines the connection request password used by the ACS to connect to the device. The valid value is a string of up to 256 characters. By default, no password is defined.
Web: Default Inform Interval CLI: inform-interval **[TR069PeriodicInformInterval]**	Defines the inform interval (in seconds) at which the device periodically communicates with the ACS. Each time the device communicates with the ACS, the ACS sends a response indicating whether or not the ACS has an action to execute on the device. The valid value is 0 to 4294967295. The default is 60.
[TR069RetryinimumWaitInterval]	Defines the minimum interval (in seconds) that the device waits before attempting again to communicate with the ACS after the previous communication attempt failure. The valid value is 1 to 65535. The default is 5.
CLI: debug-mode **[TR069DebugMode]**	Defines the debug mode level, which is the type of messages sent to the Syslog server. The valid value is between 0 and 3, where 0 (default) means no debug messages are sent and 3 is all message types are sent.

67.1.7 Serial Parameters

The RS-232 serial parameters are described in the table below.

Table 67-7: 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 this parameter to take effect, a device reset is required.
EMS: Baud Rate **[SerialBaudRate]**	Defines the RS-232 baud rate. The valid values include the following: 1200, 2400, 9600, 14400, 19200, 38400, 57600, or 115200 (default).

Parameter	Description
	Note: For this parameter to take effect, a device reset is required.
EMS: Data **[SerialData]**	Defines the RS-232 data bit. ▪ **[7]** = 7-bit ▪ **[8]** = (Default) 8-bit **Note:** For this parameter to take effect, a device reset is required.
EMS: Parity **[SerialParity]**	Defines the RS-232 polarity. ▪ **[0]** = (Default) None ▪ **[1]** = Odd ▪ **[2]** = Even **Note:** For this parameter to take effect, a device reset is required.
EMS: Stop **[SerialStop]**	Defines the RS-232 stop bit. ▪ **[1]** = (Default) 1-bit (default) ▪ **[2]** = 2-bit **Note:** For this parameter to take effect, a device reset is required.
EMS: Flow Control **[SerialFlowControl]**	Defines the RS-232 flow control. ▪ **[0]** = (Default) None ▪ **[1]** = Hardware **Note:** For this parameter to take effect, a device reset is required.

67.1.8 Auxiliary and Configuration File Name Parameters

The configuration files (i.e., auxiliary files) can be loaded to the device using the Web interface. For loading these files using the *ini* file, you need to configure these files in the *ini* file and configured whether they must be stored in the non-volatile memory. The table below lists the *ini* file parameters associated with these auxiliary files. For more information on the auxiliary files, see "Loading Auxiliary Files" on page 599.

Table 67-8: Auxiliary and Configuration File Parameters

Parameter	Description
General Parameters	
[SetDefaultOnIniFileProcess]	Determines if all the device's parameters are set to their defaults before processing the updated *ini* file. ▪ **[0]** = Disable - parameters not included in the downloaded *ini* file are not returned to default settings (i.e., retain their current settings). ▪ **[1]** = Enable (default). **Note:** This parameter is applicable only for automatic HTTP update or Web *ini* file upload (not applicable if the *ini* file is loaded using BootP).
[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	

Parameter	Description
Web/EMS: Call Progress Tones File **[CallProgressTonesFilename]**	Defines the name of the file containing the Call Progress Tones definitions. For more information on how to create and load this file, refer to *DConvert Utility User's Guide*. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Prerecorded Tones File **[PrerecordedTonesFileName]**	Defines the name (and path) of the file containing the Prerecorded Tones. **Note:** For this parameter to take effect, a device reset is required.
Web: CAS File EMS: Trunk Cas Table Index [CASFileName_x]	Defines the CAS file name (e.g., 'E_M_WinkTable.dat'), which defines the CAS protocol (where x denotes the CAS file ID 0 to 7). It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device's trunks, using the parameter CASTableIndex or it can be associated per B-channel using the parameter CASChannelIndex. **Note:** For this parameter to take effect, a device reset is required.
Web: Dial Plan EMS: Dial Plan Name [CasTrunkDialPlanName_x]	Defines the Dial Plan name (up to 11-character strings) per trunk. **Note:** The x in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
Web: Dial Plan File EMS: Dial Plan File Name **[DialPlanFileName]**	Defines the name (and path) of the Dial Plan file. This file should be created using AudioCodes DConvert utility (refer to *DConvert Utility User's Guide*).
[UserInfoFileName]	Defines the name (and path) of the file containing the User Information data.

67.1.9 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

Table 67-9: Automatic Update of Software and Configuration Files Parameters

Parameter	Description
General Automatic Update Parameters	
[AutoUpdateCmpFile]	Enables the Automatic Update mechanism for the cmp file. ▪ **[0]** = (Default) The Automatic Update mechanism doesn't apply to the cmp file. ▪ **[1]** = The Automatic Update mechanism includes the cmp file. **Note:** For this parameter to take effect, a device reset is required.
CLI: 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 this parameter to take effect, a device reset is required.
CLI: configure system > automatic-update > predefined-time **[AutoUpdatePredefinedTime]**	Defines schedules (time of day) for performing automatic updates. The format of this parameter is "hh:mm", where *hh* denotes the hour and *mm* the minutes, for example, 20:18. **Notes:**

Parameter	Description
	- For this parameter to take effect, a device reset is required. - The actual update time is randomized by five minutes to reduce the load on the Web servers.
CLI: automatic-update > http-user-agent **[AupdHttpUserAgent]**	Defines the User-Agent HTTP header in the HTTP Get requests sent by the device to the provisioning server for the Automatic Update mechanism. The valid value is a string of up to 511 characters. By default, this parameter is not defined. In other words, the User-Agent header is set to "Mozilla/4.0 (compatible; AudioCodes; <NAME>;<VER>)", where the tags are replaced with actual values. **Notes:** - The variable tags are case-sensitive. - The tags can be defined in any order. - The tags must be defined adjacent to one another (i.e., no spaces or special characters).
CLI: automatic-update > auto-firmware **[AutoCmpFileUrl]**	Defines the filename and path (URL) to the provisioning server from where the software file (.cmp) can be downloaded, based on timestamp for the Automatic Updated mechanism. The valid value is an IP address in dotted-decimal notation or an FQDN.
[AUPDDigestUsername]	Defines the username for digest (MD5 cryptographic hashing) access authentication with the HTTP server used for the Automatic Update feature. The valid value is a string of up to 50 characters. By default, no value is defined.
[AUPDDigestPassword]	Defines the password for digest (MD5 cryptographic hashing) access authentication with the HTTP server used for the Automatic Update feature. The valid value is a string of up to 50 characters. By default, no value is defined.
EMS: AUPD Verify Certificates CLI: system > tls > aupd-verify-cert **[AUPDVerifyCertificates]**	Determines whether the Automatic Update mechanism verifies server certificates when using HTTPS. - **[0]** = Disable (default) - **[1]** = Enable
CLI: configure system > automatic-update > crc-check regular **[AUPDCheckIfIniChanged]**	Determines whether the Automatic Update mechanism performs CRC to determine if the configuration file (ini, Startup Script, or CLI Script files) has changed prior to processing. - **[0]** = (Default) Do not check CRC. The file is loaded whenever the server provides it. - **[1]** = Check CRC for the entire file. Any change, including line order, causes the file to be re-processed. - **[2]** = Check CRC for individual lines. Use this option when the HTTP server scrambles the order of lines in the provided file.

Parameter	Description
CLI: automatic-update > use-zero-conf-certs [AupdUseZeroConfCerts]	Enables the Automatic Update mechanism to use the same client-server certificate as used for the Zero Configuration feature, instead of the "regular" certificate used for Automatic Update. ▪ [0] = (Default) The device uses the "regular" certificate for Automatic Update. ▪ [1] = The device uses the Zero Configuration certificate for the Automatic Update feature.
CLI: config-system > automatic-update tftp-block-size **[AUPDTftpBlockSize]**	Defines the size of the TFTP data blocks (packets) when downloading a file from a TFTP server for the Automatic Update mechanism. This is in accordance to RFC 2348. TFTP block size is the physical packet size (in bytes) that a network can transmit. When configured to a value higher than the default (512 bytes), but lower than the client network's Maximum Transmission Unit (MTU), the file download speed can be significantly increased. The valid value is 512 to 8192. The default is 512. **Notes:** ▪ A higher value does not necessarily mean better performance. ▪ The block size should be small enough to avoid IP fragmentation in the client network (i.e., below MTU). ▪ This feature is applicable only to TFTP servers that support this option.
[ResetNow]	Invokes an immediate device reset. This option can be used to activate offline (i.e., not on-the-fly) parameters that are loaded using the parameter IniFileUrl. ▪ **[0]** = (Default) The immediate restart mechanism is disabled. ▪ **[1]** = The device immediately resets after an *ini* file with this parameter set to 1 is loaded. **Note:** If you use this parameter in an ini file for periodic Automatic Update feature with non-HTTP (e.g., TFTP) and without CRC, the device resets after every file download.
Software/Configuration File URL Path for Automatic Update Parameters	
CLI: automatic-update > firmware **[CmpFileURL]**	Defines the name of the *cmp* file and the path to the server (IP address or FQDN) from where the device can load the *cmp* file and update itself. The *cmp* file can be loaded using HTTP/HTTPS. For example, http://192.168.0.1/filename. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When this parameter is configured, the device always loads the *cmp* file after it is reset. ▪ The *cmp* file is validated before it's burned to flash. The checksum of the *cmp* file is also compared to the previously burnt checksum to avoid unnecessary resets. ▪ The maximum length of the URL address is 255 characters.

Parameter	Description
CLI: voice-configuration **[IniFileURL]**	Defines the name of the *ini* file and the path to the server (IP address or FQDN) on which it is located. The *ini* file can be loaded using HTTP/HTTPS. For example: http://192.168.0.1/filename http://192.8.77.13/config<MAC> https://<username>:<password>@<IP address>/<file name> **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When using HTTP or HTTPS, the date and time of the *ini* file are validated. Only more recently dated *ini* files are loaded. ▪ The optional string <MAC> is replaced with the device's MAC address. Therefore, the device requests an *ini* file name that contains its MAC address. This option allows the loading of specific configurations for specific devices. ▪ The maximum length of the URL address is 99 characters.
CLI: cli-script <URL> [AUPDCliScriptURL]	Defines the URL of the server where the CLI Script file containing the device's configuration is located. This file is used for automatic provisioning.
CLI: startup-script <URL> [AUPDStartupScriptURL]	Defines the URL of the server where the CLI Startup Script file containing the device's configuration is located. This file is used for automatic provisioning.
CLI: prerecorded-tones **[PrtFileURL]**	Defines the name of the Prerecorded Tones (PRT) file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
CLI: call-progress-tones **[CptFileURL]**	Defines the name of the CPT file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
CLI: cas-table [CasFileURL]	Defines the name of the CAS file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file. **Note:** The maximum length of the URL address is 99 characters.
CLI: tls-root-cert **[TLSRootFileUrl]**	Defines the name of the TLS trusted root certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
CLI: tls-cert **[TLSCertFileUrl]**	Defines the name of the TLS certificate file and the URL from where it can be downloaded. **Note:** For this parameter to take effect, a device reset is required.
CLI: tls-private-key **[TLSPkeyFileUrl]**	Defines the URL for downloading a TLS private key file using the Automatic Update facility.
[UserInfoFileURL]	Defines the name of the User Information file and the path to the server (IP address or FQDN) on which it is located. For example: http://server_name/file, https://server_name/file **Note:** The maximum length of the URL address is 99 characters.

67.2 Networking Parameters

This subsection describes the device's networking parameters.

67.2.1 Multiple VoIP Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

Table 67-10: IP Network Interfaces and VLAN Parameters

Parameter	Description
Interface Table	
Web: Interface Table EMS: IP Interface Settings CLI: configure voip > interface network-if display **[InterfaceTable]**	This table parameter configures the Interface table. The format of the ini file table parameter is as follows: [InterfaceTable] FORMAT InterfaceTable_Index = InterfaceTable_ApplicationTypes, InterfaceTable_InterfaceMode, InterfaceTable_IPAddress, InterfaceTable_PrefixLength, InterfaceTable_Gateway, InterfaceTable_VlanID, InterfaceTable_InterfaceName, InterfaceTable_PrimaryDNSServerIPAddress, InterfaceTable_SecondaryDNSServerIPAddress, InterfaceTable_UnderlyingDevice; [\InterfaceTable] For a detailed description of this table, see "Configuring IP Network Interfaces" on page 130. **Note:** For this parameter to take effect, a device reset is required.
VLAN Parameters	
[EnableNTPasOAM]	Defines the application type for Network Time Protocol (NTP) services. ▪ **[1]** = OAMP (default) ▪ **[0]** = Control **Note:** For this parameter to take effect, a device reset is required.

67.2.2 Routing Parameters

The IP network routing parameters are described in the table below.

Table 67-11: IP Network Routing Parameters

Parameter	Description
Static Route Table	
Web/EMS**:** Static Route Table CLI: configure voip > static **[StaticRouteTable]**	Defines up to 30 static VoIP IP routes for the device. The format of the ini file table parameter is as follows: [StaticRouteTable] FORMAT StaticRouteTable_Index = StaticRouteTable_DeviceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [\StaticRouteTable] For a description of this parameter, see "Configuring Static IP Routes" on page 139.

67.2.3 Open Solution Network (OSN) Parameters

The OSN server parameters are described in the table below.

Table 67-12: OSN Parameters

Parameter	Description
Web: OSN Native VLAN ID CLI: configure system > interface osn > native-vlan <id> **[OSNAccessVlan]**	Defines the OSN Native VLAN ID. The valid value is 0 to 4000. When set to 0 (default), the OSN uses the device's OAMP VLAN ID. When set to any other value, it specifies a VLAN ID configured in the Ethernet Device table and which is assigned to a Media and/or Control application in the Interface table.
Web: Block OSN Port CLI: configure system > interface osn > shutdown **[OSNBlockPort]**	Enables or disables the Ethernet port of the internal switch that interfaces with the OSN. ▪ **[0]** Enable (default) ▪ **[1]** Disable

67.2.4 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

Table 67-13: QoS Parameters

Parameter	Description
Layer-2 Class Of Service (CoS) Parameters (VLAN Tag Priority Field)	
Web: DiffServ Table EMS: QoS Settings – DSCP to QoS Mapping CLI: configure voip > vlan-mapping [DiffServToVlanPriority]	This table parameter configures DiffServ-to-VLAN Priority mapping. For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet. The format of this ini file is as follows: [DiffServToVlanPriority] FORMAT DiffServToVlanPriority_Index = DiffServToVlanPriority_DiffServ, DiffServToVlanPriority_VlanPriority; [\DiffServToVlanPriority] For example: DiffServToVlanPriority 0 = 46, 6; DiffServToVlanPriority 1 = 40, 6; DiffServToVlanPriority 2 = 26, 4; DiffServToVlanPriority 3 = 10, 2; For a description of this table, see Configuring Quality of Service on page 142. **Note:** For this parameter to take effect, a device reset is required.
Layer-3 Class of Service (TOS/DiffServ) Parameters	
Web: Media Premium QoS EMS: Premium Service Class Media Diff Serv CLI: media-qos	Global parameter that defines the DiffServ value for Premium Media CoS content. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IPDiffServ). For a detailed description of this

Parameter	Description
[PremiumServiceClassMediaDiffServ]	parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Control Premium QoS EMS: Premium Service Class Control Diff Serv CLI: control-qos **[PremiumServiceClassControlDiffServ]**	Global parameter that defines the DiffServ value for Premium Control CoS content (Call Control applications). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SigIPDiffServ). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Gold QoS EMS: Gold Service Class Diff Serv CLI: gold-qos **[GoldServiceClassDiffServ]**	Defines the DiffServ value for the Gold CoS content (Streaming applications). The valid range is 0 to 63. The default is 26.
Web: Bronze QoS EMS: Bronze Service Class Diff Serv CLI: bronze-qos **[BronzeServiceClassDiffServ]**	Defines the DiffServ value for the Bronze CoS content (OAMP applications). The valid range is 0 to 63. The default is 10.

67.2.5 NAT and STUN Parameters

The Network Address Translation (NAT) parameters are described in the table below.

Table 67-14: NAT Parameters

Parameter	Description
NAT Parameters	
Web/EMS: NAT Mode CLI: disable-NAT-traversal **[NATMode]**	Enables the NAT feature for media when the device communicates with UAs located behind NAT. ▪ **[0]** Auto-Detect = NAT is performed only if necessary. If the UA is identified as being located behind NAT, the device sends the media packets to the public IP address:port obtained from the source address of the first media packet received from the UA. Otherwise, the packets are sent using the IP address:port obtained from the address in the first received SIP message. Note also that if the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA, does it determine whether the UA is behind NAT. ▪ **[1]** NAT Is Not Used = (Default) NAT feature is disabled. The device always sends the media packets to the remote UA using the IP address:port obtained from the first received SIP message. ▪ **[2]** NAT Is Used = NAT is always performed. The device always sends the media packets to the remote UA using the source address obtained from the first media packet from the UA. In this mode, the device does not send any packets until it receives the first packet from the UA (in order to obtain the IP address).

Parameter	Description
	For more information on handling calls from UAs behind NAT, see "First Incoming Packet Mechanism" on page 154.
Web: NAT IP Address EMS: Static NAT IP Address CLI: nat-ip-addr **[StaticNatIP]**	Defines the global (public) IP address of the device to enable static NAT between the device and the Internet. **Note:** For this parameter to take effect, a device reset is required.
Web: SIP NAT Detection CLI: configure voip/sip-definition advanced-settings/sip-nat-detect **[SIPNatDetection]**	Enables the device to detect whether the incoming INVITE message is sent from an endpoint located behind NAT. ▪ **[0]** Disable = Disables the device's NAT Detection mechanism. Incoming SIP messages are processed as received from endpoints that are not located behind NAT and sent according to the SIP standard. ▪ **[1]** Enable (default) = Enables the device's NAT Detection mechanism.

67.2.6 NFS Parameters

The Network File Systems (NFS) configuration parameters are described in the table below.

Table 67-15: NFS Parameters

Parameter	Description
CLI: base-port **[NFSBasePort]**	Defines the start of the range of numbers used for local UDP ports used by the NFS client. The maximum number of local ports is maximum channels plus maximum NFS servers. The valid range is 0 to 65535. The default is 47000.
NFS Table	
Web: NFS Table EMS: NFS Settings CLI: configure system/nfs/servers **[NFSServers]**	This table parameter defines up to 16 NFS file systems so that the device can access a remote server's shared files and directories for loading cmp, ini, and auxiliary files (using the Automatic Update mechanism). The format of this table ini file parameter is as follows: [NFSServers] FORMAT NFSServers_Index = NFSServers_HostOrIP, NFSServers_RootPath, NFSServers_NfsVersion, NFSServers_AuthType, NFSServers_UID, NFSServers_GID, NFSServers_VlanType; [\NFSServers] For example: NFSServers 1 = 101.1.13, /audio1, 3, 1, 0, 1, 1; For a detailed description of this table, see "Configuring NFS Settings" on page 148.

67.2.7 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table 67-16: DNS Parameters

Parameter	Description
Internal DNS Table	
Web: Internal DNS Table EMS: DNS Information CLI: configure voip > voip-network dns Dns2Ip **[DNS2IP]**	This table parameter defines the internal DNS table for resolving host names into IP addresses. Up to four different IP addresses (in dotted-decimal notation) can be assigned to a host name. The format of this parameter is as follows: [Dns2Ip] FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress, Dns2Ip_FourthIpAddress; [\Dns2Ip] For example: Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, 4.4.4.4; For a detailed description of this table, see "Configuring the Internal DNS Table" on page 144.
Internal SRV Table	
Web: Internal SRV Table EMS: DNS Information CLI: configure voip > voip-network dns Srv2Ip **[SRV2IP]**	This table parameter defines the internal SRV table for resolving host names into DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of this parameter is as follows: [SRV2IP] FORMAT SRV2IP_Index = SRV2IP_InternalDomain, SRV2IP_TransportType, SRV2IP_Dns1, SRV2IP_Priority1, SRV2IP_Weight1, SRV2IP_Port1, SRV2IP_Dns2, SRV2IP_Priority2, SRV2IP_Weight2, SRV2IP_Port2, SRV2IP_Dns3, SRV2IP_Priority3, SRV2IP_Weight3, SRV2IP_Port3; [\SRV2IP] For example: SRV2IP 0 = SrvDomain,0,Dnsname1,1,1,500,Dnsname2,2,2,501,$$,0,0,0; For a detailed description of this table, see "Configuring the Internal SRV Table" on page 145.

67.2.8 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

Table 67-17: DHCP Parameters

Parameter	Description
Web: Enable DHCP EMS: DHCP Enable **[DHCPEnable]**	Enables DHCP functionality. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required.

Parameter	Description
	- For a detailed description of DHCP, see "DHCP-based Provisioning" on page 631. - This parameter is a "hidden" parameter. Once defined and saved to flash memory, its value doesn't revert to default even if the parameter doesn't appear in the *ini* file.
EMS: DHCP Speed Factor **[DHCPSpeedFactor]**	Defines the DHCP renewal speed. - **[0]** = Disable - **[1]** = (Default) Normal - **[2]** to **[10]** = Fast When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is divided by this factor. Some DHCP-enabled routers perform better when set to 4. **Note:** For this parameter to take effect, a device reset is required.

67.2.9 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

Table 67-18: 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 121.	
Web: NTP Server Address EMS: Server IP Address CLI: primary-server **[NTPServerIP]**	Defines the IP address (in dotted-decimal notation or as an FQDN) of the NTP server. The advantage of using an FQDN is that multiple IP addresses can be resolved from the DNS server, providing NTP server redundancy. The default IP address is 0.0.0.0 (i.e., internal NTP client is disabled).
Web: NTP Secondary Server Address **[NTPSecondaryServerIP]**	Defines a second NTP server's address as an FQDN or an IP address (in dotted-decimal notation). This NTP is used for redundancy; if the primary NTP server fails, then this NTP server is used. The default IP address is 0.0.0.0.
Web: NTP UTC Offset EMS: UTC Offset CLI: utc-offset **[NTPServerUTCOffset]**	Defines the Universal Time Coordinate (UTC) offset (in seconds) from the NTP server. The default offset is 0. The offset range is -43200 to 43200. **Note:** The offset setting is applied only on the hour. For example, if you configure this parameter at 15:42, the device applies the setting only at 16:00.
Web: NTP Update Interval EMS: Update Interval CLI: update-interval **[NTPUpdateInterval]**	Defines the time interval (in seconds) that the NTP client requests for a time update. The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647. **Note:** It is not recommend to set this parameter to beyond one month (i.e., 2592000 seconds).

Parameter	Description
Web: NTP Authentication Key Identifier CLI: configure system > ntp > auth-key-id **[NtpAuthKeyId]**	Defines the NTP authentication key identifier for authenticating NTP messages. The identifier must match the value configured on the NTP server. The NTP server may have several keys configured for different clients; this number identifies which key is used. The valid value is 1 to 65535. The default is 0 (i.e., no authentication is done).
Web: NTP Authentication Secret Key CLI: configure system > ntp > auth-key-md5 **[ntpAuthMd5Key]**	Defines the secret authentication key shared between the device (client) and the NTP server, for authenticating NTP messages. The valid value is a string of up to 32 characters. By default, no key is defined.
Daylight Saving Time Parameters	
Web: Day Light Saving Time EMS: Mode CLI: summer-time **[DayLightSavingTimeEnable]**	Enables daylight saving time. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Start Time / Day of Month Start EMS: Start CLI: start **[DayLightSavingTimeStart]**	Defines the date and time when daylight saving begins. This value can be configured using any of the following formats: ▪ Day of year - *mm:dd:hh:mm*, where: ✓ *mm* denotes month ✓ *dd* denotes date of the month ✓ *hh* denotes hour ✓ *mm* denotes minutes For example, "05:01:08:00" denotes daylight saving starting from May 1 at 8 A.M. ▪ Day of month - *mm:day/wk:hh:mm*, where: ✓ *mm* denotes month (e.g., 04) ✓ *day* denotes day of week (e.g., FRI) ✓ *wk* denotes week of the month (e.g., 03) ✓ *hh* denotes hour (e.g., 23) ✓ *mm* denotes minutes (e.g., 10) For example, "04:FRI/03:23:00" denotes Friday, the third week of April, at 11 P.M. The week field can be 1-5, where 5 denotes the last occurrence of the specified day in the specified month. For example, "04:FRI/05:23:00" denotes the last Friday of April, at 11 P.M.
Web: End Time / Day of Month End EMS: End CLI: end **[DayLightSavingTimeEnd]**	Defines the date and time when daylight saving ends. For a description of the format of this value, see the DayLightSavingTimeStart parameter.
Web/EMS: Offset CLI: offset **[DayLightSavingTimeOffset]**	Defines the daylight saving time offset (in minutes). The valid range is 0 to 120. The default is 60. **Note:** The offset setting is applied only on the hour. For example, if you configure this parameter at 15:42, the device applies the setting only at 16:00.

67.2.10 Power over Ethernet Parameters

The power-over-Ethernet (PoE) parameters are described in the table below.

Table 67-19: PoE Parameters

Parameter	Description
Web: Power over Ethernet Settings CLI: configure system > interface poe-table **[POETable]**	This table enables PoE (IEEE 802.3af-2003 and IEEE 802.3at) per LAN port and configures the maximum power consumption allowed per port for Class 0 clients connected to it. [POETable] FORMAT POETable_Index = POETable_PortEnable, POETable_PortPower, POETable_PortATEnable; [\POETable] Where: ▪ Index = Port number (where 0 is Port 1) ▪ PortEnable = enables [1] or disables [0] IEEE 802.3af-2003 PoE ▪ PortPower = defines maximum power consumption ▪ PortATEnable = enables [1] or disables [0] IEEE 802.3at PoE **Note:** For a description of this parameter, see "Configuring Power over Ethernet" on page 125.

67.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

67.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

Table 67-20: General Debugging and Diagnostic Parameters

Parameter	Description
CLI: enablesecsyslog [EnableSecSyslog]	Enables the reporting of security-related events for the data-router networking. When enabled, the data-router access list rules, configured using the access-list CLI command, which are set to "log", send Syslog messages whenever traffic matching the access list is encountered. ▪ [0] = (Default) Disabled ▪ [1] = Enabled
EMS: Enable Diagnostics **[EnableDiagnostics]**	Determines the method for verifying correct functioning of the different hardware components on the device. On completion of the check and if the test fails, the device sends information on the test results of each hardware component to the Syslog server. ▪ **[0]** = (Default) Rapid and Enhanced self-test mode. ▪ **[1]** = Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY and Flash). ▪ **[2]** = A quicker version of the Detailed self-test mode (full test of DSPs, PCM, Switch, LAN, PHY, but partial test of Flash). **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Web: Enable LAN Watchdog [EnableLanWatchDog]	Enables the LAN watchdog feature, which detects any logical network failure. ▪ [0] Disable (default) ▪ [1] Enable & Reset = Enables LAN watchdog. If the device detects a network failure, the device resets. ▪ [2] Enable & No Reset = Enables LAN watchdog. If the device detects a network failure, it does not undergo a reset. The LAN watchdog periodically checks the device's overall communication integrity by pinging the network. If it detects a communication failure for more than three minutes, the device performs a self test: ▪ Test succeeds: The problem is due to a logical link failure (i.e., Ethernet cable disconnected on the remote switch) and the following mechanisms are activated if enabled: ✓ Busy Out mechanism (see the EnableBusyOut parameter) ✓ Lifeline mechanism (configured by the LifeLineType parameter) ▪ Test fails: The device resets (if this parameter is set to [2]) to overcome the internal communication error. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ LAN watchdog is applicable only if the Ethernet connection is full duplex.
[LifeLineType]	Defines the condition(s) upon which the Lifeline analog (FXS) feature is activated. The Lifeline feature can be activated upon a power outage or network failure (i.e., loss of IP connectivity). Upon any of these conditions, the Lifeline feature provides PSTN connectivity and thus call continuity for the FXS phone users. If the device is in Lifeline mode and the scenario that caused it to enter Lifeline (e.g., power outage) no longer exists (e.g., power returns), the device exists Lifeline and operates as normal. ▪ [0] = (Default) Lifeline is activated upon power outage. ▪ [1] = Lifeline is activated upon power outage. ▪ [2] = Lifeline is activated upon a power outage network failure (logical link disconnection), or when the Trunk Group is in Busy Out state (see the EnableBusyOut parameter). The Lifeline (FXS) phone is connected to the following port: ▪ FXS Port 1 FXS Port 1 connects to the POTS (Lifeline) phone as well as to the PSTN / PBX, using a splitter cable. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to FXS interfaces. ▪ To enable Lifeline upon network failure, the LAN Watchdog feature must be activated (see the EnableLANWatchDog parameter). ▪ The number of supported Lifelines depends on the device’s hardware configuration. For the combined FXS/FXO configuration, one Lifeline is available; for the 12-FXS configuration, up to three Lifelines are available. ▪ For information on Lifeline cabling, refer to the Installation

Parameter	Description
	Manual.
Web: Delay After Reset [sec] CLI: delay-after-reset **[GWAppDelayTime]**	Defines the time interval (in seconds) that the device's operation is delayed after a reset. The valid range is 0 to 45. The default is 7 seconds. **Note:** This feature helps overcome connection problems caused by some LAN routers or IP configuration parameters' modifications by a DHCP server.
[EnableAutoRAITransmitBER]	Enables the device to send a remote alarm indication (RAI) when the bit error rate (BER) is greater than 0.001. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Ignore BRI LOS Alarm CLI: ignore-bri-los-alarm [IgnoreBRILOSAlarm]	Enables the device to ignore LOS alarms received from the BRI user-side trunk and attempts to make a call (relevant for IP-to-Tel calls). ▪ [0] Disable ▪ [1] Enable (default) **Note:** This parameter is applicable only to BRI interfaces.

67.3.2 SIP Test Call Parameters

The SIP Signaling Test Call parameters are described in the table below.

Table 67-21: SIP Test Call Parameters

Parameter	Description
Web: Test Call DTMF String CLI:testcall-dtmf-string **[TestCallDtmfString]**	Defines the DTMF tone that is played for answered test calls (incoming and outgoing). The DTMF string can be up to 15 strings. The default is "3212333". If no string is defined (empty), DTMF is not played.
Web: Test Call ID CLI: testcall-id **[TestCallID]**	Defines the test call prefix number (*ID*) of the simulated phone on the device. Incoming calls received with this called prefix number are identified as test calls. This can be any string of up to 15 characters. By default, no number is defined. **Notes:** ▪ This parameter is only for testing incoming calls destined to this prefix number. ▪ This feature is applicable to all applications (Gateway and SBC).

Parameter	Description
Web: SBC Test ID CLI: sbc-test-id [SBCtestID]	Defines the SBC test call prefix (ID) for identifying SBC test calls that traverse the device to register with an external routing entity such as an IP PBX or proxy server. This parameter functions together with the TestCallID parameter, which defines the prefix of the simulated endpoint. Upon receiving an incoming call with this prefix, the device removes the prefix, enabling it to forward the test call to the external entity. Upon receiving the call from the external entity, the device identifies the call as a test call according to its prefix, defined by the TestCallID, and then sends the call to the simulated endpoint. For example, assume SBCTestID is set to 4 and TestCallID to 2. If a call is received with called destination 4200, the device removes the prefix 4 and routes the call to the IP PBX. When it receives the call from the IP PBX, it identifies the call as a test call (i.e., prefix 2) and therefore, sends it to the simulated endpoint. The valid value can be any string of up to 15 characters. By default, no number is defined. **Note:** This feature is applicable only to the SBC application.
Test Call Table	
Web: Test Call Table CLI: configure system > test-call > test-call-table **[Test_Call]**	Defines the local and remote endpoints to be tested. [Test_Call] FORMAT Test_Call_Index = Test_Call_EndpointURI, Test_Call_CalledURI, Test_Call_RouteBy, Test_Call_IPGroupID, Test_Call_DestAddress, Test_Call_DestTransportType, Test_Call_SRD, Test_Call_ApplicationType, Test_Call_AutoRegister, Test_Call_UserName, Test_Call_Password, Test_Call_CallParty, Test_Call_MaxChannels, Test_Call_CallDuration, Test_Call_CallsPerSecond, Test_Call_TestMode, Test_Call_TestDuration, Test_Call_Play, Test_Call_ScheduleInterval, Test_Call_QOEProfile, Test_Call_BWProfile; [\Test_Call] For a description of this table, see "Configuring Test Call Endpoints" on page 739.

67.3.3 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table 67-22: Syslog, CDR and Debug Parameters

Parameter	Description
Web: Enable Syslog EMS: Syslog enable CLI: syslog **[EnableSyslog]**	Determines whether the device sends logs and error messages (e.g., CDRs) generated by the device to a Syslog server. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ If you enable Syslog, you must enter an IP address of the Syslog server (using the SyslogServerIP parameter). ▪ Syslog messages may increase the network traffic. ▪ To configure Syslog SIP message logging levels, use the GwDebugLevel parameter.

Parameter	Description
	▪ By default, logs are also sent to the RS-232 serial port. For how to establish serial communication with the device, refer to the Installation Manual.
Web/EMS: Syslog Server IP Address CLI: syslog-ip **[SyslogServerIP]**	Defines the IP address (in dotted-decimal notation) of the computer on which the Syslog server is running. The Syslog server is an application designed to collect the logs and error messages generated by the device. The default IP address is 0.0.0.0.
Web: Syslog Server Port EMS: Syslog Server Port Number CLI: syslog-port **[SyslogServerPort]**	Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514.
Web: CDR Server IP Address EMS: IP Address of CDR Server CLI: cdr-srvr-ip-adrr **[CDRSyslogServerIP]**	Defines the destination IP address to where CDR logs are sent. The default value is a null string, which causes CDR messages to be sent with all Syslog messages to the Syslog server. **Notes:** ▪ The CDR messages are sent to UDP port 514 (default Syslog port). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web/EMS: CDR Report Level CLI: cdr-report-level **[CDRReportLevel]**	Enables media and signaling-related CDRs to be sent to a Syslog server and determines the call stage at which they are sent. ▪ **[0]** None = (Default) CDRs are not used. ▪ **[1]** End Call = CDR is sent to the Syslog server at the end of each call. ▪ **[2]** Start & End Call = CDR report is sent to Syslog at the start and end of each call. ▪ **[3]** Connect & End Call = CDR report is sent to Syslog at connection and at the end of each call. ▪ **[4]** Start & End & Connect Call = CDR report is sent to Syslog at the start, at connection, and at the end of each call. **Notes:** ▪ For the SBC application, this parameter enables only signaling-related CDRs. To enable media-related CDRs for SBC calls, use the MediaCDRReportLevel parameter. ▪ The CDR Syslog message complies with RFC 3161 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational). ▪ This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1).
Web: Media CDR Report Level [MediaCDRReportLevel]	Enables media-related CDRs of SBC calls to be sent to a Syslog server and determines the call stage at which they are sent. ▪ [0] None = (Default) No media-related CDR is sent. ▪ [1] End Media = Sends a CDR only at the end of the call. ▪ [2] Start & End Media = Sends a CDR once the media starts. In some calls it may only be after the call is established, but in other calls the media may start at ringback tone. A CDR is also sent upon termination (end) of the media in the call. ▪ [3] Update & End Media = Sends a CDR when an update occurs

Parameter	Description
	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.
Web/EMS: Debug Level CLI: configure system/logging/debug-level **[GwDebugLevel]**	Enables Syslog debug reporting and logging level. ▪ **[0]** No Debug = (Default) Debug is disabled. ▪ **[1]** Basic = Sends debug logs of incoming and outgoing SIP messages. ▪ **[5]** Detailed = Sends debug logs of incoming and outgoing SIP message as well as many other logged processes. **Note:** When debug reporting is enabled, in order to view Syslog messages with Wireshark, you need to install AudioCodes Wireshark plug-in (acsyslog.dll). Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and are displayed using the ‘acsyslog’ filter instead of the regular ‘syslog’ filter.
Web: Syslog Optimization CLI: configure system/logging/syslog-optimization **[SyslogOptimization]**	Enables the device to accumulate and bundle multiple debug messages into a single UDP packet and then send it to a Syslog server. The benefit of this feature is that it reduces the number of UDP Syslog packets, thereby improving (optimizing) CPU utilization. ▪ **[0]** Disable ▪ **[1]** Enable (default) **Note:** The size of the bundled message is configured by the MaxBundleSyslogLength parameter.
CLI: mx-syslog-lgth **[MaxBundleSyslogLength]**	Defines the maximum size (in bytes) threshold of logged Syslog messages bundled into a single UDP packet, after which they are sent to a Syslog server. The valid value range is 0 to 1220 (where 0 indicates that no bundling occurs). The default is 1220. **Note:** This parameter is applicable only if the GWDebugLevel parameter is enabled.
Web: Syslog CPU Protection CLI: configure system/logging/syslog-cpu-protection **[SyslogCpuProtection]**	Enables the protection of the device's CPU resources during debug reporting, ensuring voice traffic is unaffected. If CPU resources drop (i.e., high CPU usage) to a critical level (threshold), the device automatically lowers the debug level to free up CPU resources that were required for the previous debug-level functionality. When sufficient CPU resources become available again, the device increases the debug level. The threshold is configured by the 'Debug Level High Threshold' parameter (see below). ▪ **[0]** Disable ▪ **[1]** Enable (default)

Parameter	Description
Web: Debug Level High Threshold CLI: debug-level-high-threshold [DebugLevelHighThreshold]	Defines the threshold (in percentage) for automatically switching to a different debug level, depending on CPU usage. The parameter is applicable only if the 'Syslog CPU Protection' parameter is enabled. The valid value is 0 to 100. The default is 90. The debug level is changed upon the following scenarios: ▪ CPU usage equals threshold: Debug level is reduced one level. ▪ CPU usage is at least 5% greater than threshold: Debug level is reduced another level. ▪ CPU usage is 5 to 19% less than threshold: Debug level is increased by one level. ▪ CPU usage is at least 20% less than threshold: Debug level is increased by another level. For example, assume that the threshold is set to 70% and the Debug Level to Detailed (5). When CPU usage reaches 70%, the debug level is reduced to Basic (1). When CPU usage increases by 5% or more than the threshold (i.e., greater than 75%), the debug level is disabled **-** No Debug (0). When the CPU usage decreases to 5% less than the threshold (e.g., 65%), the debug level is increased to Basic (1). When the CPU usage decreases to 20% less than the threshold (e.g., 50%), the debug level changes to Detailed (5). **Note:** The device does not increase the debug level to a level that is higher than what you configured for the 'Debug Level' parameter.
Web: Syslog Facility Number EMS: SyslogFacility **[SyslogFacility]**	Defines the Facility level (0 through 7) of the device's Syslog messages, according to RFC 3164. This allows you to identify Syslog messages generated by the device. This is useful, for example, if you collect the device's and other equipments' Syslog messages, at one single server. The device's Syslog messages can easily be identified and distinguished from other Syslog messages by its Facility level. Therefore, in addition to filtering Syslog messages according to IP address, the messages can be filtered according to Facility level. ▪ **[16]** = (Default) local use 0 (local0) ▪ **[17]** = local use 1 (local1) ▪ **[18]** = local use 2 (local2) ▪ **[19]** = local use 3 (local3) ▪ **[20]** = local use 4 (local4) ▪ **[21]** = local use 5 (local5) ▪ **[22]** = local use 6 (local6) ▪ **[23]** = local use 7 (local7)
Web: CDR Session ID CLI: cdr-seq-num **[CDRSyslogSeqNum]**	Enables or disables the inclusion of the sequence number (S=) in CDR Syslog messages. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Web: Activity Types to Report via Activity Log Messages CLI: config-system > logging > activity-log **[ActivityListToLog]**	Defines the operations (activities) in the Web interface that are reported to a Syslog server. ▪ **[pvc]** Parameters Value Change = Changes made on-the-fly to parameters. Note that the *ini* file parameter, EnableParametersMonitoring can also be used to set this option, using values **[0]** (disable) or **[1]** (enable). ▪ **[afl]** Auxiliary Files Loading = Loading of auxiliary files. ▪ **[dr]** Device Reset = Resetting of the device through the

Parameter	Description
	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. **Note:** For the *ini* file parameter, enclose values in single quotation marks, for example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc'.
CLI: oamp-default-network-src data/voip [OAMPDefaultNetworkSource]	Defines the network interface from where the device sends Syslog messages to a Syslog server. - [0] Data (default) = Syslog messages are sent from the WAN interface. - [1] VoIP= Syslog messages are sent from the VoIP LAN interface for OAMP.
CLI: isdn-facility-trace [FacilityTrace]	Enables ISDN traces of Facility Information Elements (IE) for ISDN call diagnostics. This allows you to trace all the parameters contained in the Facility IE and view them in the Syslog. - [0] Disable (default) - [1] Enable **Note:** For this feature to be functional, the GWDebugLevel parameter must be enabled (i.e., set to at least level 1).
Web: Debug Recording Destination IP CLI: configure system > logging > dbg-rec-dest-ip **[DebugRecordingDestIP]**	Defines the IP address of the server for capturing debug recording.
Web: Debug Recording Destination Port CLI: configure system > logging > dbg-rec-dest-port **[DebugRecordingDestPort]**	Defines the UDP port of the server for capturing debug recording. The default is 925.

Parameter	Description
Debug Recording Status CLI: configure system > logging > dbg-rec-status **[DebugRecordingStatus]**	Activates or de-activates debug recording. ▪ **[0]** Stop (default) ▪ **[1]** Start
Web: Enable Core Dump **[EnableCoreDump]**	Enables the automatic generation of a Core Dump file upon a device crash. ▪ **[0]** Disable (disable) ▪ **[1]** Enable
Web: Core Dump Destination IP **[CoreDumpDestIP]**	Defines the IP address of the remote server where you want the device to send the Core Dump file. By default, no IP address is defined.
Logging Filters Table	
Web: Logging Filters Table CLI: configure system > logging > logging-filters **[LoggingFilters]**	This table parameter defines logging filtering rules for Syslog messages and debug recordings. The format of the ini file table parameter is: [LoggingFilters] FORMAT LoggingFilters_Index = LoggingFilters_FilterType, LoggingFilters_Value, LoggingFilters_Syslog, LoggingFilters_CaptureType; [\LoggingFilters] For a detailed description of this table, see "Filtering Syslog Messages and Debug Recordings" on page 722.

67.3.4 Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

Table 67-23: RAI Parameters

Parameter	Description
[EnableRAI]	Enables Resource Available Indication (RAI) alarm generation if the device's busy endpoints exceed a user-defined threshold, configured by the RAIHighThreshold parameter. When enabled and the threshold is crossed, the device sends the SNMP trap, acBoardCallResourcesAlarm. ▪ **[0]** = (Default) Disable ▪ **[1]** = Enable **Note:** For this parameter to take effect, a device reset is required.
[RAIHighThreshold]	Defines the high threshold percentage of total calls that are active (busy endpoints). When the percentage of the device's busy endpoints exceeds this high threshold, the device sends the SNMP acBoardCallResourcesAlarm alarm trap with a 'major' alarm status. 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 (trunks are physically connected and synchronized with no alarms and endpoints are defined in the Trunk Group table).

Parameter	Description
[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.

67.4 Security Parameters

This subsection describes the device's security parameters.

67.4.1 General Security Parameters

The general security parameters are described in the table below.

Table 67-24: General Security Parameters

Parameter	Description
Firewall Table	
Web/EMS: Internal Firewall Parameters CLI: configure voip > access-list **[AccessList]**	This table parameter defines the device's access list (firewall), which defines network traffic filtering rules. The format of this parameter is as follows: [AccessList] FORMAT AccessList_Index = AccessList_Source_IP, AccessList_Source_Port, AccessList_PrefixLen, AccessList_Source_Port, AccessList_Start_Port, AccessList_End_Port, AccessList_Protocol, AccessList_Use_Specific_Interface, AccessList_Interface_ID, AccessList_Packet_Size, AccessList_Byte_Rate, AccessList_Byte_Burst, AccessList_Allow_Type; [\AccessList] For example: AccessList 10 = mgmt.customer.com, , , 32, 0, 80, tcp, 1, OAMP, 0, 0, 0, allow; AccessList 22 = 10.4.0.0, , , 16, 4000, 9000, any, 0, , 0, 0, 0, block; In the example above, Rule #10 allows traffic from the host ‘mgmt.customer.com’ destined to TCP ports 0 to 80 on interface OAMP (OAMP). Rule #22 blocks traffic from the subnet 10.4.xxx.yyy destined to ports 4000 to 9000. For a detailed description of this table, see "Configuring Firewall Settings" on page 159.
Media Latching	
Web/EMS: Inbound Media Latch Mode CLI: inbound-media-latch-mode [InboundMediaLatchMode]	Enables the Media Latching feature. ▪ [0] Strict = Device latches onto the first original stream (IP address:port). It does not latch onto any other stream during the session. ▪ [1] Dynamic = (Default) Device latches onto the first stream. If it

Parameter	Description
	receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) from a different source(s) and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other stream. If other packets of a different media type are received from the new stream, based on IP address and SSRC for RTCP/RTP and based on IP address only for T.38, the packet is accepted immediately. Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream. ▪ [2] Dynamic-Strict = Device latches onto the first stream. If it receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) all from the same source which is different to the first stream and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other stream. If other packets of different media type are received from the new stream based on IP address and SSRC for RTCP and based on IP address only for T.38, the packet is accepted immediately. Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream. ▪ [3] Strict-On-First = Typically used for NAT, where the correct IP address:port is initially unknown. The device latches onto the stream received in the first packet. The device does not change this stream unless a packet is later received from the original source.
New RTP Stream Packets [NewRtpStreamPackets]	Defines the minimum number of continuous RTP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
New RTCP Stream Packets [NewRtcpStreamPackets]	Defines the minimum number of continuous RTCP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
New SRTP Stream Packets [NewSRTPStreamPackets]	Defines the minimum number of continuous SRTP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
New SRTCP Stream Packets [NewSRTCPStreamPackets]	Defines the minimum number of continuous SRTCP packets received by the device's channel to allow latching onto the new incoming stream. The valid range is 0 to 20. The default is 3. If set to 0, the device is left exposed to attacks against multiple packet streams.
Timeout To Relatch RTP (msec) [TimeoutToRelatchRTPMsec]	Defines a period (msec) during which if no packets are received from the current RTP session, the channel can re-latch onto

Parameter	Description
	another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch SRTP [TimeoutToRelatchSRTPMsec]	Defines a period (msec) during which if no packets are received from the current SRTP session, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch Silence [TimeoutToRelatchSilenceMsec]	Defines a period (msec) during which if no packets are received from the current RTP/SRTP session and the channel is in silence mode, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.
Timeout To Relatch RTCP [TimeoutToRelatchRTCPMsec]	Defines a period (msec) during which if no packets are received from the current RTCP session, the channel can re-latch onto another RTCP stream. The valid range is any value from 0. The default is 10,000.
Fax Relay Rx/Tx Timeout [FaxRelayTimeoutSec]	Defines a period (sec) during which if no T.38 packets are received or sent from the current T.38 fax relay session, the channel can re-latch onto another stream. The valid range is 0 to 255. The default is 10.

67.4.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

Table 67-25: HTTPS Parameters

Parameter	Description
Web: Secured Web Connection (HTTPS) EMS: HTTPS Only CLI: secured-connection **[HTTPSOnly]**	Determines the protocol used to access the Web interface. ▪ **[0]** HTTP and HTTPS (default). ▪ **[1]** HTTPs Only = Unencrypted HTTP packets are blocked. **Note:** For this parameter to take effect, a device reset is required.
EMS: HTTPS Port CLI: https-port **[HTTPSPort]**	Defines the local Secured HTTPS port of the device. This parameter allows secure remote device Web management from the LAN. To enable secure Web management from the LAN, configure the desired port. The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: HTTPS Cipher String CLI: https-cipher-string **[HTTPSCipherString]**	Defines the Cipher string for HTTPS (in OpenSSL cipher list format). For the valid range values, refer to URL http://www.openssl.org/docs/apps/ciphers.html. The default is ‘RC4:EXP’ (Export encryption algorithms). For example, use ‘ALL’ for all ciphers suites (e.g., for ARIA encryption for TLS). The only ciphers available are RC4 and DES, and the cipher bit strength is limited to 56 bits. **Notes:**

Parameter	Description
	- For this parameter to take effect, a device reset is required. - If the installed Software License Key includes the Strong Encryption feature, the default of this parameter is changed to 'RC4:EXP', enabling RC-128bit encryption. - The value 'ALL' can be configured only if the installed Software License Key includes the Strong Encryption feature.
Web: HTTP Authentication Mode EMS: Web Authentication Mode CLI: http-auth-mode **[WebAuthMode]**	Determines the authentication mode used for the Web interface. - **[0]** Basic Mode =Basic authentication (clear text) is used. - **[1]** Web Based Authentication = (Default) Digest authentication (MD5) is used. **Note:** If you enable RADIUS login (i.e., the WebRADIUSLogin parameter is set to 1), you must set the WebAuthMode parameter to Basic Mode [0].
Web: Requires Client Certificates for HTTPS connection CLI: req-client-cert **[HTTPSRequireClientCertificate]**	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 this parameter to take effect, a device reset is required. - For a description on implementing client certificates, see "TLS for Remote Device Management" on page 119.

67.4.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

Table 67-26: SRTP Parameters

Parameter	Description
Web: Media Security EMS: Enable Media Security CLI: media-security-enable **[EnableMediaSecurity]**	Enables Secure Real-Time Transport Protocol (SRTP). - **[0]** Disable (default) - **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Media Security Behavior CLI: media-sec-bhvior **[MediaSecurityBehaviour]**	Global parameter that defines the handling of SRTP (when the EnableMediaSecurity parameter is set to 1). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MediaSecurityBehaviour). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **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. **Note:** This parameter is applicable only to the Gateway application.

Parameter	Description
Web: Master Key Identifier (MKI) Size EMS: Packet MKI Size CLI: SRTP-tx-packet-MKI-size **[SRTPTxPacketMKISize]**	Global parameter that defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MKISize). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Symmetric MKI Negotiation EMS: Enable Symmetric MKI CLI: symmetric-mki **[EnableSymmetricMKI]**	Global parameter that enables symmetric MKI negotiation. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableSymmetricMKI). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web/EMS: Offered SRTP Cipher Suites CLI: offer-srtp-cipher **[SRTPofferedSuites]**	Defines the offered crypto suites (cipher encryption algorithms) for SRTP. ▪ **[0]** All = (Default) All available crypto suites. ▪ **[1]** AES-CM-128-HMAC-SHA1-80 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 80-bit tag. ▪ **[2]** AES-CM-128-HMAC-SHA1-32 = device uses AES-CM encryption with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag. ▪ **[4]** ARIA-CM-128-HMAC-SHA1-80 = device uses ARIA encryption algorithm with a 128-bit key and HMAC-SHA1 message authentication with a 32-bit tag. ▪ **[8]** ARIA-CM-192-HMAC-SHA1-80 = device uses ARIA encryption algorithm with a 192-bit key and HMAC-SHA1 message authentication with a 32-bit tag. **Notes:** ▪ For enabling ARIA encryption, use the AriaProtocolSupport parameter. ▪ This parameter also affects the selection of the crypto in the device's answer. For example, if the device receives an offer with two crypto lines containing HMAC_SHA1_80 and HMAC_SHA_32, it uses the HMAC_SHA_32 key in its SIP 200 OK response if the parameter is set to 2.

Parameter	Description
Web: Aria Protocol Support CLI: ARIA-protocol-support [AriaProtocolSupport]	Enables ARIA algorithm cipher encryption for SRTP. This is an alternative option to the existing support for the AES algorithm. ARIA is a symmetric key block cipher algorithm standard developed by the Korean National Security Research Institute. ▪ [0] Disable (default) ▪ [1] Enable **Notes:** ▪ To configure the ARIA bit-key encryption size (128 or 192 bit) with HMAC SHA-1 cryptographic hash function, use the SRTPofferedSuites parameter. ▪ The ARIA 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 621.
Web: Disable Authentication On Transmitted RTP Packets EMS: RTP AuthenticationDisable Tx CLI: RTP-authentication-disable-tx **[RTPAuthenticationDisableTx]**	Enables authentication on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTP Packets EMS: RTP EncryptionDisable Tx CLI: RTP-encryption-disable-tx **[RTPEncryptionDisableTx]**	Enables encryption on transmitted RTP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
Web: Disable Encryption On Transmitted RTCP Packets EMS: RTCP EncryptionDisable Tx CLI: RTCP-encryption-disable-tx **[RTCPEncryptionDisableTx]**	Enables encryption on transmitted RTCP packets in a secured RTP session. ▪ **[0]** Enable (default) ▪ **[1]** Disable
CLI: srtp-state-behavior-mode **[ResetSRTPStateUponRekey]**	Global parameter that enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_ResetSRTPStateUponRekey). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **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.

67.4.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

Table 67-27: TLS Parameters

Parameter	Description
TLS Contexts Table	
Web: TLS Contexts Table CLI: 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 this table, see "Configuring TLS Certificate Contexts" on page 109.
Web: TLS Client Re-Handshake Interval EMS: TLS Re Handshake Interval CLI: tls-re-hndshk-int **[TLSReHandshakeInterval]**	Defines the time interval (in minutes) between TLS Re-Handshakes initiated by the device. The interval range is 0 to 1,500 minutes. The default is 0 (i.e., no TLS Re-Handshake).
Web: TLS Mutual Authentication EMS: SIPS Require Client Certificate **[SIPSRequireClientCertificate]**	Defines the device's mode of operation regarding mutual authentication and certificate verification for TLS connections. ▪ **[0]** Disable = (Default) ✓ Device acts as a client: Verification of the server's certificate depends on the VerifyServerCertificate parameter. ✓ Device acts as a server: The device does not request the client certificate. ▪ **[1]** Enable = ✓ Device acts as a client: Verification of the server certificate is required to establish the TLS connection. ✓ Device acts as a server: The device requires the receipt and verification of the client certificate to establish the TLS connection. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This feature can be configured per SIP Interface (see "Configuring SIP Interfaces" on page 267). ▪ The SIPS certificate files can be changed using the parameters HTTPSCertFileName and HTTPSRootFileName.
Web/EMS: 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

Parameter	Description
	server or client for the TLS connection. When the device receives a remote certificate and this parameter is not disabled, the IP address from which the certificate is received is compared with the addresses defined for the Proxy Sets. If no Proxy Set with the source address is found, the connection is refused. Otherwise, the value of SubjectAltName field in the certificate is compared with the addresses\ DNS Names of the classified Proxy Set. If a match is found for any of the configured Proxies, the TLS connection is established. The comparison is performed if the SubjectAltName is either a DNS name (DNSName) or an IP address. If no match is found and the SubjectAltName is marked as 'critical', the TLS connection is not established. If DNSName is used, the certificate can also use wildcards ('*') to replace parts of the domain name. If the SubjectAltName is not marked as 'critical' and there is no match, the CN value of the SubjectName field is compared with the parameter TLSRemoteSubjectName. If a match is found, the connection is established; otherwise, the connection is terminated. **Note:** If you set this parameter to [2] (Server & Client), for this functionality to operate you also need to set the SIPSRequireClientCertificate parameter to [1] (Enable).
Web: TLS Client Verify Server Certificate EMS: Verify Server Certificate CLI: tls-vrfy-srvr-cert **[VerifyServerCertificate]**	Determines whether the device, when acting as a client for TLS connections, verifies the Server certificate. The certificate is verified with the Root CA information. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** If Subject Name verification is necessary, the parameter PeerHostNameVerificationMode must be used as well.
Web: Strict Certificate Extension Validation CLI: require-strict-cert **[RequireStrictCert]**	Enables the validation of the extensions (keyUsage and extentedKeyUsage) of peer certificates. This validation ensures that the signing CA is authorized to sign certificates and that the end-entity certificate is authorized to negotiate a secure TLS connection. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: TLS Remote Subject Name CLI: tls-rmt-subs-name **[TLSRemoteSubjectName]**	Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. If the CN uses a domain name, the certificate can also use wildcards ('*') to replace parts of the domain name. The valid range is a string of up to 49 characters. **Note:** This parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.

Parameter	Description
Web: TLS Expiry Check Start CLI: expiry-check-start **[TLSExpiryCheckStart]**	Defines the number of days before the installed TLS server certificate is to expire at which the device must send a trap (acCertificateExpiryNotification) to notify of this. The valid value is 0 to 3650. The default is 60.
Web: TLS Expiry Check Period CLI: expiry-check-period **[TLSExpiryCheckPeriod]**	Defines the periodical interval (in days) for checking the TLS server certificate expiry date. The valid value is 1 to 3650. The default is 7.

67.4.5 SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

Table 67-28: SSH Parameters

Parameter	Description
Web/EMS: Enable SSH Server CLI: ssh **[SSHServerEnable]**	Enables the device's embedded SSH server. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: Server Port cli: ssh-port **[SSHServerPort]**	Defines the port number for the embedded SSH server. Range is any valid port number. The default port is 22.
Web/EMS: SSH Admin Key CLI: ssh-admin-key **[SSHAdminKey]**	Defines the RSA public key for strong authentication for logging in to the SSH interface (if enabled). The value should be a base64-encoded string. The value can be a maximum length of 511 characters.
Web: Require Public Key EMS: EMS: SSH Require Public Key CLI: ssh-require-public-key **[SSHRequirePublicKey]**	Enables RSA public keys for SSH. ▪ **[0]** = (Default) RSA public keys are optional if a value is configured for the parameter SSHAdminKey. ▪ **[1]** = RSA public keys are mandatory. **Note:** To define the key size, use the TLSPkeySize parameter.
Web: Max Payload Size EMS: SSH Max Payload Size CLI: ssh-max-payload-size **[SSHMaxPayloadSize]**	Defines the maximum uncompressed payload size (in bytes) for SSH packets. The valid value is 550 to 32768. The default is 32768.
Web: Max Binary Packet Size EMS: SSH Max Binary Packet Size CLI: ssh-max-binary-packet-size **[SSHMaxBinaryPacketSize]**	Defines the maximum packet size (in bytes) for SSH packets. The valid value is 582 to 35000. The default is 35000.
Web: Maximum SSH Sessions EMS: Telnet SSH Max Sessions CLI: ssh-max-sessions **[SSHMaxSessions]**	Defines the maximum number of simultaneous SSH sessions. The valid range is 1 to 5. The default is 2 sessions.
Web: Enable Last Login Message CLI: ssh-last-login-message **[SSHEnableLastLoginMessage]**	Enables message display in SSH sessions of the time and date of the last SSH login. The SSH login message displays the number of unsuccessful login attempts since the last successful login. ▪ **[0]** Disable ▪ **[1]** Enable (default)

Parameter	Description
	Note: The last SSH login information is cleared when the device is reset.
Web: Max Login Attempts CLI: ssh-max-login-attempts **[SSHMaxLoginAttempts]**	Defines the maximum SSH login attempts allowed for entering an incorrect password by an administrator before the SSH session is rejected. The valid range is 1 to 3. the default is 3.

67.4.6 TAACS+ Parameters

The TACACS+ parameters are described in the table below.

Table 67-29: TACACS+ Parameters

Parameter	Description
CLI: aaa authentication login tacacs+ [TacPlusEnable]	Enables the Terminal Access Controller Access-Control System (TACACS+) remote authentication protocol and user authentication for CLI login. ▪ [0] = Disabled (default) ▪ [1] = Enabled **Note:** For this parameter to take effect, a device reset is required.
CLI: tacacs-server host <host-ip> [TacPlusServerIP]	Defines the IP address (in dotted-decimal notation) of the TACACS+ primary authentication server.
CLI: tacacs-server host <host-ip> [TacPlusSecondaryServerIP]	Defines the IP address (in dotted-decimal notation) of the TACACS+ secondary authentication server.
CLI: tacacs-server port <port-num> [TacPlusPort]	Defines the TACACS+ authentication port (UDP) for authenticating with the RADIUS server. The valid value range is 1 to 15. The default is 49.
CLI: tacacs-server timeout <seconds> [TacPlusTimeout]	Defines the TACACS+ response timeout (in seconds). If no response is received within this period, retransmission is required. The valid value range is 1 to 15. The default is 5.
CLI: tacacs-server key <password> [TacPlusSharedSecretand]	Defines the TACACS+ shared secret between client and server. The valid value can be a string of up to 64 characters. The default is “msbg”.

67.4.7 IDS Parameters

The Intrusion Detection System (IDS) parameters are described in the table below.

Table 67-30: IDS Parameters

Parameter	Description
Web: Intrusion Detection System (IDS) CLI: enable-ids	Enables the IDS feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable

Parameter	Description
[EnableIDS]	**Note:** For this parameter to take effect, a device reset is required.
CLI: ids-clear-period **[IDSAlarmClearPeriod]**	Defines the interval (in seconds) after which an IDS alarm is cleared from the Active Alarms table if no thresholds are crossed during this time. However, this "quiet" period must be at least twice the Threshold Window value. For example, if IDSAlarmClearPeriod is set to 20 sec and the Threshold Window is set to 15 sec, the IDSAlarmClearPeriod parameter is ignored and the alarm is cleared only after 30 seconds (2 x 15 sec). The valid value is 0 to 86400. The default is 300.
IDS Policy Table	
Web: IDS Policy Table **[IDSPolicy]**	Defines IDS Policies. The format of the ini file parameter is: [IDSPolicy] FORMAT IDSPolicy_Index = IDSPolicy_Name, IDSPolicy_Description; [\IDSPolicy] For a detailed description of this table, see "Configuring IDS Policies" on page 165.
IDS Rule Table	
Web: IDS Rule Table **[IDSRule]**	Defines rules for IDS Policies. The format of the ini file parameter is: [IDSRule] FORMAT IDSRule_Index = IDSRule_Policy, IDSRule_RuleID, IDSRule_Reason, IDSRule_ThresholdScope, IDSRule_ThresholdWindow, IDSRule_MinorAlarmThreshold, IDSRule_MajorAlarmThreshold, IDSRule_CriticalAlarmThreshold, IDSRule_DenyThreshold, IDSRule_DenyPeriod; [\IDSRule] For a detailed description of this table, see "Configuring IDS Policies" on page 165.
IDS Match Table	
Web: IDS Match Table **[IDSMatch]**	Defines target rules per IDS Policy. The format of the ini file parameter is: [IDSMatch] FORMAT IDSMatch_Index = IDSMatch_SIPInterface, IDSMatch_ProxySet, IDSMatch_Subnet, IDSMatch_Policy; [\IDSMatch] For a detailed description of this table, see "Assigning IDS Policies" on page 169.

67.4.8 OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

Table 67-31: OCSP Parameters

Parameter	Description
Web: Enable OCSP Server EMS: OCSP Enable	Enables or disables certificate checking using OCSP. ▪ **[0]** Disable (default)

Parameter	Description
CLI: enable **[OCSPEnable]**	▪ **[1]** Enable For a description of OCSP, see Configuring Certificate Revocation Checking (OCSP).
Web: Primary Server IP EMS: OCSP Server IP CLI: server-ip **[OCSPServerIP]**	Defines the IP address of the OCSP server. The default IP address is 0.0.0.0.
Web: Secondary Server IP CLI: secondary-server-ip **[OCSPSecondaryServerIP]**	Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default IP address is 0.0.0.0.
Web: Server Port EMS: OCSP Server Port CLI: server-port **[OCSPServerPort]**	Defines the OCSP server's TCP port number. The default port number is 2560.
Web: Default Response When Server Unreachable EMS: OCSP Default Response CLI: default-response **[OCSPDefaultResponse]**	Determines whether the device allows or rejects peer certificates when the OCSP server cannot be contacted. ▪ **[0]** Reject (default) ▪ **[1]** Allow

67.5 Quality of Experience Parameters

The Quality of Experience (QoE) parameters are described in the table below.

Table 67-32: Quality of Experience Parameters

Parameter	Description
SEM Parameters	
Web: Server IP CLI: 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 this parameter to take effect, a device reset is required.
Web: Redundant Server IP CLI: configure voip > qoe configuration > set secondary-server-ip **[QOESecondaryServerIp]**	Defines the IP address of the secondary SEM server to where the quality experience reports are sent. This is applicable when the SEM/EMS server is in Geographical Redundancy HA mode. **Note:** For this parameter to take effect, a device reset is required.
Web: Port CLI: configure voip/qoe configuration/port **[QOEPort]**	Defines the port of the SEM server. The valid value range is 0 to 65534. The default is 5000.
Web: Interface Name CLI: configure voip/qoe configuration/interface-name **[QOEInterfaceName]**	Defines the IP network interface on which the quality experience reports are sent. The default is the OAMP interface. **Note:** For this parameter to take effect, a device reset is required.
Quality of Experience Profile Table	

Parameter	Description
Web: Quality of Experience Profile CLI: configure voip/qoe qoe-profile **[QOEProfile]**	This table parameter defines Quality of Experience Profiles. The format of the ini file table parameter is as follows: [QOEProfile] FORMAT QOEProfile_Index = QOEProfile_Name, QOEProfile_SensitivityLevel; [\QOEProfile] For a detailed description of this table, see "Configuring Quality of Experience Profiles" on page 248.
Quality of Experience Color Rules Table	
Web: Quality of Experience Color Rules CLI: configure voip/qoe qoe-profile qoe-color-rules **[QOEColorRules]**	This table parameter defines Quality of Experience Color Rules. The format of the ini file table parameter is as follows: [QOEColorRules] FORMAT QOEColorRules_Index = QOEColorRules_QoeProfile, QOEColorRules_ColorRuleIndex, QOEColorRules_monitoredParam, QOEColorRules_direction, QOEColorRules_profile, QOEColorRules_GreenYellowThreshold, QOEColorRules_GreenYellowHysteresis, QOEColorRules_YellowRedThreshold, QOEColorRules_YellowRedHysteresis; [\QOEColorRules] For a detailed description of this table, see "Configuring Quality of Experience Profiles" on page 248.
Bandwidth Profile Table	
Web: Bandwidth Profile CLI: configure voip/qoe bw-profile **[BWProfile]**	This table parameter defines Bandwidth Profiles. The format of the ini file table parameter is as follows: [BWProfile] FORMAT BWProfile_Index = BWProfile_Name, BWProfile_EgressAudioBandwidth, BWProfile_IngressAudioBandwidth, BWProfile_EgressVideoBandwidth, BWProfile_IngressVideoBandwidth, BWProfile_TotalEgressBandwidth, BWProfile_TotalIngressBandwidth, BWProfile_WarningThreshold, BWProfile_hysteresis, BWProfile_GenerateAlarms; [\BWProfile] For a detailed description of this table, see "Configuring Bandwidth Profiles" on page 252. **Note:** For this parameter to take effect, a device reset is required.
Media Enhancement Profile Table	
Web: Media Enhancement Profile CLI: configure voip/qoe media-enhancement **[MediaEnhancementProfile]**	This table parameter defines Media Enhancement Profiles. The format of the ini file table parameter is as follows: [MediaEnhancementProfile] FORMAT MediaEnhancementProfile_Index = MediaEnhancementProfile_ProfileName; [\MediaEnhancementProfile] For a detailed description of this table, see "Configuring Media Enhancement Profiles" on page 255.
Media Enhancement Rules Table	

Parameter	Description
Web: Media Enhancement Rules CLI: configure voip/qoe media-enhancement-rules **[MediaEnhancementRules]**	This table parameter defines Media Enhancement Rules. The format of the ini file table parameter is as follows: [MediaEnhancementRules] FORMAT MediaEnhancementRules_Index = MediaEnhancementRules_MediaEnhancementProfile, MediaEnhancementRules_RuleIndex, MediaEnhancementRules_Trigger, MediaEnhancementRules_Color, MediaEnhancementRules_ActionRule, MediaEnhancementRules_ActionValue; [\MediaEnhancementRules] For a detailed description of this table, see "Configuring Media Enhancement Profiles" on page 255.

67.6 Control Network Parameters

67.6.1 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

Table 67-33: Proxy, Registration and Authentication SIP Parameters

Parameter	Description
IP Group Table	
Web: IP Group Table EMS: Endpoints > IP Group CLI: configure voip > voip-network ip-group **[IPGroup]**	This table configures IP Groups. The ini file format of this parameter is as follows: [IPGroup] FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Description, IPGroup_ProxySetId, IPGroup_SIPGroupName, IPGroup_ContactUser, IPGroup_EnableSurvivability, IPGroup_ServingIPGroup, IPGroup_SipReRoutingMode, IPGroup_AlwaysUseRouteTable, IPGroup_RoutingMode, IPGroup_SRD, IPGroup_MediaRealm, IPGroup_ClassifyByProxySet, IPGroup_ProfileId, IPGroup_MaxNumOfRegUsers, IPGroup_InboundManSet, IPGroup_OutboundManSet, IPGroup_RegistrationMode, IPGroup_AuthenticationMode, IPGroup_MethodList, IPGroup_EnableSBCClientForking, IPGroup_SourceUriInput, IPGroup_DestUriInput, IPGroup_ContactName, IPGroup_Username, IPGroup_Password, IPGroup_UUIFormat, IPGroup_QOEProfile, IPGroup_BWProfile, IPGroup_MediaEnhancementProfile, IPGroup_AlwaysUseSourceAddr; [/IPGroup] For a description of this table, see "Configuring IP Groups" on page 270. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
Authentication per Port Table	
Web: Authentication Table EMS: SIP Endpoints > Authentication CLI: configure voip/gw analoggw authentication [Authentication]	This table parameter defines a user name and password for authenticating each device port. The format of the ini file table parameter is as follows: [Authentication] FORMAT Authentication_Index = Authentication_UserId, Authentication_UserPassword, Authentication_Module, Authentication_Port; [\Authentication] Where, ▪ Module = Module number, where 1 denotes the module in Slot 1 ▪ Port = Port number, where 1 denotes the Port 1 of the module For example: Authentication 1 = lee,1552,1,2; (user name "lee" with password 1552 for authenticating Port 2 of Module 1) For a description o this table, see Configuring Authentication on page 477. **Note:** This parameter is applicable only to FXS and FXO interfaces.
Account Table	
Web: Account Table EMS: SIP Endpoints > Account CLI: configure voip > sip-definition account **[Account]**	This table parameter configures the Account table for registering and/or authenticating (digest) Trunk Groups or IP Groups (e.g., an IP-PBX) to another Serving IP Group (e.g., an Internet Telephony Service Provider - ITSP). The format of the ini file table parameter is as follows: [Account] FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroup, Account_ServingIPGroup, Account_Username, Account_Password, Account_HostName, Account_Register, Account_ContactUser, Account_ApplicationType; [\Account] For a detailed description of this table, see "Configuring Registration Accounts" on page 291.
Proxy Registration Parameters	
Web: Use Default Proxy EMS: Proxy Used CLI: enable-proxy [IsProxyUsed]	Enables the use of Proxy Set ID 0 (for backward compatibility). ▪ [0] No = (Default) Proxy Set 0 is not used. ▪ [1] Yes = Proxy Set ID 0 is used. **Notes:** ▪ This parameter must be used only for backward compatibility. If not required for backward compatibility, make sure that this parameter is disabled, and use the Proxy Set table for configuring all your Proxy Sets (except for Proxy Set ID 0). ▪ If you are not using a proxy server, you must configure routing rules to route the call. ▪ This parameter is applicable only to the Gateway application.

Parameter	Description
Web/EMS: Proxy Name CLI: proxy-name **[ProxyName]**	Defines the Home Proxy domain name. If specified, this name is used as the Request-URI in REGISTER, INVITE and other SIP messages, and as the host part of the To header in INVITE messages. If not specified, the Proxy IP address is used instead. The valid value is a string of up to 49 characters. **Note:** This parameter functions together with the UseProxyIPasHost parameter.
Web: Use Proxy IP as Host CLI: use-proxy-ip-as-host **[UseProxyIPasHost]**	Enables the use of the proxy server's IP address (in dotted-decimal notation) as the host name in SIP From and To headers in REGISTER requests. ▪ **[0]** Disable (default) ▪ **[1]** Enable If this parameter is disabled and the device registers to an IP Group (i.e., proxy server), it uses the string configured by the ProxyName parameter as the host name in the REGISTER's Request-URI and uses the string configured by the IP Group table parameter, SIPGroupName as the host name in the To and From headers. If the IP Group is configured with a Proxy Set that has multiple IP addresses, all the REGISTER messages sent to these proxies are sent with the same host name. **Note:** If this parameter is disabled and the ProxyName parameter is not configured, the proxy's IP address is used as the host name in the REGISTER Request-URI.
Web: Redundancy Mode EMS: Proxy Redundancy Mode CLI: redundancy-mode **[ProxyRedundancyMode]**	Determines whether the device switches back to the primary Proxy after using a redundant Proxy. ▪ **[0]** Parking = (Default) The device continues working with a redundant (now active) Proxy until the next failure, after which it works with the next redundant Proxy. ▪ **[1]** Homing = The device always tries to work with the primary Proxy server (i.e., switches back to the primary Proxy whenever it's available). **Note:** To use this Proxy Redundancy mechanism, you need to enable the keep-alive with Proxy option, by setting the parameter EnableProxyKeepAlive to 1 or 2.
Web: Proxy IP List Refresh Time EMS: IP List Refresh Time CLI: proxy-ip-lst-rfrsh-time **[ProxyIPListRefreshTime]**	Defines the time interval (in seconds) between each Proxy IP list refresh. The range is 5 to 2,000,000. The default interval is 60.
Web: Enable Fallback to Routing Table EMS: Fallback Used CLI: fallback-to-routing [IsFallbackUsed]	Determines whether the device falls back to the Outbound IP Routing table for call routing when Proxy servers are unavailable. ▪ [0] Disable = (Default) Fallback is not used. ▪ [1] Enable = The Outbound IP Routing table is used when Proxy servers are unavailable. When the device falls back to the Outbound IP Routing table, it continues scanning for a Proxy. When the device locates an active Proxy, it switches from internal routing back to Proxy routing. **Note:** To enable the redundant Proxies mechanism, set the

Parameter	Description
	parameter EnableProxyKeepAlive to 1 or 2.
Web/EMS: Prefer Routing Table CLI: prefer-routing-table [PreferRouteTable]	Determines whether the device's internal routing table takes precedence over a Proxy for routing calls. ▪ [0] No = (Default) Only a Proxy server is used to route calls. ▪ [1] Yes = The device checks the routing rules in the Outbound IP Routing table for a match with the Tel-to-IP call. Only if a match is not found is a Proxy used.
Web/EMS: Always Use Proxy CLI: always-use-proxy **[AlwaysSendToProxy]**	Determines whether the device sends SIP messages and responses through a Proxy server. ▪ **[0]** Disable = (Default) Use standard SIP routing rules. ▪ **[1]** Enable = All SIP messages and responses are sent to the Proxy server. **Note:** This parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1).
Web: SIP ReRouting Mode EMS: SIP Re-Routing Mode CLI: sip-rerouting-mode [SIPReroutingMode]	Determines the routing mode after a call redirection (i.e., a 3xx SIP response is received) or transfer (i.e., a SIP REFER request is received). ▪ [0] Standard = (Default) INVITE messages that are generated as a result of Transfer or Redirect are sent directly to the URI, according to the Refer-To header in the REFER message, or Contact header in the 3xx response. ▪ [1] Proxy = Sends a new INVITE to the Proxy. Note: This option is applicable only if a Proxy server is used and the parameter AlwaysSendtoProxy is set to 0. ▪ [2] Routing Table = Uses the Routing table to locate the destination and then sends a new INVITE to this destination. **Notes:** ▪ This parameter is applicable only to the Gateway application. ▪ When this parameter is set to [1] and the INVITE sent to the Proxy fails, the device re-routes the call according to the Standard mode [0]. ▪ When this parameter is set to [2] and the INVITE fails, the device re-routes the call according to the Standard mode [0]. If DNS resolution fails, the device attempts to route the call to the Proxy. If routing to the Proxy also fails, the Redirect/Transfer request is rejected. ▪ When this parameter is set to [2], the XferPrefix parameter can be used to define different routing rules for redirect calls. ▪ This parameter is disregarded if the parameter AlwaysSendToProxy is set to 1.
Web/EMS: DNS Query Type CLI: dns-query **[DNSQueryType]**	Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to resolve Proxy and Registrar servers and to resolve all domain names that appear in the SIP Contact and Record-Route headers. ▪ **[0]** A-Record = (Default) No NAPTR or SRV queries are performed. ▪ **[1]** SRV = If the Proxy/Registrar IP address parameter, Contact/Record-Route headers, or IP address configured in the routing tables contain a domain name, an SRV query is

Parameter	Description
	performed. The device uses the first host name received from the SRV query. The device then performs a DNS A-record query for the host name to locate an IP address. ▪ **[2]** NAPTR = An NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type. **Notes:** ▪ If the Proxy/Registrar IP address parameter, the domain name in the Contact/Record-Route headers, or the IP address configured in the routing tables contain a domain name with a port definition, the device performs a regular DNS A-record query. ▪ If a specific Transport Type is configured, a NAPTR query is not performed. ▪ To enable NAPTR/SRV queries for Proxy servers only, use the global parameter ProxyDNSQueryType, or use the proxy Set table.
Web: Proxy DNS Query Type CLI: proxy-dns-query **[ProxyDNSQueryType]**	Global parameter that defines the DNS query record type for resolving the Proxy server's configured domain name (FQDN) into an IP address. ▪ **[0]** A-Record (default) = A-record DNS query. ▪ **[1]** SRV = If the Proxy IP address parameter contains a domain name without port definition (e.g., ProxyIP = domain.com), an SRV query is performed. The SRV query returns up to four Proxy host names and their weights. The device then performs DNS A-record queries for each Proxy host name (according to the received weights) to locate up to four Proxy IP addresses. Thus, if the first SRV query returns two domain names and the A-record queries return two IP addresses each, no additional searches are performed. ▪ **[2]** NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the Proxy IP address parameter contains a domain name with port definition (e.g., ProxyIP = domain.com:5080), the device performs a regular DNS A-record query. If a specific Transport Type is defined, a NAPTR query is not performed. **Notes:** ▪ This functionality can be configured per Proxy Set in the Proxy Set table (see "Configuring Proxy Sets" on page 282). ▪ When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
Web/EMS: Use Gateway Name for OPTIONS CLI: use-gw-name-for-opt [UseGatewayNameForOptions]	Determines whether the device uses its IP address or string name ("gateway name") in keep-alive SIP OPTIONS messages (host part of the Request-URI). The "gateway name" is configured using the SIPGatewayName parameter. The device

Parameter	Description
	uses the OPTIONS request as a keep-alive message with its primary and redundant SIP proxy servers (i.e., the EnableProxyKeepAlive parameter is set to 1). ▪ **[0]** No = (Default) Device's IP address is used in keep-alive OPTIONS messages. ▪ **[1]** Yes = Device's "gateway name" is used in keep-alive OPTIONS messages. ▪ **[2]** Server = Device's IP address is used in the From and To headers in keep-alive OPTIONS messages.
Web/EMS: User Name CLI: user-name-4-auth [UserName]	Defines the user name used for registration and Basic/Digest authentication with a Proxy/Registrar server. By default, no value is defined. **Notes:** ▪ This parameter is applicable only to the Gateway application. ▪ This parameter is applicable only if single device registration is used (i.e., the parameter AuthenticationMode is set to authentication per gateway). ▪ Instead of configuring this parameter, the Authentication table can be used (see Authentication on page 477).
Web/EMS: Password CLI: password-4-auth **[Password]**	Defines the password for Basic/Digest authentication with a Proxy/Registrar server. A single password is used for all device ports. The default is 'Default_Passwd'. **Note:** Instead of configuring this parameter, the Authentication table can be used (see Authentication on page 477).
Web/EMS: Cnonce CLI: cnonce-4-auth **[Cnonce]**	Defines the Cnonce string used by the SIP server and client to provide mutual authentication. The value is free format, i.e., 'Cnonce = 0a4f113b'. The default is 'Default_Cnonce'.
Web/EMS: Mutual Authentication Mode CLI: mutual-authentication **[MutualAuthenticationMode]**	Determines the device's mode of operation when Authentication and Key Agreement (AKA) Digest Authentication is used. ▪ **[0]** Optional = (Default) Incoming requests that don't include AKA authentication information are accepted. ▪ **[1]** Mandatory = Incoming requests that don't include AKA authentication information are rejected.
Web/EMS: Challenge Caching Mode CLI: challenge-caching [SIPChallengeCachingMode]	Determines the mode for Challenge Caching, which reduces the number of SIP messages transmitted through the network. The first request to the Proxy is sent without authorization. The Proxy sends a 401/407 response with a challenge. This response is saved for further uses. A new request is re-sent 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. ▪ [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.

Parameter	Description
	- [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. **Notes:** - This parameter is applicable only to the Gateway application. - Challenge Caching is used with all proxies and not only with the active one.
Proxy IP Table	
Web: Proxy IP Table EMS: Proxy IP CLI: configure voip > voip-network proxy-ip **[ProxyIP]**	This table parameter defines the Proxy Set table with Proxy Set IDs, each with up to 10 Proxy server IP addresses (or FQDN). The format of the ini file table parameter is as follows: [ProxyIP] FORMAT ProxyIp_Index = ProxyIp_IpAddress, ProxyIp_TransportType, ProxyIp_ProxySetId; [\ProxyIP] For a description of this table, see "Configuring Proxy Sets" on page 282. To configure the Proxy Set attributes (such as Proxy Load Balancing) in the ini file, use the ProxySet parameter.
Proxy Set Table	
Web: Proxy Set Table EMS: Proxy Set CLI: configure voip > voip-network proxy-set **[ProxySet]**	This table parameter defines the Proxy Set ID table. This includes, for example, Proxy keep-alive and load balancing and redundancy mechanisms. The format of the ini file table parameter is as follows: [ProxySet] FORMAT ProxySet_Index = ProxySet_ProxyName, ProxySet_EnableProxyKeepAlive, ProxySet_ProxyKeepAliveTime, ProxySet_ProxyLoadBalancingMethod, ProxySet_IsProxyHotSwap, ProxySet_SRD, ProxySet_ClassificationInput, ProxySet_ProxyRedundancyMode, ProxySet_DNSResolveMethod, ProxySet_KeepAliveFailureResp; [\ProxySet] For a description of this table, see "Configuring Proxy Sets" on page 282. For configuring the IP addresses per Proxy Set in the ini file, use the ProxyIP parameter.
Registrar Parameters	
Web: Enable Registration EMS: Is Register Needed CLI: enable-registration [IsRegisterNeeded]	Enables the device to register to a Proxy/Registrar server. - [0] Disable = (Default) The device doesn't register to Proxy/Registrar server. - [1] Enable = The device registers to Proxy/Registrar server when the device is powered up and at every user-defined interval (configured by the parameter RegistrationTime).

Parameter	Description
	Notes: - This parameter is applicable only to the Gateway application. - The device sends a REGISTER request for each channel or for the entire device (according to the AuthenticationMode parameter).
Web/EMS: Registrar Name CLI: registrar-name [RegistrarName]	Defines the Registrar domain name. If specified, the name is used as the Request-URI in REGISTER messages. If it isn't specified (default), the Registrar IP address, or Proxy name or IP address is used instead. The valid range is up to 100 characters. **Note:** This parameter is applicable only to the Gateway application.
Web: Registrar IP Address EMS: Registrar IP CLI: ip-addrr-rgstrr [RegistrarIP]	Defines the IP address (or FQDN) and port number (optional) of the Registrar server. The IP address is in dotted-decimal notation, e.g., 201.10.8.1:<5080>. **Notes:** - This parameter is applicable only to the Gateway application. - If not specified, the REGISTER request is sent to the primary Proxy server. - When a port number is specified, DNS NAPTR/SRV queries aren't performed, even if the parameter DNSQueryType is set to 1 or 2. - If the parameter RegistrarIP is set to an FQDN and is resolved to multiple addresses, the device also provides real-time switching (hotswap mode) between different Registrar IP addresses (the parameter IsProxyHotSwap is set to 1). If the first Registrar doesn't respond to the REGISTER message, the same REGISTER message is sent immediately to the next Proxy. To allow this mechanism, the parameter EnableProxyKeepAlive must be set to 0. - When a specific transport type is defined using the parameter RegistrarTransportType, a DNS NAPTR query is not performed even if the parameter DNSQueryType is set to 2.
Web/EMS: Registrar Transport Type CLI: registrar-transport [RegistrarTransportType]	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the Registrar. - [-1] Not Configured (default) - [0] UDP - [1] TCP - [2] TLS **Notes:** - This parameter is applicable only to the Gateway application. - When set to 'Not Configured', the value of the parameter SIPTransportType is used.
Web/EMS: Registration Time CLI: registration-time **[RegistrationTime]**	Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. This parameter also defines the time interval between Keep-Alive

Parameter	Description
	messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER). Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider. The valid range is 10 to 2,000,000. The default is 180.
Web: Re-registration Timing [%] EMS: Time Divider CLI: re-registration-timing **[RegistrationTimeDivider]**	Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server. The valid range is 50 to 100. The default is 50. For example: If this parameter is set to 70% and the Registration Expires time is 3600, the device re-sends its registration request after 3600 x 70% (i.e., 2520 sec). **Note**: This parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0.
Web/EMS: Registration Retry Time CLI: registration-retry-time **[RegistrationRetryTime]**	Defines the time interval (in seconds) after which a registration request is re-sent if registration fails with a 4xx response or if there is no response from the Proxy/Registrar server. The default is 30 seconds. The range is 10 to 3600.
Web: Registration Time Threshold EMS: Time Threshold CLI: registration-time-thres **[RegistrationTimeThreshold]**	Defines a threshold (in seconds) for re-registration timing. If this parameter is greater than 0, but lower than the computed re-registration timing (according to the parameter RegistrationTimeDivider), the re-registration timing is set to the following: timing set by the Registration server in the SIP Expires header minus the value of the parameter RegistrationTimeThreshold. The valid range is 0 to 2,000,000. The default is 0.
Web: Re-register On INVITE Failure EMS: Register On Invite Failure CLI: reg-on-invite-fail [RegisterOnInviteFailure]	Enables immediate re-registration if no response is received for an INVITE request sent by the device. ▪ [0] Disable (default) ▪ [1] Enable = The device immediately expires its re-registration timer and commences re-registration to the same Proxy upon any of the following scenarios: ✓ The response to an INVITE request is 407 (Proxy Authentication Required) without an authentication header included. ✓ The remote SIP UA abandons a call before the device has received any provisional response (indicative of an outbound proxy server failure). ✓ The remote SIP UA abandons a call and the only provisional response the device has received for the call is 100 Trying (indicative of a home proxy server failure, i.e., the failure of a proxy in the route after the outbound proxy). ✓ The device terminates a call due to the expiration of RFC 3261 Timer B or due to the receipt of a 408 (Request Timeout) response and the device has not received any provisional response for the call (indicative of an outbound proxy server failure). ✓ The device terminates a call due to the receipt of a 408 (Request Timeout) response and the only provisional response the device has received for the call is the 100

Parameter	Description
	Trying provisional response (indicative of a home proxy server failure). **Note:** This parameter is applicable only to the Gateway application.
Web: ReRegister On Connection Failure EMS: Re Register On Connection Failure CLI: reg-on-conn-failure **[ReRegisterOnConnectionFailure]**	Enables the device to perform SIP re-registration upon TCP/TLS connection failure. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web: Gateway Registration Name EMS: Name CLI: gw-registration-name [GWRegistrationName]	Defines the user name that is used in the From and To headers in SIP REGISTER messages. If no value is specified (default) for this parameter, the UserName parameter is used instead. **Notes:** ▪ This parameter is applicable only to the Gateway application. ▪ This parameter is applicable only for single registration per device (i.e., AuthenticationMode is set to 1). When the device registers each channel separately (i.e., AuthenticationMode is set to 0), the user name is set to the channel's phone number.
Web/EMS: Registration Mode CLI: authentication-mode [AuthenticationMode]	Determines the device's registration and authentication method. ▪ [0] Per Endpoint = Registration and authentication is performed separately for each endpoint/B-channel. This is typically used for FXS interfaces, where each endpoint registers (and authenticates) separately with its user name and password. ▪ [1] Per Gateway = (Default) Single registration and authentication for the entire device. This is typically used for FXO interfaces and digital modules. ▪ [3] Per FXS = Registration and authentication for FXS endpoints. **Note:** This parameter is applicable only to the Gateway application.
Web: Set Out-Of-Service On Registration Failure EMS: Set OOS On Registration Fail CLI: set-oos-on-reg-failure [OOSOnRegistrationFail]	Enables setting the endpoint, trunk, or entire device (i.e., all endpoints) to out-of-service if registration fails. ▪ [0] Disable (default) ▪ [1] Enable If the registration is per endpoint (i.e., AuthenticationMode is set to 0) or per Account (see Configuring Hunt Group Settings on page 367) and a specific endpoint/Account registration fails (SIP 4xx or no response), then that endpoint is set to out-of-service until a success response is received in a subsequent registration request. When the registration is per the entire device (i.e., AuthenticationMode is set to 1) and registration fails, all endpoints are set to out-of-service. If all the Accounts of a specific Trunk Group fail registration and if the Trunk Group comprises a complete trunk, then the entire trunk is set to out-of-service. **Notes:**

Parameter	Description
	- This parameter is applicable only to the Gateway application. - The out-of-service method is configured using the FXSOOSBehavior parameter.
CLI: expl-un-reg **[UnregistrationMode]**	Enables the device to perform explicit unregisters. - **[0]** Disable (default) - **[1]** Enable = The device sends an asterisk ("*") value in the SIP Contact header, instructing the Registrar server to remove all previous registration bindings. The device removes SIP User Agent (UA) registration bindings in a Registrar, according to RFC 3261. Registrations are soft state and expire unless refreshed, but they can also be explicitly removed. A client can attempt to influence the expiration interval selected by the Registrar. A UA requests the immediate removal of a binding by specifying an expiration interval of "0" for that contact address in a REGISTER request. UA's should support this mechanism so that bindings can be removed before their expiration interval has passed. Use of the "*" Contact header field value allows a registering UA to remove all bindings associated with an address-of-record (AOR) without knowing their precise values. **Note:** The REGISTER-specific Contact header field value of "*" applies to all registrations, but it can only be used if the Expires header field is present with a value of "0".
Web/EMS: Add Empty Authorization Header CLI: add-empty-author-hdr **[EmptyAuthorizationHeader]**	Enables the inclusion of the SIP Authorization header in initial registration (REGISTER) requests sent by the device. - **[0]** Disable (default) - **[1]** Enable The Authorization header carries the credentials of a user agent (UA) in a request to a server. The sent REGISTER message populates the Authorization header with the following parameters: - username - set to the value of the private user identity - realm - set to the domain name of the home network - uri - set to the SIP URI of the domain name of the home network - nonce - set to an empty value - response - set to an empty value For example: `Authorization: Digest username=alice_private@home1.net, realm=”home1.net”, nonce=””, response=”e56131d19580cd833064787ecc”` **Note:** This registration header is according to the IMS 3GPP TS24.229 and PKT-SP-24.220 specifications.
Web: Add initial Route Header CLI: add-init-rte-hdr **[InitialRouteHeader]**	Enables the inclusion of the SIP Route header in initial registration or re-registration (REGISTER) requests sent by the device. - **[0]** Disable (default) - **[1]** Enable

Parameter	Description

	When the device sends a REGISTER message, the Route header includes either the Proxy's FQDN, or IP address and port according to the configured Proxy Set, for example:
	`Route: <sip:10.10.10.10;lr;transport=udp>`
	or
	`Route: <sip: pcscf-gm.ims.rr.com;lr;transport=udp>`

EMS: Ping Pong Keep Alive **[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.
EMS: Ping Pong Keep Alive Time **[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.

Parameter	Description
Web: Max Generated Register Rate CLI: configure voip/sip-definition proxy-and-registration/max-gen-reg-rate **[MaxGeneratedRegistersRate]**	Defines the maximum number of user register requests that the device can send to a proxy (or registrar) server per second. This feature is useful in that it can be used to prevent an overload on the device's CPU caused by the sending of many (for example, 1,000) registration requests to a proxy server at any given time. The valid value is 30 to 300 register requests per second. The default is 100.

67.6.2 Network Application Parameters

The SIP network application parameters are described in the table below.

Table 67-34: SIP Network Application Parameters

Parameter	Description
Signaling Routing Domain Table	
Web: SRD Settings EMS: SRD Table CLI: configure voip > voip-network srd **[SRD]**	This table parameter configures the Signaling Routing Domains (SRD). The format of the ini file table parameter is as follows: [SRD] FORMAT SRD_Index = SRD_Name, SRD_MediaRealm, SRD_IntraSRDMediaAnchoring, SRD_BlockUnRegUsers, SRD_MaxNumOfRegUsers, SRD_EnableUnAuthenticatedRegistrations; [\SRD] For a detailed description of this table, see "Configuring SRDs" on page 264.
SIP Interface Table	
Web: SIP Interface Table EMS: SIP Interfaces Table CLI: configure voip > voip-network sip-interface **[SIPInterface]**	This table parameter configures SIP Interfaces. The SIP Interface represents a SIP signaling entity, comprising ports (UDP, TCP, and TLS) and associated with a specific IP interface and an SRD. The format of the ini file table parameter is as follows: [SIPInterface] FORMAT SIPInterface_Index = SIPInterface_InterfaceName, SIPInterface_NetworkInterface, SIPInterface_ApplicationType, SIPInterface_UDPPort, SIPInterface_TCPPort, SIPInterface_TLSPort, SIPInterface_SRD, SIPInterface_MessagePolicy, SIPInterface_TLSMutualAuthentication, SIPInterface_TCPKeepaliveEnable, SIPInterface_ClassificationFailureResponseType, SIPInterface_PreClassificationManSet; [\SIPInterface] For a detailed description of this table, see "Configuring SIP Interfaces" on page 267.
[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.

Parameter	Description
	▪ 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.
[TCPKeepAliveRetry]	Defines the number of unacknowledged keep-alive probes to send before considering the connection down. The valid value is 1 to 100. The default is 5. **Note:** TCP keepalive is enabled per SIP Interface in the SIP Interface table.
NAT Translation Table	
Web: NAT Translation Table CLI: configure voip > voip-network NATTranslation **[NATTranslation]**	This table parameter defines NAT rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses. This allows, for example, the separation of VoIP traffic between different ISTP's, and topology hiding (of internal IP addresses to the “public” network). Each IP interface (configured in the Interface table - InterfaceTable parameter) can be associated with a NAT rule in this table, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range). The format of the ini file table parameter is as follows: [NATTranslation] FORMAT NATTranslation_Index = NATTranslation_SourceIPInterfaceName, NATTranslation_TargetIPAddress, NATTranslation_SourceStartPort, NATTranslation_SourceEndPort, NATTranslation_TargetStartPort, NATTranslation_TargetEndPort; [\NATTranslation] For a detailed description of this table, see "Configuring NAT Translation per IP Interface" on page 151.
Media Realm Table	
Web: Media Realm Table EMS: Media Realm CLI: configure voip > voip-network realm **[CpMediaRealm]**	This table parameter defines Media Realms. The Media Realm table allows you to divide a Media-type interface (defined in the Interface table) into several realms, where each realm is specified by a UDP port range. The format of the ini file table parameter is as follows: [CpMediaRealm] FORMAT CpMediaRealm_Index = CpMediaRealm_MediaRealmName, CpMediaRealm_IPv4IF, CpMediaRealm_IPv6IF, CpMediaRealm_PortRangeStart, CpMediaRealm_MediaSessionLeg, CpMediaRealm_PortRangeEnd, CpMediaRealm_IsDefault, CpMediaRealm_QoeProfile, CpMediaRealm_BWProfile; [\CpMediaRealm] For a detailed description of this table, see "Configuring Media Realms" on page 259.
Remote Media Subnet Table	

Parameter	Description
Web: Remote Media Subnet EMS: Remote Media Subnet CLI: configure voip > voip-network realm remotemediasubnet **[SubRealm]**	This table parameter defines Remote Media Subnets. The format of the ini file table parameter is as follows: [RemoteMediaSubnet] FORMAT RemoteMediaSubnet_Index = RemoteMediaSubnet_Realm, RemoteMediaSubnet_RemoteMediaSubnetIndex, RemoteMediaSubnet_RemoteMediaSubnetName, RemoteMediaSubnet_PrefixLength, RemoteMediaSubnet_AddressFamily, RemoteMediaSubnet_DstIPAddress, RemoteMediaSubnet_QOEProfileName, RemoteMediaSubnet_BWProfileName; [\RemoteMediaSubnet] For a detailed description of this table, see "Configuring Remote Media Subnets" on page 262.

67.7 General SIP Parameters

The general SIP parameters are described in the table below.

Table 67-35: General SIP Parameters

Parameter	Description
Web: Send reject on overload CLI: configure voip/sip-definition advanced-settings/reject-on-ovrld **[SendRejectOnOverload]**	Disables the sending of SIP 503 (Service Unavailable) responses upon receipt of new SIP dialog-initiating requests when the device's CPU is overloaded and thus, unable to accept and process new SIP messages. ▪ **[0]** Disable = No SIP 503 response is sent when CPU overloaded. ▪ **[1]** Enable (default) = SIP 503 response is sent when CPU overloaded. ▪ **Note:** Even if this parameter is disabled (i.e., 503 is not sent), the device still discards the new SIP dialog-initiating requests when the CPU is overloaded.
Web: SIP 408 Response upon non-INVITE CLI: enbl-non-inv-408 **[EnableNonInvite408Reply]**	Enables the device to send SIP 408 responses (Request Timeout) upon receipt of non-INVITE transactions. Disabling this response complies with RFC 4320/4321. By default, and in certain circumstances such as a timeout expiry, the device sends a SIP 408 Request Timeout in response to non-INVITE requests (e.g., REGISTER). ▪ **[0]** Disable = SIP 408 response is not sent upon receipt of non-INVITE messages (to comply with RFC 4320). ▪ **[1]** Enable = (Default) SIP 408 response is sent upon receipt of non-INVITE messages, if necessary.
Web: SIP Remote Reset CLI: sip-remote-reset **[EnableSIPRemoteReset]**	Enables a specific device action upon the receipt of a SIP NOTIFY request, where the action depends on the value received in the Event header. ▪ **[0]** Disable (default) ▪ **[1]** Enable The action depends on the Event header value: ▪ 'check-sync;reboot=false': triggers the regular Automatic Update feature (if Automatic Update has been enabled on the device)

Parameter	Description
	▪ 'check-sync;reboot=true': triggers a device reset ▪ 'cwmp-connect': triggers connection with TR-069 **Note:** The Event header value is proprietary to AudioCodes.
Web/EMS: Max SIP Message Length [KB] **[MaxSIPMessageLength]**	Defines the maximum size (in Kbytes) for each SIP message that can be sent over the network. The device rejects messages exceeding this user-defined size. The valid value range is 1 to 50. The default is 50.
[SIPForceRport]	Determines whether the device sends SIP responses to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the SIP Via header. ▪ **[0]** = (Default) Disabled. The device sends the SIP response to the UDP port defined in the Via header. If the Via header contains the 'rport' parameter, the response is sent to the UDP port from where the SIP request is received. ▪ **[1]** = Enabled. SIP responses are sent to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the Via header.
Web: Reject Cancel after Connect CLI: reject-cancel-after-connect [**RejectCancelAfterConnect]**	Determines whether the device accepts or rejects a SIP CANCEL request received after the receipt of a 200 OK, during an established call. ▪ **[0]** = (Default) Accepts the CANCEL, by responding with a 200 OK and terminating the call session. ▪ **[1]** = Rejects the CANCEL, by responding with a SIP 481 Call/Transaction Does Not Exist, and maintaining the call session.
Web: Verify Received RequestURI CLI: verify-rcvd-requri **[VerifyReceevedRequestUri]**	Enables the device to reject SIP requests (such as ACK, BYE, or re-INVITE) whose user part in the Request-URI is different from the user part received in the Contact header of the last sent SIP request. ▪ **[0]** Disable = (Default) Even if the user is different, the device accepts the SIP request. ▪ **[1]** Enable = If the user is different, the device rejects the SIP request (BYE is responded with 481; re-INVITE is responded with 404; ACK is ignored).
Web: Max Number of Active Calls EMS: Maximum Concurrent Calls CLI: max-nb-of--act-calls **[MaxActiveCalls]**	Defines the maximum number of simultaneous active calls supported by the device. If the maximum number of calls is reached, new calls are not established. The valid range is 1 to the maximum number of supported channels. The default value is the maximum available channels (i.e., no restriction on the maximum number of calls).
Web: Number of Calls Limit **[IpProfile_CallLimit,]**	Defines the maximum number of concurrent calls per IP Profile (see "Configuring IP Profiles" on page 314).
Web: QoS statistics in SIP Release Call **[QoSStatistics]**	Enables the device to include call quality of service (QoS) statistics in SIP BYE and SIP 200 OK response to BYE, using the proprietary SIP header X-RTP-Stat. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable The X-RTP-Stat header provides the following statistics: ▪ Number of received and sent voice packets ▪ Number of received and sent voice octets ▪ Received packet loss, jitter (in ms), and latency (in ms)

Parameter	Description

	The X-RTP-Stat header contains the following fields:
	- PS=<voice packets sent>
	- OS=<voice octets sent>
	- PR=<voice packets received>
	- OR=<voice octets received>
	- PL=<receive packet loss>
	- JI=<jitter in ms>
	- LA=<latency in ms>
	Below is an example of the X-RTP-Stat header in a SIP BYE message:
	`BYE sip:302@10.33.4.125 SIP/2.0`
	`Via: SIP/2.0/UDP 10.33.4.126;branch=z9hG4bKac2127550866`
	`Max-Forwards: 70`
	`From: <sip:401@10.33.4.126;user=phone>;tag=1c2113553324`
	`To: <sip:302@company.com>;tag=1c991751121`
	`Call-ID: 991750671245200001912@10.33.4.125`
	`CSeq: 1 BYE`
	`X-RTP-Stat: PS=207;OS=49680;;PR=314;OR=50240;PL=0;JI=600;LA=40;`
	`Supported: em,timer,replaces,path,resource-priority`
	`Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE`
	`User-Agent: Sip-Gateway-/v.6.80A.227.005`
	`Reason: Q.850 ;cause=16 ;text="local"`
	`Content-Length: 0`

Web/EMS: PRACK Mode CLI: prack-mode **[PrackMode]**	Determines the PRACK (Provisional Acknowledgment) mechanism mode for SIP 1xx reliable responses. - **[0]** Disable - **[1]** Supported (default) - **[2]** Required **Notes:** - The Supported and Required headers contain the '100rel' tag. - The device sends PRACK messages if 180/183 responses are received with '100rel' in the Supported or Required headers.
Web/EMS: Enable Early Media CLI: early-media **[EnableEarlyMedia]**	Global parameter that enables the Early Media feature for sending media (e.g., ringing) before the call is established. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEarlyMedia) or Tel Profiles. For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314 or in the Tel Profile table, see Configuring Tel Profiles on page 311. **Notes:** - If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.

Parameter	Description
	- This parameter is applicable only to the Gateway application.
Web/EMS: Enable Early 183 CLI: early-183 [EnableEarly183]	Global parameter that enables the device to send SIP 183 responses with SDP to the IP upon receipt of INVITE messages. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEarly183). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 314. **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.
[IgnoreAlertAfterEarlyMedia]	Determines the device's interworking of Alerting messages from PRI to SIP. - [0] = Disabled (default) - [1] = Enabled When enabled, if the device sends a 183 response with an SDP (due to a received ISDN Progress or Proceeding with PI messages) and an Alerting message is then received from the Tel side (with or without Progress Indicator), the device does not send an additional 18x response, and the voice channel remains open. However, if the device did not send a 183 with an SDP and it receives an Alert without PI, the device sends a 180 (without SDP). If it receives an Alert with PI it sends a 183with an SDP. When disabled, the device sends additional 18x responses as a result of receiving Alerting and Progress messages, regardless of whether or not a 18x response was already sent. **Note:** This parameter is applicable only if the EnableEarlyMedia parameter is set to 1 (i.e., enabled).
Web: 183 Message Behavior EMS: SIP 183 Behaviour CLI: 183-msg-behavior [SIP183Behaviour]	Digital: Defines the ISDN message that is sent when the 183 Session Progress message is received for IP-to-Tel calls. Analog: Defines the response of the device upon receipt of a SIP 183 response. - [0] Progress = (Default) Digital: The device sends a Progress message. Analog: A 183 response (without SDP) does not cause the device to play a ringback tone. - [1] Alert = Digital: The device sends an Alerting message (upon receipt of a 183 response) instead of an ISDN Progress message. Analog: 183 response is handled by the device as if a 180 Ringing response is received, and the device plays a ringback tone. **Note:** The parameter is applicable only to the Gateway application.
[ReleaseIP2ISDNCallOnProgressWithCause]	Typically, if an Q.931 Progress message with a Cause is received from the PSTN for an outgoing IP-to-ISDN call and the EnableEarlyMedia parameter is set to 1 (i.e., the Early Media feature is enabled), the device interworks the Progress to 183 + SDP to enable the originating party to hear the PSTN announcement about the call failure. Conversely, if EnableEarlyMedia is set to 0, the device disconnects the call by sending a SIP 4xx response to the originating party. However, if the ReleaseIP2ISDNCallOnProgressWithCause parameter is set to 1, then the device sends a SIP 4xx response even if the EnableEarlyMedia parameter is set to 1. - [0] = (Default) If a Progress with Cause message is received from the PSTN for an outgoing IP-to-ISDN call, the device does not disconnect the call by sending a SIP 4xx response to the originating party.

Parameter	Description
	- [1] = The device sends a SIP 4xx response when the EnableEarlyMedia parameter is set to 0. - [2] = The device always sends a SIP 4xx response, even if he EnableEarlyMedia parameter is set to 1.
Web: Session-Expires Time EMS: Sip Session Expires CLI: session-expires-time **[SIPSessionExpires]**	Defines the numerical value sent in the Session-Expires header in the first INVITE request or response (if the call is answered). The valid range is 1 to 86,400 sec. The default is 0 (i.e., the Session-Expires header is disabled).
Web: Minimum Session-Expires EMS: Minimal Session Refresh Value CLI: min-session-expires **[MinSE]**	Defines the time (in seconds) that is used in the Min-SE header. This header defines the minimum time that the user agent refreshes the session. The valid range is 10 to 100,000. The default is 90.
Web/EMS: Session Expires Disconnect Time CLI: session-exp-disconnect-time **[SessionExpiresDisconnectTime]**	Defines a session expiry timeout. The device disconnects the session (sends a SIP BYE) if the refresher did not send a refresh request before one-third (1/3) of the session expires time, or before the time configured by this parameter (the minimum of the two). The valid range is 0 to 32 (in seconds). The default is 32.
Web/EMS: Session Expires Method CLI: session-exp-method **[SessionExpiresMethod]**	Determines the SIP method used for session-timer updates. - **[0]** Re-INVITE = (Default) Uses re-INVITE messages for session-timer updates. - **[1]** UPDATE = Uses UPDATE messages. **Notes:** - The device can receive session-timer refreshes using both methods. - The UPDATE message used for session-timer is excluded from the SDP body.
[RemoveToTagInFailureResponse]	Determines whether the device removes the 'to' header tag from final SIP failure responses to INVITE transactions. - **[0]** = (Default) Do not remove tag. - **[1]** = Remove tag.
[EnableRTCPAttribute]	Enables the use of the 'rtcp' attribute in the outgoing SDP. - **[0]** = Disable (default) - **[1]** = Enable **Note:** This parameter is applicable only to the Gateway application.
EMS: Options User Part **[OPTIONSUserPart]**	Defines the user part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the endpoint number (analog) or configuration parameter 'Username' value (digital) 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.
CLI: configure voip/gw digitalgw digital-gw-parameters/trunk-status-reporting [TrunkStatusReportingMode	Enables the device to not respond to received SIP OPTIONS messages from, and/or send keep-alive messages to, a proxy server associated with Trunk Group ID 1 if all its member trunks are down. - [0] Disable (default)

Parameter	Description
]	▪ [1] Don't reply OPTIONS = The device does not respond to SIP OPTIONS received from the proxy associated with Trunk Group 1 when all its trunks are down. ▪ [2] Don't send Keep-Alive = The device does not send keep-alive messages to the proxy associated with Trunk Group 1 when all its trunks are down. ▪ [3] Don't Reply and Send = Both options [1] and [2] are applied. **Notes:** ▪ When this parameter is set to not respond to SIP OPTIONS received from the proxy, it is applicable only if the OPTIONS message does not include a user part in the Request-URI. ▪ The proxy server is determined by the Proxy Set that is associated with the Serving IP Group defined for the Trunk Group in the Trunk Group Settings table.
Web: TDM Over IP Minimum Calls For Trunk Activation EMS: TDM Over IP Min Calls For Trunk Activation [TDMOverIPMinCallsForTrunkActivation]	Defines the minimal number of SIP dialogs that must be established when using TDM Tunneling, for the specific trunk to be considered active. When using TDM Tunneling, if calls from this defined number of B-channels pertaining to a specific Trunk fail (i.e., SIP dialogs are not correctly set up), an AIS alarm is sent on this trunk toward the PSTN and all current calls are dropped. The originator gateway continues the INVITE attempts. When this number of calls succeed (i.e., SIP dialogs are correctly set up), the AIS alarm is cleared. The valid range is 0 to 31. The default is 0 (i.e., don't send AIS alarms). **Note:** TDM Tunneling is applicable only to E1/T1 interfaces (not BRI).
[TDMoIPInitiateInviteTime]	Defines the time (in msec) between the first INVITE issued within the same trunk when implementing the TDM tunneling application. The valid value range is 500 to 1000. The default is 500. **Note:** TDM Tunneling is applicable only to E1/T1 interfaces (not BRI).
[TDMoIPInviteRetryTime]	Defines the time (in msec) between call release and a new INVITE when implementing the TDM tunneling application. The valid value range is 10,000 to 20,000. The default is 10,000. **Note:** TDM Tunneling is applicable only to E1/T1 interfaces (not BRI).
Web: Fax Signaling Method EMS: Fax Used CLI: fax-sig-method **[IsFaxUsed]**	Global parameter that defines the SIP signaling method for establishing and transmitting a fax session when the device detects a fax. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IsFaxUsed). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **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.

Parameter	Description
	▪ **[1]** = Enable. The device assumes that the G.711 coder received in the INVITE SDP offer is a VBD coder. For example, if the device is configured with G.729 and G.711 VBD coders and it receives an INVITE with an SDP offer containing G.729 and "regular" G.711 coders, it sends an SDP answer containing G.729 and G.711 VBD coders, allowing a subsequent bypass (passthrough) session if fax/modem signals are detected during the call. **Note:** This parameter is applicable only if G.711 VBD coder(s) with regular G.711 payload types 0 or 8 are configured for the device (using the CodersGroup parameter).
CLI: fax-vbd-behvr **[FaxVBDBehavior]**	Determines the device's fax transport behavior when G.711 VBD coder is negotiated at call start. ▪ **[0]** = (Default) If the device is configured with a VBD coder (see the CodersGroup parameter) and is negotiated OK at call start, then both fax and modem signals are sent over RTP using the bypass payload type (and no mid-call VBD or T.38 Re-INVITEs occur). ▪ **[1]** = If the IsFaxUsed parameter is set to 1, the channel opens with the FaxTransportMode parameter set to 1 (relay). This is required to detect mid-call fax tones and to send T.38 Re-INVITE messages upon fax detection. If the remote party supports T.38, the fax is relayed over T.38. **Notes:** ▪ If VBD coder negotiation fails at call start and if the IsFaxUsed parameter is set to 1 (or 3), then the channel opens with the FaxTransportMode parameter set to 1 (relay) to allow future detection of fax tones and sending of T.38 Re-INVITES. In such a scenario, the FaxVBDBehavior parameter has no effect. ▪ This feature can be used only if the remote party supports T.38 fax relay; otherwise, the fax fails.
[NoAudioPayloadType]	Defines the payload type of the outgoing SDP offer. The valid value range is 96 to 127 (dynamic payload type). The default is 0 (i.e. NoAudio is not supported). For example, if set to 120, the following is added to the INVITE SDP: `a=rtpmap:120 NoAudio/8000\r\n` **Note:** For incoming SDP offers, NoAudio is always supported.
Web: SIP Transport Type EMS: Transport Type CLI: app-sip-transport-type **[SIPTransportType]**	Determines the default transport layer for outgoing SIP calls initiated by the device. ▪ **[0]** UDP (default) ▪ **[1]** TCP ▪ **[2]** TLS (SIPS) **Notes:** ▪ It's recommended to use TLS for communication with a SIP Proxy and not for direct device-to-device communication. ▪ For received calls (i.e., incoming), the device accepts all these protocols. ▪ The value of this parameter is also used by the SAS application as the default transport layer for outgoing SIP calls.
Web: SIP UDP Local Port EMS: Local SIP Port CLI: sip-udp-local-port	Defines the local UDP port for SIP messages. The valid range is 1 to 65534. The default is 5060.

Parameter	Description
[LocalSIPPort]	
Web: SIP TCP Local Port EMS: TCP Local SIP Port CLI: sip-tcp-local-port **[TCPLocalSIPPort]**	Defines the local TCP port for SIP messages. The valid range is 1 to 65535. The default is 5060.
Web: SIP TLS Local Port EMS: TLS Local SIP Port CLI: sip-tls-local-port **[TLSLocalSIPPort]**	Defines the local TLS port for SIP messages. The valid range is 1 to 65535. The default is 5061. **Note:** The value of this parameter must be different from the value of the parameter TCPLocalSIPPort.
Web: Display Default SIP Port CLI: display-default-sip-port **[DisplayDefaultSIPPort]**	Enables the device to add the default SIP port 5060 (UDP/TCP) or 5061 (TLS) to outgoing messages that are received without a port. This condition also applies to manipulated messages where the resulting message has no port number. The device adds the default port number to the following SIP headers: Request-Uri, To, From, P-Asserted-Identity, P-Preferred-Identity, and P-Called-Party-ID. If the message is received with a port number other than the default, for example, 5070, the port number is not changed. An example of a SIP From header with the default port is shown below: `From: <sip:+4000@10.8.4.105:5060;user=phone>;tag=f25419a96a;epid=009FAB8F3E` ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: Enable SIPS CLI: enable-sips **[EnableSIPS]**	Enables secured SIP (SIPS URI) connections over multiple hops. ▪ **[0]** Disable (default) ▪ **[1]** Enable When the SIPTransportType parameter is set to 2 (i.e., TLS) and the parameter EnableSIPS is disabled, TLS is used for the next network hop only. When the parameter SIPTransportType is set to 2 or 1 (i.e., TCP or TLS) and EnableSIPS is enabled, TLS is used through the entire connection (over multiple hops). **Note:** If this parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails.
Web/EMS: Enable TCP Connection Reuse CLI: tcp-conn-reuse **[EnableTCPConnectionReuse]**	Enables the reuse of the same TCP connection for all calls to the same destination. ▪ **[0]** Disable = Uses a separate TCP connection for each call. ▪ **[1]** Enable = (Default) Uses the same TCP connection for all calls. **Notes:** ▪ ▪ For the SAS application, this feature is configured using the SASConnectionReuse parameter.
Web: Fake TCP alias CLI: fake-tcp-alias **[FakeTCPalias]**	Enables the re-use of the same TCP/TLS connection for sessions with the same user, even if the "alias" parameter is not present in the SIP Via header of the first INVITE. ▪ **[0]** Disable = (Default) TCP/TLS connection reuse is done only if the "alias" parameter is present in the Via header of the first INVITE. ▪ **[1]** Enable **Note:** To enable TCP/TLS connection re-use, set the

Parameter	Description
	EnableTCPConnectionReuse parameter to 1.
Web/EMS: Reliable Connection Persistent Mode CLI: reliable-conn-persistent **[ReliableConnectionPersistentMode]**	Enables setting of all TCP/TLS connections as persistent and therefore, not released. ▪ **[0]** = (Default) Disable. All TCP connections (except those that are set to a proxy IP) are released if not used by any SIP dialog\transaction. ▪ **[1]** = Enable - TCP connections to all destinations are persistent and not released unless the device reaches 70% of its maximum TCP resources. While trying to send a SIP message connection, reuse policy determines whether live connections to the specific destination are re-used. Persistent TCP connection ensures less network traffic due to fewer setting up and tearing down of TCP connections and reduced latency on subsequent requests due to avoidance of initial TCP handshake. For TLS, persistent connection may reduce the number of costly TLS handshakes to establish security associations, in addition to the initial TCP connection set up. **Note:** If the destination is a Proxy server, the TCP/TLS connection is persistent regardless of the settings of this parameter.
Web/EMS: TCP Timeout CLI: tcp-timeout **[SIPTCPTimeout]**	Defines the Timer B (INVITE transaction timeout timer) and Timer F (non-INVITE transaction timeout timer), as defined in RFC 3261, when the SIP transport type is TCP. The valid range is 0 to 40 sec. The default is 64 * SipT1Rtx parameter value. For example, if SipT1Rtx is set to 500 msec, then the default of SIPTCPTimeout is 32 sec.
Web: SIP Destination Port EMS: Destination Port CLI: sip-dst-port **[SIPDestinationPort]**	Defines the SIP destination port for sending initial SIP requests. The valid range is 1 to 65534. The default port is 5060. **Note:** SIP responses are sent to the port specified in the Via header.
Web: Use user=phone in SIP URL EMS: Is User Phone CLI: user=phone-in-url **[IsUserPhone]**	Determines whether the 'user=phone' string is added to the SIP URI and SIP To header. ▪ **[0]** No = 'user=phone' string is not added. ▪ **[1]** Yes = (Default) 'user=phone' string is part of the SIP URI and SIP To header.
Web: Use user=phone in From Header EMS: Is User Phone In From CLI: phone-in-from-hdr **[IsUserPhoneInFrom]**	Determines whether the 'user=phone' string is added to the From and Contact SIP headers. ▪ **[0]** No = (Default) Doesn't add 'user=phone' string. ▪ **[1]** Yes = 'user=phone' string is part of the From and Contact headers.
Web: Use Tel URI for Asserted Identity CLI: uri-for-assert-id **[UseTelURIForAssertedID]**	Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers. ▪ **[0]** Disable = (Default) 'sip:' ▪ **[1]** Enable = 'tel:'
Web: Tel to IP No Answer Timeout EMS: IP Alert Timeout CLI: tel2ip-no-ans-timeout [IPAlertTimeout]	Defines the time (in seconds) that the device waits for a 200 OK response from the called party (IP side) after sending an INVITE message, for Tel-to-IP calls. If the timer expires, the call is released. The valid range is 0 to 3600. The default is 180.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td>Web: Enable Remote Party ID
EMS: Enable RPI Header
CLI: remote-party-id
[EnableRPIheader]</td><td>Enables Remote-Party-Identity headers for calling and called numbers for Tel-to-IP calls.<ul><li>[0] Disable (default).</li><li>[1] Enable = Remote-Party-Identity headers are generated in SIP INVITE messages for both called and calling numbers.</li></ul></td></tr>
<tr><td>Web: Enable History-Info Header
EMS: Enable History Info
CLI: hist-info-hdr
[EnableHistoryInfo]</td><td>Enables usage of the History-Info header.<ul><li>[0] Disable (default)</li><li>[1] Enable</li></ul>User Agent Client (UAC) Behavior:<ul><li>Initial request: The History-Info header is equal to the Request-URI. If a PSTN Redirect number is received, it is added as an additional History-Info header with an appropriate reason.</li><li>Upon receiving the final failure response, the device copies the History-Info as is, adds the reason of the failure response to the last entry, and concatenates a new destination to it (if an additional request is sent). The order of the reasons is as follows:<ol type="a"><li>Q.850 Reason</li><li>SIP Reason</li><li>SIP Response code</li></ol></li><li>Upon receiving the final response (success or failure), the device searches for a Redirect reason in the History-Info (i.e., 3xx/4xx SIP reason). If found, it is passed to ISDN according to the following table:</li></ul>
<table>
<tr><th>SIP Reason Code</th><th>ISDN Redirecting Reason</th></tr>
<tr><td>302 - Moved Temporarily</td><td>Call Forward Universal (CFU)</td></tr>
<tr><td>408 - Request Timeout</td><td rowspan="3">Call Forward No Answer (CFNA)</td></tr>
<tr><td>480 - Temporarily Unavailable</td></tr>
<tr><td>487 - Request Terminated</td></tr>
<tr><td>486 - Busy Here</td><td rowspan="2">Call Forward Busy (CFB)</td></tr>
<tr><td>600 - Busy Everywhere</td></tr>
</table>
<ul><li>If history reason is a Q.850 reason, it is translated to the SIP reason (according to the SIP-ISDN tables) and then to ISDN Redirect reason according to the table above.</li></ul>User Agent Server (UAS) Behavior:<ul><li>The History-Info header is sent only in the final response.</li><li>Upon receiving a request with History-Info, the UAS checks the policy in the request. If a 'session', 'header', or 'history' policy tag is found, the (final) response is sent without History-Info; otherwise, it is copied from the request.</li></ul></td></tr>
<tr><td>Web: Use Tgrp Information
EMS: Use SIP Tgrp
CLI: use-tgrp-inf
[UseSIPTgrp]</td><td>Determines whether the SIP 'tgrp' parameter is used. This SIP parameter specifies the Trunk Group to which the call belongs (according to RFC 4904). For example, the SIP message below indicates that the call belongs to Trunk Group ID 1:<pre>INVITE sip::+16305550100;tgrp=1;trunk-context=example.com@10.1.0.3;user=phone SIP/2.0</pre><ul><li>[0] Disable = (Default) The 'tgrp' parameter isn't used.</li><li>[1] Send Only = The Trunk Group number or name (configured in the Hunt Group Settings) is added to the 'tgrp' parameter value in the Contact header of outgoing SIP messages. If a Trunk Group</li></ul></td></tr>
</table>

Parameter	Description
	number / name is not associated with the call, the 'tgrp' parameter isn't included. If a 'tgrp' value is specified in incoming messages, it is ignored. ▪ [2] Send and Receive = The functionality of outgoing SIP messages is identical to the functionality described for option [1]. In addition, for incoming SIP INVITEs, if the Request-URI includes a 'tgrp' parameter, the device routes the call according to that value (if possible). The Contact header in the outgoing SIP INVITE (Tel-to-IP call) contains "tgrp=<source trunk group ID>;trunk-context=<gateway IP address>”. The <source trunk group ID> is the Trunk Group ID where incoming calls from Tel is received. For IP-Tel calls, the SIP 200 OK device's response contains “tgrp=<destination trunk group ID>;trunk-context=<gateway IP address>”. The <destination trunk group ID> is the Trunk Group ID used for outgoing Tel calls. The <gateway IP address> in “trunk-context” can be configured using the SIPGatewayName parameter. ▪ [3] Hotline = Interworks the hotline "Off Hook Indicator" parameter between SIP and ISDN: ✓ For IP-to-ISDN calls: - The device interworks the SIP tgrp=hotline parameter (received in INVITE) to ISDN Setup with the Off Hook Indicator IE of “Voice”, and “Speech” Bearer Capability IE. Note that the Off Hook Indicator IE is described in UCR 2008 specifications. - The device interworks the SIP tgrp=hotline-ccdata parameter (received in INVITE) to ISDN Setup with an Off Hook Indicator IE of “Data”, and with “Unrestricted 64k” Bearer Capability IE. The following is an example of the INVITE with tgrp=hotline-ccdata: INVITE sip:1234567;tgrp=hotline-ccdata;trunk-context=dsn.mil@example.com ✓ For ISDN-to-IP calls: - The device interworks ISDN Setup with an Off Hook Indicator of “Voice” to SIP INVITE with “tgrp=hotline;trunk-context=dsn.mil” in the Contact header. - The device interworks ISDN Setup with an Off Hook indicator of “Data” to SIP INVITE with “tgrp=hotline-ccdata;trunk-context=dsn.mil” in the Contact header. - If ISDN Setup does not contain an Off Hook Indicator IE and the Bearer Capability IE contains “Unrestricted 64k”, the outgoing INVITE includes “tgrp=ccdata;trunk-context=dsn.mil”. If the Bearer Capability IE contains “Speech”, the INVITE in this case does not contain tgrp and trunk-context parameters. ▪ [4] Hotline Extended = Interworks the ISDN Setup message’s hotline "OffHook Indicator" Information Element (IE) to SIP INVITE’s Request-URI and Contact headers. (Note: For IP-to-ISDN calls, the device handles the call as described in option [3].) ✓ The device interworks ISDN Setup with an Off Hook Indicator of “Voice” to SIP INVITE Request-URI and Contact header with “tgrp=hotline;trunk-context=dsn.mil”. ✓ The device interworks ISDN Setup with an Off Hook indicator of “Data” to SIP INVITE Request-URI and Contact header with “tgrp=hotline-ccdata;trunk-context=dsn.mil”. ✓ If ISDN Setup does not contain an Off Hook Indicator IE and the Bearer Capability IE contains “Unrestricted 64k”, the outgoing INVITE Request-URI and Contact header includes

Parameter	Description
	"tgrp=ccdata;trunk-context=dsn.mil". If the Bearer Capability IE contains "Speech", the INVITE in this case does not contain tgrp and trunk-context parameters. **Note:** IP-to-Tel configuration (using the PSTNPrefix parameter) overrides the 'tgrp' parameter in incoming INVITE messages.
Web/EMS: TGRP Routing Precedence CLI: tgrp-routing-prec [TGRProutingPrecedence]	Determines the precedence method for routing IP-to-Tel calls - according to the Inbound IP Routing table or according to the SIP 'tgrp' parameter. ▪ [0] = (Default) IP-to-Tel routing is determined by the Inbound IP Routing table (PSTNPrefix parameter). If a matching rule is not found in this table, the device uses the Trunk Group parameters for routing the call. ▪ [1] = The device first places precedence on the 'tgrp' parameter for IP-to-Tel routing. If the received INVITE Request-URI does not contain the 'tgrp' parameter or if the Trunk Group number is not defined, then the Inbound IP Routing table is used for routing the call. Below is an example of an INVITE Request-URI with the 'tgrp' parameter, indicating that the IP call should be routed to Trunk Group 7: `INVITE sip:200;tgrp=7;trunk-context=example.com@10.33.2.68;user=phone SIP/2.0` **Notes:** ▪ For enabling routing based on the 'tgrp' parameter, the UseSIPTgrp parameter must be set to 2. ▪ For IP-to-Tel routing based on the 'dtg' parameter (instead of the 'tgrp' parameter), use the parameter UseBroadsoftDTG.
CLI: use-dtg [UseBroadsoftDTG]	Determines whether the device uses the 'dtg' parameter for routing IP-to-Tel calls to a specific Trunk Group. ▪ [0] Disable (default) ▪ [1] Enable When this parameter is enabled, if the Request-URI in the received SIP INVITE includes the 'dtg' parameter, the device routes the call to the Trunk Group according to its value. This parameter is used instead of the 'tgrp/trunk-context' parameters. The 'dtg' parameter appears in the INVITE Request-URI (and in the To header). For example, the received SIP message below routes the call to Trunk Group ID 56: `INVITE sip:123456@192.168.1.2;dtg=56;user=phone SIP/2.0` **Note:** If the Trunk Group is not found based on the 'dtg' parameter, the Inbound IP Routing table is used instead for routing the call to the appropriate Trunk Group.
Web/EMS: Enable GRUU CLI: enable-gruu **[EnableGRUU]**	Determines whether the Globally Routable User Agent URIs (GRUU) mechanism is used, according to RFC 5627. This is used for obtaining a GRUU from a registrar and for communicating a GRUU to a peer within a dialog. ▪ **[0]** Disable (default) ▪ **[1]** Enable A GRUU is a SIP URI that routes to an instance-specific UA and can be reachable from anywhere. There are a number of contexts in which it is desirable to have an identifier that addresses a single UA (using GRUU) rather than the group of UA's indicated by an Address of

Parameter	Description

	Record (AOR). For example, in call transfer where user A is talking to user B, and user A wants to transfer the call to user C. User A sends a REFER to user C:
	`REFER sip:C@domain.com SIP/2.0`
	`From: sip:A@domain.com;tag=99asd`
	`To: sip:C@domain.com`
	`Refer-To: (URI that identifies B's UA)`
	The Refer-To header needs to contain a URI that user C can use to place a call to user B. This call needs to route to the specific UA instance that user B is using to talk to user A. User B should provide user A with a URI that has to be usable by anyone. It needs to be a GRUU.
	▪ Obtaining a GRUU: The mechanism for obtaining a GRUU is through registrations. A UA can obtain a GRUU by generating a REGISTER request containing a Supported header field with the value “gruu”. The UA includes a “+sip.instance” Contact header parameter of each contact for which the GRUU is desired. This Contact parameter contains a globally unique ID that identifies the UA instance. The global unique ID is created from one of the following:
	✓ If the REGISTER is per the device’s client (endpoint), it is the MAC address concatenated with the phone number of the client.
	✓ If the REGISTER is per device, it is the MAC address only.
	✓ When using TP, “User Info” can be used for registering per endpoint. Thus, each endpoint can get a unique id – its phone number. The globally unique ID in TP is the MAC address concatenated with the phone number of the endpoint.
	If the remote server doesn’t support GRUU, it ignores the parameters of the GRUU. Otherwise, if the remote side also supports GRUU, the REGISTER responses contain the “gruu” parameter in each Contact header. This parameter contains a SIP or SIPS URI that represents a GRUU corresponding to the UA instance that registered the contact. The server provides the same GRUU for the same AOR and instance-id when sending REGISTER again after registration expiration. RFC 5627 specifies that the remote target is a GRUU target if its’ Contact URL has the "gr" parameter with or without a value.
	▪ Using GRUU: The UA can place the GRUU in any header field that can contain a URI. It must use the GRUU in the following messages: INVITE request, its 2xx response, SUBSCRIBE request, its 2xx response, NOTIFY request, REFER request and its 2xx response.

EMS: Is CISCO Sce Mode **[IsCiscoSCEMode]**	Determines whether a Cisco gateway exists at the remote side. ▪ **[0]** = (Default) No Cisco gateway exists at the remote side. ▪ **[1]** = A Cisco gateway exists at the remote side. When a Cisco gateway exists at the remote side, the device must set the value of the 'annexb' parameter of the fmtp attribute in the SDP to 'no'. This logic is used if the parameter EnableSilenceCompression is set to 2 (enable without adaptation). In this case, Silence Suppression is used on the channel but not declared in the SDP. **Note:** The IsCiscoSCEMode parameter is applicable only when the selected coder is G.729.

Parameter	Description

Web: User-Agent Information
EMS: User Agent Display Info
CLI: user-agent-info
[UserAgentDisplayInfo]
	Defines the string that is used in the SIP User-Agent and Server response headers. When configured, the string <UserAgentDisplayInfo value>/software version' is used, for example:
	`User-Agent: myproduct/v.6.80A.227.005`
	If not configured, the default string, <AudioCodes product-name>/software version' is used, for example:
	`User-Agent: Audiocodes-Sip-Gateway-Mediant 800B MSBR/v.6.80A.227.005`
	The maximum string length is 50 characters.
	Note: The software version number and preceding forward slash (/) cannot be modified. Therefore, it is recommended not to include a forward slash in the parameter's value (to avoid two forward slashes in the SIP header, which may cause problems).

Web/EMS: SDP Session Owner CLI: sdp-session-owner **[SIPSDPSessionOwner]**	Defines the value of the Owner line ('o' field) in outgoing SDP messages. The valid range is a string of up to 39 characters. The default is "AudiocodesGW". For example: `o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126`
CLI: sdp-ver-nego **[EnableSDPVersionNegotiation]**	Enables the device to ignore new SDP re-offers (from the media negotiation perspective) in certain scenarios (such as session expires). According to RFC 3264, once an SDP session is established, a new SDP offer is considered a new offer only when the SDP origin value is incremented. In scenarios such as session expires, SDP negotiation is irrelevant and thus, the origin field is not changed. Even though some SIP devices don't follow this behavior and don't increment the origin value even in scenarios where they want to re-negotiate, the device can assume that the remote party operates according to RFC 3264, and in cases where the origin field is not incremented, the device does not re-negotiate SDP capabilities. ▪ **[0]** Disable = (Default) The device negotiates any new SDP re-offer, regardless of the origin field. ▪ **[1]** Enable = The device negotiates only an SDP re-offer with an incremented origin field.
Web/EMS: Subject CLI: usr-def-subject **[SIPSubject]**	Defines the Subject header value in outgoing INVITE messages. If not specified, the Subject header isn't included (default). The maximum length is up to 50 characters.
[CoderPriorityNegotiation]	Defines the priority for coder negotiation in the incoming SDP offer, between the device's or remote UA's coder list. ▪ [0] = (Default) Coder negotiation is given higher priority to the remote UA's list of supported coders. ▪ [1] = Coder negotiation is given higher priority to the device's (local) supported coders list. ▪ Note: This parameter is applicable only to the Gateway/IP-to-IP application.
Web: Send All Coders on Retrieve CLI: send-all-cdrs-on-rtrv [SendAllCodersOnRetrieve]	Enables coder re-negotiation in the sent re-INVITE for retrieving an on-hold call. ▪ [0] Disable = (Default) Sends only the initially chosen coder when the call was first established and then put on-hold. ▪ [1] Enable = Includes all supported coders in the SDP of the re-

Parameter	Description
	INVITE sent to the call made un-hold (retrieved). The used coder is therefore, re-negotiated. This parameter is useful in the following call scenario example: 1 Party A calls party B and coder G.711 is chosen. 2 Party B is put on-hold while Party A blind transfers Party B to Party C. 3 Party C answers and Party B is made un-hold. However, as Party C supports only G.729 coder, re-negotiation of the supported coder is required. **Note:** This parameter is applicable only to the Gateway application.
Web: Multiple Packetization Time Format EMS: Multi Ptime Format CLI: mult-ptime-format **[MultiPtimeFormat]**	Determines whether the 'mptime' attribute is included in the outgoing SDP. ▪ **[0]** None = (Default) Disabled. ▪ **[1]** PacketCable = Includes the 'mptime' attribute in the outgoing SDP - PacketCable-defined format. The mptime' attribute enables the device to define a separate packetization period for each negotiated coder in the SDP. The 'mptime' attribute is only included if this parameter is enabled even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value.
EMS: Enable P Time **[EnablePtime]**	Determines whether the 'ptime' attribute is included in the SDP. ▪ **[0]** = Remove the 'ptime' attribute from SDP. ▪ **[1]** = (Default) Include the 'ptime' attribute in SDP.
Web/EMS: 3xx Behavior CLI: 3xx-behavior **[3xxBehavior]**	Determines the device's behavior regarding call identifiers when a 3xx response is received for an outgoing INVITE request. The device can either use the same call identifiers (Call-ID, To, and From tags) or change them in the new initiated INVITE. ▪ **[0]** Forward = (Default) Use different call identifiers for a redirected INVITE message. ▪ **[1]** Redirect = Use the same call identifiers.
Web/EMS: Enable P-Charging Vector CLI: p-charging-vector **[EnablePChargingVector]**	Enables the inclusion of the P-Charging-Vector header to all outgoing INVITE messages. ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: Retry-After Time CLI: retry-aftr-time **[RetryAfterTime]**	Defines the time (in seconds) used in the Retry-After header when a 503 (Service Unavailable) response is generated by the device. The time range is 0 to 3,600. The default is 0.
Web/EMS: Fake Retry After [sec] CLI: fake-retry-after **[FakeRetryAfter]**	Determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by this parameter. ▪ **[0]** Disable (default) ▪ Any positive value (in seconds) for defining the period When enabled, this feature allows the device to operate with Proxy servers that do not include the Retry-After SIP header in SIP 503 (Service Unavailable) responses to indicate an unavailable service. The Retry-After header is used with the 503 (Service Unavailable)

Parameter	Description
	response to indicate how long the service is expected to be unavailable to the requesting SIP client. The device maintains a list of available proxies, by using the Keep-Alive mechanism. The device checks the availability of proxies by sending SIP OPTIONS every keep-alive timeout to all proxies. If the device receives a SIP 503 response to an INVITE, it also marks that the proxy is out of service for the defined "Retry-After" period.
Web/EMS: Enable P-Associated-URI Header CLI: p-associated-uri-hdr **[EnablePAssociatedURIHeader]**	Determines the device usage of the P-Associated-URI header. This header can be received in 200 OK responses to REGISTER requests. When enabled, the first URI in the P-Associated-URI header is used in subsequent requests as the From/P-Asserted-Identity headers value. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).
Web/EMS: Source Number Preference CLI: src-nb-preference **[SourceNumberPreference]**	Determines from which SIP header the source (calling) number is obtained in incoming INVITE messages. ▪ If not configured or if any string other than "From" or "Pai2" is configured, the calling number is obtained from a specific header using the following logic: a. P-Preferred-Identity header. b. If the above header is not present, then the first P-Asserted-Identity header is used. c. If the above header is not present, then the Remote-Party-ID header is used. d. If the above header is not present, then the From header is used. ▪ "**From**" = The calling number is obtained from the From header. ▪ "**Pai2**" = The calling number is obtained using the following logic: a. If a P-Preferred-Identity header is present, the number is obtained from it. b. If no P-Preferred-Identity header is present and two P-Asserted-Identity headers are present, the number is obtained from the second P-Asserted-Identity header. c. If only one P-Asserted-Identity header is present, the calling number is obtained from it. **Notes:** ▪ The "From" and "Pai2" values are not case-sensitive. ▪ Once a URL is selected, all the calling party parameters are set from this header. If P-Asserted-Identity is selected and the Privacy header is set to 'id', the calling number is assumed restricted.
CLI: src-hdr-4-called-nb [SelectSourceHeaderForCalledNumber]	Determines the SIP header used for obtaining the called number (destination) for IP-to-Tel calls. ▪ [0] Request-URI header = (Default) Obtains the destination number from the user part of the Request-URI. ▪ [1] To header = Obtains the destination number from the user part of the To header. ▪ [2] P-Called-Party-ID header = Obtains the destination number from the P-Called-Party-ID header.
Web/EMS: Enable Reason Header CLI: reason-header	Enables the usage of the SIP Reason header. ▪ **[0]** Disable

Parameter	Description
[EnableReasonHeader]	▪ **[1]** Enable (default)
Web/EMS: Gateway Name CLI: gw-name **[SIPGatewayName]**	Defines a name for the device (e.g., device123.com). This name is used as the host part of the SIP URI in the From header. If not specified, the device's IP address is used instead (default). **Notes:** ▪ Ensure that the parameter value is the one with which the Proxy has been configured with to identify the device. ▪ This parameter can also be configured for an IP Group (in the IP Group table).
[ZeroSDPHandling]	Determines the device's response to an incoming SDP that includes an IP address of 0.0.0.0 in the SDP's Connection Information field (i.e., "c=IN IP4 0.0.0.0"). ▪ **[0]** = (Default) Sets the IP address of the outgoing SDP's c= field to 0.0.0.0. ▪ **[1]** = Sets the IP address of the outgoing SDP c= field to the IP address of the device. If the incoming SDP doesn't contain the "a=inactive" line, the returned SDP contains the "a=recvonly" line.
Web/EMS: Enable Delayed Offer CLI: delayed-offer **[EnableDelayedOffer]**	Determines whether the device sends the initial INVITE message with or without an SDP. Sending the first INVITE without SDP is typically done by clients for obtaining the far-end's full list of capabilities before sending their own offer. (An alternative method for obtaining the list of supported capabilities is by using SIP OPTIONS, which is not supported by every SIP agent.) ▪ **[0]** Disable = (Default) The device sends the initial INVITE message with an SDP. ▪ **[1]** Enable = The device sends the initial INVITE message without an SDP.
[DisableCryptoLifeTimeInSDP]	Enables the device to send "a=crypto" lines without the lifetime parameter in the SDP. For example, if the SDP contains "a=crypto:12 AES_CM_128_HMAC_SHA1_80 inline:hhQe10yZRcRcpIFPkH5xYY9R1de37ogh9G1MpvNp|2^31", it removes the lifetime parameter "2^31". ▪ **[0]** Disable (default) ▪ **[1]** Enable
Web/EMS: Enable Contact Restriction CLI: contact-restriction **[EnableContactRestriction]**	Determines whether the device sets the Contact header of outgoing INVITE requests to 'anonymous' for restricted calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable
CLI: anonymous-mode [AnonymousMode]	Determines whether the device's IP address is used as the URI host part instead of "anonymous.invalid" in the INVITE's From header for Tel-to-IP calls. ▪ [0] = (Default) If the device receives a call from the Tel with blocked caller ID, it sends an INVITE with From: "anonymous"<anonymous@anonymous.invalid> ▪ [1] = The device's IP address is used as the URI host part instead of "anonymous.invalid". This parameter may be useful, for example, for service providers who identify their SIP Trunking customers by their source phone number or IP address, reflected in the From header of the SIP INVITE. Therefore,

Parameter	Description
	even customers blocking their Caller ID can be identified by the service provider. Typically, if the device receives a call with blocked Caller ID from the PSTN side (e.g., Trunk connected to a PBX), it sends an INVITE to the IP with a From header as follows: From: “anonymous” <anonymous@anonymous.invalid>. This is in accordance with RFC 3325. However, when this parameter is set to 1, the device replaces the "anonymous.invalid" with its IP address.
EMS: P Asserted User Name CLI: p-assrtd-usr-name [PAssertedUserName]	Defines a 'representative number' (up to 50 characters) that is used as the user part of the Request-URI in the P-Asserted-Identity header of an outgoing INVITE for Tel-to-IP calls. The default is null.
EMS: Use URL In Refer To Header **[UseAORInReferToHeader]**	Defines the source for the SIP URI set in the Refer-To header of outgoing REFER messages. ▪ **[0]** = (Default) Use SIP URI from Contact header of the initial call. ▪ **[1]** = Use SIP URI from To/From header of the initial call.
Web: Enable User-Information Usage CLI: user-inf-usage **[EnableUserInfoUsage]**	Enables the usage of the User Information, which is loaded to the device in the User Information auxiliary file. For a description on User Information, see "Loading Auxiliary Files" on page 599. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
[HandleReasonHeader]	Determines whether the device uses the value of the incoming SIP Reason header for Release Reason mapping. ▪ **[0]** = Disregard Reason header in incoming SIP messages. ▪ **[1]** = (Default) Use the Reason header value for Release Reason mapping.
[EnableSilenceSuppInSDP]	Determines the device's behavior upon receipt of SIP Re-INVITE messages that include the SDP's 'silencesupp:off' attribute. ▪ **[0]** = (Default) Disregard the 'silecesupp' attribute. ▪ **[1]** = Handle incoming Re-INVITE messages that include the 'silencesupp:off' attribute in the SDP as a request to switch to the Voice-Band-Data (VBD) mode. In addition, the device includes the attribute 'a=silencesupp:off' in its SDP offer. **Note:** This parameter is applicable only if the G.711 coder is used.
[EnableRport]	Enables the usage of the 'rport' parameter in the Via header. ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled The device adds an 'rport' parameter to the Via header of each outgoing SIP message. The first Proxy that receives this message sets the 'rport' value of the response to the actual port from where the request was received. This method is used, for example, to enable the device to identify its port mapping outside a NAT. If the Via header doesn't include the 'rport' parameter, the destination port of the response is obtained from the host part of the Via header. If the Via header includes the 'rport' parameter without a port value, the destination port of the response is the source port of the incoming request. If the Via header includes 'rport' with a port value (e.g., rport=1001), the destination port of the response is the port indicated in the 'rport' parmeter.

Parameter	Description
Web: Enable X-Channel Header EMS: X Channel Header CLI: x-channel-header **[XChannelHeader]**	Determines whether the SIP X-Channel header is added to SIP messages for providing information on the physical Trunk/B-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 Trunk number, B-channel, and the device's IP address. For example, 'x-channel: DS/DS1-5/8;IP=192.168.13.1', where: ✓ 'DS/DS-1' is a constant string ✓ '5' is the Trunk number ✓ '8' is the B-channel ✓ 'IP=192.168.13.1' is the device's IP address
Web/EMS: Progress Indicator to IP CLI: prog-ind-2ip [ProgressIndicator2IP]	Global parameter that defines the progress indicator (PI) sent to the IP. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_ProgressIndicator2IP) or Tel Profiles. For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 314 or in the Tel Profile table, see Configuring Tel Profiles on page 311. **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"). The re-INVITE is sent immediately upon receipt of the 200 OK (when the call is answered). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Note:** This parameter is applicable only if SRTP is used.
[NumberOfActiveDialogs]	Defines the maximum number of concurrent, outgoing SIP REGISTER dialogs. This parameter is used to control the registration rate. The valid range is 1 to 20. The default is 20. **Notes:** ▪ Once a 200 OK is received in response to a REGISTER message, the REGISTER message is not considered in this maximum count limit. ▪ This parameter applies only to outgoing REGISTER messages (i.e., incoming is unlimited).
EMS: Transparent Coder On Data Call [TransparentCoderOnDataCall]	▪ [0] = (Default) Only use coders from the coder list. ▪ [1] = Use Transparent coder for data calls (according to RFC 4040). The Transparent coder can be used on data calls. When the device receives a Setup message from the ISDN with 'TransferCapabilities = data', it can initiate a call using the coder 'Transparent' (even if the coder is not included in the coder list). The initiated INVITE includes the following SDP attribute: `a=rtpmap:97 CLEARMODE/8000` The default payload type is set according to the CodersGroup parameter. If the Transparent coder is not defined, the default is set to 56. The payload type is negotiated with the remote side, i.e., the selected payload type is according to the remote side selection. The receiving device must include the 'Transparent' coder in its coder list.

Parameter	Description
Web: IP to IP Application CLI: enable-ip2ip [EnableIP2IPApplication]	Enables the IP-to-IP Call Routing application. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Enable RFC 4117 Transcoding CLI: rfc4117-trnsc-enbl [EnableRFC4117Transcoding]	Enables transcoding of calls according to RFC 4117. ▪ [0] Disable (default) ▪ [1] Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For more information on transcoding, see Transcoding using Third-Party Call Control.
Web/EMS: Default Release Cause CLI: dflt-release-cse [DefaultReleaseCause]	Defines the default Release Cause (sent to IP) for IP-to-Tel calls when the device initiates a call release and an explicit matching cause for this release is not found. The default release cause is NO_ROUTE_TO_DESTINATION (3). Other common values include NO_CIRCUIT_AVAILABLE (34), DESTINATION_OUT_OF_ORDER (27), etc. **Notes:** ▪ The default release cause is described in the Q.931 notation and is translated to corresponding SIP 40x or 50x values (e.g., 3 to SIP 404, and 34 to SIP 503). ▪ Analog: For more information on mapping PSTN release causes to SIP responses, see Mapping PSTN Release Cause to SIP Response on page 395. ▪ When the Trunk is disconnected or is not synchronized, the internal cause is 27. This cause is mapped, by default, to SIP 502. ▪ For mapping SIP-to-Q.931 and Q.931-to-SIP release causes, see Configuring Release Cause Mapping on page 390. ▪ For a list of SIP responses-Q.931 release cause mapping, see Alternative Routing to Trunk upon Q.931 Call Release Cause Code on page 417.
Web: Enable Microsoft Extension CLI: microsoft-ext **[EnableMicrosoftExt]**	Enables the modification of the called and calling number for numbers received with Microsoft's proprietary "ext=xxx" parameter in the SIP INVITE URI user part. Microsoft Office Communications Server sometimes uses this proprietary parameter to indicate the extension number of the called or calling party. ▪ **[0]** Disable (default) ▪ **[1]** Enable For example, if a calling party makes a call to telephone number 622125519100 Ext. 104, the device receives the SIP INVITE (from Microsoft's application) with the URI user part as INVITE sip:622125519100;ext=104@10.1.1.10 (or INVITE tel:622125519100;ext=104). If the parameter EnableMicrosofExt is enabled, the device modifies the called number by adding an "e" as the prefix, removing the "ext=" parameter, and adding the extension number as the suffix (e.g., **e**622125519100**104**). Once modified, the device can then manipulate the number further, using the Number Manipulation tables to leave only the last 3 digits (for example) for sending to a PBX.
EMS: Use SIP URI For Diversion Header	Defines the URI format in the SIP Diversion header.

Parameter	Description
[UseSIPURIForDiversionHeader]	▪ **[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).
[EnableImmediateTrying]	Determines if and when the device sends a 100 Trying in response to an incoming INVITE request. ▪ [0] = 100 Trying response is sent upon receipt of a Proceeding message from the PSTN. ▪ [1] = (Default) 100 Trying response is sent immediately upon receipt of INVITE request.
[TransparentCoderPresentation]	Determines the format of the Transparent coder representation in the SDP. ▪ [0] = clearmode (default) ▪ [1] = X-CCD
[IgnoreRemoteSDPMKI]	Determines whether the device ignores the Master Key Identifier (MKI) if present in the SDP received from the remote side. ▪ **[0]** Disable (default) ▪ **[1]** Enable
[TrunkStatusReportingMode]	Determines whether the device responds to SIP OPTIONS if all the trunks pertaining to Trunk Group #1 are down. ▪ [0] Disable (default) ▪ [1] Enable = If all the trunks pertaining to Trunk Group #1 are down, the device does not respond to received SIP OPTIONS.
Web: Comfort Noise Generation Negotiation EMS: Comfort Noise Generation CLI: com-noise-gen-nego **[ComfortNoiseNegotiation]**	Enables negotiation and usage of Comfort Noise (CN) for Gateway calls. ▪ **[0]** Disable ▪ **[1]** Enable (default) The use of CN is indicated by including a payload type for CN on the media description line of the SDP. The device can use CN with a codec whose RTP time stamp clock rate is 8,000 Hz (G.711/G.726). The static payload type 13 is used. The use of CN is negotiated between sides. Therefore, if the remote side doesn't support CN, it is not used. Regardless of the device's settings, it always attempts to adapt to the remote SIP UA's request for CNG, as described below. To determine CNG support, the device uses the ComfortNoiseNegotiation parameter and the codec's SCE (silence suppression setting) using the CodersGroup parameter. If the ComfortNoiseNegotiation parameter is enabled, then the following occurs: ▪ If the device is the initiator, it sends a "CN" in the SDP only if the SCE of the codec is enabled. If the remote UA responds with a "CN" in the SDP, then CNG occurs; otherwise, CNG does not occur. ▪ If the device is the receiver and the remote SIP UA does not send a "CN" in the SDP, then no CNG occurs. If the remote side sends a "CN", the device attempts to be compatible with the remote side

Parameter	Description
	and even if the codec's SCE is disabled, CNG occurs. If the ComfortNoiseNegotiation parameter is disabled, then the device does not send "CN" in the SDP. However, if the codec's SCE is enabled, then CNG occurs. **Note:** This parameter is applicable only to the Gateway application.
CLI: sdp-ecan-frmt **[SDPEcanFormat]**	Defines the echo canceller format in the outgoing SDP. The 'ecan' attribute is used in the SDP to indicate the use of echo cancellation. ▪ **[0]** = (Default) The 'ecan' attribute appears on the 'a=gpmd' line. ▪ **[1]** = The 'ecan' attribute appears as a separate attribute. ▪ **[2]** = The 'ecan' attribute is not included in the SDP. ▪ **[3]** = The 'ecan' attribute and the 'vbd' parameter are not included in the SDP. **Note:** This parameter is applicable only when the IsFaxUsed parameter is set to 2, and for re-INVITE messages generated by the device as result of modem or fax tone detection.
Web/EMS: First Call Ringback Tone ID CLI: 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 this parameter). The valid range is -1 to 1,000. The default is -1 (i.e., play standard ringback tone). **Notes:** ▪ It is assumed that all ringback tones are defined in sequence in the CPT file. ▪ In case of an MLPP call, the device uses the value of this parameter plus 1 as the index of the ringback tone in the CPT file (e.g., if this value is set to 1, then the index is 2, i.e., 1 + 1).
Web: Reanswer Time EMS: Regret Time CLI: reanswer-time [RegretTime]	Analog: Defines the time interval from when the user hangs up the phone until the call is disconnected (FXS). This allows the user to hang up and then pick up the phone (before this timeout) to continue the call conversation. Thus, it's also referred to as regret time. Digital: Defines the time period the device waits for an MFC R2 Resume (Reanswer) signal once a Suspend (Clear back) signal is received from the PBX. If this timer expires, the call is released. Note that this is applicable only to the MFC-R2 CAS Brazil variant. The valid range is 0 to 255 (in seconds). The default is 0.
Web: Enable Reanswering Info CLI: reans-info-enbl [EnableReansweringINFO]	Enables the device to send a SIP INFO message with the On-Hook/Off-Hook parameter when the FXS phone goes on-hook during an ongoing call and then off-hook again, within the user-defined regret timeout (configured by the parameter RegretTime). Therefore, the device notifies the far-end that the call has been re-answered. ▪ [0] Disable (default) ▪ [1] Enable This parameter is typically implemented for incoming IP-to-Tel collect calls to the FXS port. If the FXS user does not wish to accept the collect call, the user disconnects the call by on-hooking the phone. The device notifies the softswitch (or Application server) of the unanswered collect call (on-hook) by sending a SIP INFO message. As a result, the softswitch disconnects the call (sends a BYE message to the device).

Parameter	Description
	If the call is a regular incoming call and the FXS user on-hooks the phone without intending to disconnect the call, the softswitch does not disconnect the call (during the regret time). The INFO message format is as follows: INFO sip:12345@10.50.228.164:5082 SIP/2.0 Via: SIP/2.0/UDP 127.0.0.1;branch=z9hG4bK_05_905924040-90579 From: <sip:+551137077803@ims.acme.com.br:5080;user=phone>;tag=008277765 To: <sip:notavailable@unknown.invalid>;tag=svw-0-1229428367 Call-ID: ConorCCR-0-LU-1229417827103300@dtas-stdn.fs5000group0-000.l CSeq: 1 INFO Contact: sip:10.20.7.70:5060 Content-Type: application/On-Hook (application/Off-Hook) Content-Length: 0 **Notes:** ▪ This parameter is applicable only if the parameter RegretTime is configured. ▪ This parameter is applicable only to FXS interfaces.
Web: PSTN Alert Timeout EMS: Trunk PSTN Alert Timeout CLI: pstn-alert-timeout [PSTNAlertTimeout]	Digital: Defines the Alert Timeout (in seconds) for calls sent to the PSTN. This timer is used between the time a Setup message is sent to the Tel side (IP-to-Tel call establishment) and a Connect message is received. If an Alerting message is received, the timer is restarted. If the timer expires before the call is answered, the device disconnects the call and sends a SIP 408 request timeout response to the SIP party that initiated the call. Analog: Defines the Alert Timeout (in seconds) for calls to the Tel side. This timer is used between the time a ring is generated (FXS) or a line is seized (FXO), until the call is connected. For example: If the FXS device receives an INVITE, it generates a ring to the phone and sends a SIP 180 Ringing response to the IP. If the phone is not answered within the time interval set by this parameter, the device cancels the call by sending a SIP 408 response. The valid value range is 1 to 600 (in seconds). The default is 180. **Note:** If per trunk configuration (using TrunkPSTNAlertTimeout) is set to other than default, the PSTNAlertTimeout parameter value is overridden.
Web/EMS: RTP Only Mode CLI: rtp-only-mode **[RTPOnlyMode]**	Enables the device to send and receive RTP packets to and from remote endpoints without the need to establish a SIP session. The remote IP address is determined according to the Outbound IP Routing table (Prefix parameter). The port is the same port as the local RTP port (configured by the BaseUDPPort parameter and the channel on which the call is received). ▪ **[0]** Disable (default) ▪ **[1]** Transmit & Receive = Send and receive RTP packets. ▪ **[2]** Transmit Only= Send RTP packets only. ▪ **[3]** Receive Only= Receive RTP packets only. **Notes:** ▪ To activate the RTP Only feature without using ISDN / CAS signaling, you must do the following:

Parameter	Description
	✓ Configure E1/T1 Transparent protocol type (set the ProtocoType parameter to 5 or 6). ✓ Enable the TDM-over-IP feature (set the EnableTDMoverIP parameter to 1). ▪ To configure the RTP Only mode per trunk, use the RTPOnlyModeForTrunk_x parameter. ▪ If per trunk configuration (using the RTPOnlyModeForTrunk_ID parameter) is set to a value other than the default, the RTPOnlyMode parameter value is ignored.
[RTPOnlyModeForTrunk_x]	Enables the RTP Only feature per trunk. The x in the parameter name denotes the trunk number, where 0 is Trunk 1. For a description of this parameter, see the RTPOnlyMode parameter. **Note:** For using the global parameter (i.e., setting the RTP Only feature for all trunks), set this parameter to -1 (default).
Web/EMS: Media IP Version Preference CLI: media-ip-ver-pref [MediaIPVersionPreference]	Global parameter that defines the preferred RTP media IP addressing version (IPv4 or IPv6) for outgoing SIP calls. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MediaIPVersionPreference). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 314.
Web/EMS: SIT Q850 Cause [SITQ850Cause]	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when a Special Information Tone (SIT) is detected on an IP-to-Tel call. The valid range is 0 to 127. The default is 34. **Note:** For mapping specific SIT tones, you can use the SITQ850CauseForNC, SITQ850CauseForIC, SITQ850CauseForVC, and SITQ850CauseForRO parameters.
Web/EMS: SIT Q850 Cause For NC [SITQ850CauseForNC]	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-NC (No Circuit Found Special Information Tone) is detected from the Tel side for IP-to-Tel calls. The valid range is 0 to 127. The default is 34. **Notes:** ▪ When not configured (i.e., default), the SITQ850Cause parameter is used. ▪ This parameter is applicable only to FXO interfaces.
Web/EMS: SIT Q850 Cause For IC [SITQ850CauseForIC]	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-IC (Operator Intercept Special Information Tone) is detected from the Tel for IP-to-Tel calls. The valid range is 0 to 127. The default is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
Web/EMS: SIT Q850 Cause For VC [SITQ850CauseForVC]	Defines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when SIT-VC (Vacant Circuit - non-registered number Special Information Tone) is detected from the Tel for IP-to-Tel calls. The valid range is 0 to 127. The default is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
Web/EMS: SIT Q850 Cause	Defines the Q.850 cause value specified in the SIP Reason header

Parameter	Description
For RO [SITQ850CauseForRO]	that is included in a 4xx response when SIT-RO (Reorder - System Busy Special Information Tone) is detected from the Tel for IP-to-Tel calls. The valid range is 0 to 127. The default is -1 (not configured). **Note:** When not configured (i.e., default), the SITQ850Cause parameter is used.
[GWInboundManipulationSet]	Selects the Manipulation Set ID for manipulating all inbound INVITE messages. The Manipulation Set is defined using the MessageManipulations parameter. By default, no manipulation is done (i.e. Manipulation Set ID is set to -1). **Note:** This parameter is applicable only to the Gateway application.
[GWOutboundManipulationSet]	Selects the Manipulation Set ID for manipulating all outbound INVITE messages. The Manipulation Set is defined using the MessageManipulations parameter. By default, no manipulation is done (i.e. Manipulation Set ID is set to -1). **Notes:** ▪ This parameter is used only if the Outbound Message Manipulation Set parameter of the destination IP Group is not set. ▪ This parameter is applicable only to the Gateway application.
Out-of-Service (Busy Out) Parameters	
Web/EMS: Enable Busy Out CLI: busy-out [EnableBusyOut]	Enables the Busy Out feature. ▪ [0] Disable (Default) ▪ [1] Enable When Busy Out is enabled and certain scenarios exist, the device does the following: ▪ Analog: The FXS port behaves according to the settings of the FXSOOSBehavior parameter such as plays a reorder tone when the phone is off-hooked, or changes the line polarity. ▪ Digital: All E1/T1 trunks are automatically taken out of service by taking down the D-Channel or by sending a Service Out message for T1 PRI trunks supporting these messages (NI-2, 4/5-ESS, DMS-100, and Meridian). These behaviors are done upon one of the following scenarios: ▪ The device is physically disconnected from the network (i.e., Ethernet cable is disconnected). ▪ The Ethernet cable is connected, but the device is unable to communicate with any host. For this scenario, the LAN Watchdog feature must be activated (i.e., set the EnableLANWatchDog parameter to 1). ▪ The device can't communicate with the proxy (according to the Proxy Keep-Alive mechanism) and no other alternative route exists to send the call. ▪ The IP Connectivity mechanism is enabled (using the AltRoutingTel2IPEnable parameter) and there is no connectivity to any destination IP address. **Notes:** ▪ Analog: The FXSOOSBehavior parameter determines the behavior of the FXS endpoints when a Busy Out or Graceful Lock occurs. ▪ Analog: FXO endpoints during Busy Out and Lock are inactive.

Parameter	Description
	▪ Analog: See the LifeLineType parameter for complementary optional behavior. ▪ Digital: The Busy Out behavior depends on the PSTN protocol type. ▪ Digital: The Busy-Out condition can also be applied to a specific Trunk Group. If there is no connectivity to the Serving IP Group of a specific Trunk Group (defined in the Hunt Group Settings table), all physical trunks pertaining to that Trunk Group are set to the Busy-Out condition. Each trunk uses the proper Out-Of-Service method according to the selected ISDN/CAS variant. ▪ Digital: To configure the method for setting digital trunks to Out-Of-Service, use the DigitalOOSBehavior parameter.
Web/EMS: Graceful Busy Out Timeout [sec] CLI: graceful-bsy-out-t-out [GracefulBusyOutTimeout]	Defines the timeout interval (in seconds) for Out-of-Service graceful shutdown mode for busy trunks (per trunk) if communication fails with a Proxy server (or Proxy Set). In such a scenario, the device rejects new calls from the PSTN (Serving Trunk Group), but maintains currently active calls for this user-defined timeout. Once this timeout elapses, the device terminates currently active calls and takes the trunk out of service (sending the PSTN busy-out signal). Trunks on which no calls are active are immediately taken out of service regardless of the timeout. The range is 0 to 3,600. The default is 0. **Note:** This parameter is applicable only to digital interfaces.
Web: Digital Out-Of-Service Behavior EMS: Digital OOS Behavior For Trunk Value CLI: dig-oos-behavior [DigitalOOSBehaviorForTrunk_x]	Determines the method for setting digital trunks to Out-of-Service state per trunk. ▪ [-1] Not Configured = (Default) Use the settings of the DigitalOOSBehavior parameter for per device. ▪ [0] Default = Uses default behavior for each trunk (see note below). ▪ [1] Service = Sends ISDN In or Out of Service (only for ISDN protocols that support Service message). ▪ [2] D-Channel = Takes D-Channel down or up (ISDN only). ▪ [3] Alarm = Sends or clears PSTN AIS Alarm (ISDN and CAS). ▪ [4] Block = Blocks trunk (CAS only). **Notes:** ▪ This parameter is applicable only to digital interfaces. ▪ This parameter is applicable only if the EnableBusyOut parameter is set to 1 (Enable). ▪ The default behavior (value 0) is as follows: ✓ ISDN: Use Service messages on supporting variants and use alarm on non-supporting variants. ✓ CAS: Use alarm. ▪ When updating this parameter value at run-time, you must stop the trunk and then restart it for the update to take effect. ▪ To determine the method for setting Out-Of-Service state for all trunks (i.e., per device), use the DigitalOOSBehavior parameter. ▪ The x in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
Web: Digital Out-Of-Service Behavior CLI: dig-oos-behavior [DigitalOOSBehavior]	Determines the method for setting digital trunks to Out-of-Service state. This configuration applies to all the device's trunks. For a description of this parameter's options, see the DigitalOOSBehaviorForTrunk_x parameter. **Notes:**

Parameter	Description
	- This parameter is applicable only to digital interfaces. - To configure the method for setting Out-of-Service state per trunk, use the DigitalOOSBehaviorForTrunk_x parameter. - To configure the timeout interval (in seconds) for Out-of-Service graceful shutdown mode for busy trunks if communication fails with a Proxy server (or Proxy Set), use the GracefulBusyOutTimeout parameter.
Web: Out-Of-Service Behavior EMS:FXS OOS Behavior CLI: oos-behavior [FXSOOSBehavior]	Determines the behavior of FXS endpoints when a Busy Out condition exists. - [0] None = Silence is heard when the FXS endpoint goes off-hook. - [1] Reorder Tone = (Default) The device plays a reorder tone to the connected phone / PBX. - [2] Polarity Reversal = The device reverses the polarity of the endpoint making it unusable (relevant, for example, for PBX DID lines). - [3] Reorder Tone + Polarity Reversal = Same as options [1] and [2]. - [4] Current Disconnect = The device disconnects the current to the FXS endpoint. **Notes:** - A device reset is required for this parameter to take effect when it is set to [2], [3], or [4]. - This parameter is applicable only to FXS interfaces.
Retransmission Parameters	
Web: SIP T1 Retransmission Timer [msec] EMS: T1 RTX CLI: t1-re-tx-time **[SipT1Rtx]**	Defines the time interval (in msec) between the first transmission of a SIP message and the first retransmission of the same message. The default is 500. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx. For INVITE requests, it is multiplied by two for each new retransmitted message. For all other SIP messages, it is multiplied by two until SipT2Rtx. For example, assuming SipT1Rtx = 500 and SipT2Rtx = 4000: - The first retransmission is sent after 500 msec. - The second retransmission is sent after 1000 (2*500) msec. - The third retransmission is sent after 2000 (2*1000) msec. - The fourth retransmission and subsequent retransmissions until SIPMaxRtx are sent after 4000 (2*2000) msec.
Web: SIP T2 Retransmission Timer [msec] EMS: T2 RTX CLI: t2-re-tx-time **[SipT2Rtx]**	Defines the maximum interval (in msec) between retransmissions of SIP messages (except for INVITE requests). The default is 4000. **Note:** The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx.
Web: SIP Maximum RTX EMS: Max RTX CLI: sip-max-rtx **[SIPMaxRtx]**	Defines the maximum number of UDP transmissions of SIP messages (first transmission plus retransmissions). The range is 1 to 30. The default is 7.

Parameter	Description
Web: Number of RTX Before Hot-Swap EMS: Proxy Hot Swap Rtx CLI: nb-of-rtx-b4-hot-swap **[HotSwapRtx]**	Defines the number of retransmitted INVITE/REGISTER messages before the call is routed (hot swap) to another Proxy/Registrar. The valid range is 1 to 30. The default is 3. **Note:** This parameter is also used for alternative routing. If a domain name in the Outbound IP Routing table or SBC IP-to-IP Routing table is resolved into two IP addresses, and if there is no response for HotSwapRtx retransmissions to the INVITE message that is sent to the first IP address, the device immediately initiates a call to the second IP address.
SIP Message Manipulations Table	
Web: Message Manipulations EMS: Message Manipulations CLI: configure voip > sbc manipulations message-manipulations [MessageManipulations]	This table parameter defines manipulation rules for SIP header messages. The format of the ini file table parameter is as follows: [MessageManipulations] FORMAT MessageManipulations_Index = MessageManipulations_ManSetID, MessageManipulations_MessageType, MessageManipulations_Condition, MessageManipulations_ActionSubject, MessageManipulations_ActionType, MessageManipulations_ActionValue, MessageManipulations_RowRole; [\MessageManipulations] For example, the below configuration changes the user part of the SIP From header to 200: MessageManipulations 1 = 0, Invite.Request, , Header.From.Url.User, 2, 200, 0; For a detailed description of this table, see Configuring SIP Message Manipulation on page 299.
Message Policy Table	
Web: Message Policy Table CLI: configure voip > sbc message-policy [MessagePolicy]	This table parameter configures SIP message policy rules for blocking (blacklist) unwanted incoming SIP messages or allowing (whitelist) receipt of desired messages. The format of the ini file table parameter is as follows: [MessagePolicy] FORMAT MessagePolicy_Index = MessagePolicy_Policy, MessagePolicy_MaxMessageLength, MessagePolicy_MaxHeaderLength, MessagePoliy_MaxBodyLength, MessagePolicy_MaxNumHeaders, MessagePolicy_MaxNumBodies, MessagePolicy_SendRejection, MessagePolicy_MethodListType, MessagePolicy_MethodList, MessagePolicy_BodyListType, MessagePolicy_BodyList; [/MessagePolicy] For a detailed description of this table, see Configuring SIP Message Policy Rules.

67.8 Coders and Profile Parameters

The profile parameters are described in the table below.

Table 67-36: Profile Parameters

Parameter	Description
Coders Table / Coder Groups Table	
Web: Coders Table/Coder Group Settings EMS: Coders Group CLI: configure voip > coders-and-profiles coders-group [CodersGroup0] [CodersGroup1] [CodersGroup2] [CodersGroup3] [CodersGroup4] [CodersGroup5] [CodersGroup6] [CodersGroup7] [CodersGroup8] [CodersGroup9]	This table parameter defines the device's coders. Each group can consist of up to 10 coders. The first Coder Group is the default coder list and the default Coder Group. The format of the ini file table parameter is as follows: [CodersGroup<0-9>] FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce; [\CodersGroup<0-9] For example, below are defined two Coder Groups (0 and 1): [CodersGroup0] FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime, CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce; CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0; CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0; CodersGroup0 2 = eg711Ulaw, 10, 0, 71, 0; [\CodersGroup0] [CodersGroup1] FORMAT CodersGroup1_Index = CodersGroup1_Name, CodersGroup1_pTime, CodersGroup1_rate, CodersGroup1_PayloadType, CodersGroup1_Sce; CodersGroup1 0 = Transparent, 20, 0, 56, 0; CodersGroup1 1 = g726, 20, 0, 23, 0; [\CodersGroup1] **Notes:** ▪ For a list of supported coders and a detailed description of this table, see Configuring Default Coders on page 307. ▪ For configuring Coder Groups, see "Configuring Coder Groups" on page 310. ▪ The coder name is case-sensitive.
IP Profile Table	
Web: IP Profile Settings EMS: Protocol Definition > IP Profile CLI: configure voip > coders-and-profiles ip-profile **[IPProfile]**	This table parameter configures the IP Profile table. Each IP Profile ID includes a set of parameters (which are typically configured separately using their individual "global" parameters). You can later assign these IP Profiles to outbound IP routing rules (Prefix parameter), inbound IP routing rules and IP Groups. The format of the ini file table parameter is as follows: [IPProfile] FORMAT IpProfile_Index = IpProfile_ProfileName, IpProfile_IpPreference, IpProfile_CodersGroupID, IpProfile_IsFaxUsed, IpProfile_JitterBufMinDelay, IpProfile_JitterBufOptFactor, IpProfile_IPDiffServ, IpProfile_SigIPDiffServ, IpProfile_SCE, IpProfile_RTPRedundancyDepth, IpProfile_RemoteBaseUDPPort, IpProfile_CNGmode, IpProfile_VxxTransportType, IpProfile_NSEMode,

Parameter	Description
	IpProfile_IsDTMFUsed, IpProfile_PlayRBTone2IP, IpProfile_EnableEarlyMedia, IpProfile_ProgressIndicator2IP, IpProfile_EnableEchoCanceller, IpProfile_CopyDest2RedirectNumber, IpProfile_MediaSecurityBehaviour, IpProfile_CallLimit, IpProfile_DisconnectOnBrokenConnection, IpProfile_FirstTxDtmfOption, IpProfile_SecondTxDtmfOption, IpProfile_RxDTMFOption, IpProfile_EnableHold, IpProfile_InputGain, IpProfile_VoiceVolume, IpProfile_AddIEInSetup, IpProfile_SBCExtensionCodersGroupID, IpProfile_MediaIPVersionPreference, IpProfile_TranscodingMode, IpProfile_SBCAllowedMediaTypes, IpProfile_SBCAllowedCodersGroupID, IpProfile_SBCAllowedVideoCodersGroupID, IpProfile_SBCAllowedCodersMode, IpProfile_SBCMediaSecurityBehaviour, IpProfile_SBCRFC2833Behavior, IpProfile_SBCAlternativeDTMFMethod, IpProfile_SBCAssertIdentity, IpProfile_AMDSensitivityParameterSuit, IpProfile_AMDSensitivityLevel, IpProfile_AMDMaxGreetingTime, IpProfile_AMDMaxPostSilenceGreetingTime, IpProfile_SBCDiversionMode, IpProfile_SBCHistoryInfoMode, IpProfile_EnableQSIGTunneling, IpProfile_SBCFaxCodersGroupID, IpProfile_SBCFaxBehavior, IpProfile_SBCFaxOfferMode, IpProfile_SBCFaxAnswerMode, IpProfile_SbcPrackMode, IpProfile_SBCSessionExpiresMode, IpProfile_SBCRemoteUpdateSupport, IpProfile_SBCRemoteReinviteSupport, IpProfile_SBCRemoteDelayedOfferSupport, IpProfile_SBCRemoteReferBehavior, IpProfile_SBCRemote3xxBehavior, IpProfile_SBCRemoteMultiple18xSupport, IpProfile_SBCRemoteEarlyMediaResponseType, IpProfile_SBCRemoteEarlyMediaSupport, IpProfile_EnableSymmetricMKI, IpProfile_MKISize, IpProfile_SBCEnforceMKISize, IpProfile_SBCRemoteEarlyMediaRTP, IpProfile_SBCRemoteSupportsRFC3960, IpProfile_SBCRemoteCanPlayRingback, IpProfile_EnableEarly183, IpProfile_EarlyAnswerTimeout, IpProfile_SBC2833DTMFPayloadType, IpProfile_SBCUserRegistrationTime, IpProfile_ResetSRTPStateUponRekey, IpProfile_AmdMode, IpProfile_SBCReliableHeldToneSource, IpProfile_GenerateSRTPKeys, IpProfile_SBCPlayHeldTone, IpProfile_SBCRemoteHoldFormat, IpProfile_SBCRemoteReplacesBehavior, IpProfile_SBCSDPPtimeAnswer, IpProfile_SBCPreferredPTime, IpProfile_SBCUseSilenceSupp, IpProfile_SBCRTPRedundancyBehavior, IpProfile_SBCPlayRBTToTransferee, IpProfile_SBCRTCPMode, IpProfile_SBCJitterCompensation, IpProfile_SBCRemoteRenegotiateOnFaxDetection, IpProfile_JitterBufMaxDelay; [\IPProfile] For a description of this table, see "Configuring IP Profiles" on page 314.
Tel Profile Table	
Web: Tel Profile Settings EMS: Protocol Definition > Telephony Profile CLI: configure voip > coders-and-profiles tel-profile [TelProfile]	This table parameter configures the Tel Profile table. Each Tel Profile ID includes a set of parameters (which are typically configured separately using their individual, "global" parameters). You can later assign these Tel Profile IDs to other elements such as in the Trunk Group table (TrunkGroup parameter). Therefore, Tel Profiles allow you to apply the same settings of a group of parameters to multiple channels, or apply specific settings to different channels.

Parameter	Description
	The format of the ini file table parameter is as follows: [TelProfile] FORMAT TelProfile_Index = TelProfile_ProfileName, TelProfile_TelPreference, TelProfile_CodersGroupID, TelProfile_IsFaxUsed, TelProfile_JitterBufMinDelay, TelProfile_JitterBufOptFactor, TelProfile_IPDiffServ, TelProfile_SigIPDiffServ, TelProfile_DtmfVolume, TelProfile_InputGain, TelProfile_VoiceVolume, TelProfile_EnableReversePolarity, TelProfile_EnableCurrentDisconnect, TelProfile_EnableDigitDelivery, TelProfile_EnableEC, TelProfile_MWIAnalog, TelProfile_MWIDisplay, TelProfile_FlashHookPeriod, TelProfile_EnableEarlyMedia, TelProfile_ProgressIndicator2IP, TelProfile_TimeForReorderTone, TelProfile_EnableDIDWink, TelProfile_IsTwoStageDial, TelProfile_DisconnectOnBusyTone, TelProfile_EnableVoiceMailDelay, TelProfile_DialPlanIndex, TelProfile_Enable911PSAP, TelProfile_SwapTelToIpPhoneNumbers, TelProfile_EnableAGC, TelProfile_ECNlpMode, TelProfile_DigitalCutThrough, TelProfile_EnableFXODoubleAnswer, TelProfile_CallPriorityMode; [\TelProfile] For a description of this parameter, see Configuring Tel Profiles on page 311.

67.9 Channel Parameters

This subsection describes the device's channel parameters.

67.9.1 Voice Parameters

The voice parameters are described in the table below.

Table 67-37: Voice Parameters

Parameter	Description
Web/EMS: Input Gain CLI: 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) or Tel Profiles. For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314 or in the Tel Profile table, see Configuring Tel Profiles on page 311. **Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
Web: Voice Volume EMS: Volume (dB) CLI: voice-volume **[VoiceVolume]**	Global parameter that defines the voice gain control (in decibels). This defines the level of the transmitted (IP-to-Tel) signal. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VoiceVolume) or Tel Profiles. For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314 or in the Tel Profile table, see Configuring Tel Profiles on page 311.

Parameter	Description
	Note: If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
EMS: Payload Format CLI: G726-voice-payload-format **[VoicePayloadFormat]**	Determines the bit ordering of the G.726/G.727 voice payload format. ▪ **[0]** = (Default) Little Endian ▪ **[1]** = Big Endian **Note:** To ensure high voice quality when using G.726/G.727, both communicating ends should use the same endianness format. Therefore, when the device communicates with a third-party entity that uses the G.726/G.727 voice coder and voice quality is poor, change the settings of this parameter (between Big Endian and Little Endian).
Web: MF Transport Type CLI: MF-transport-type **[MFTransportType]**	Currently, not supported.
Web: Silence Suppression EMS: Silence Compression Mode CLI: silence-compression-mode **[EnableSilenceCompression]**	Global parameter that enables the Silence Suppression feature. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SCE). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Echo Canceler EMS: Echo Canceller Enable CLI: 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) or Tel Profiles. For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314 or Tel Profile table, see Configuring Tel Profiles on page 311. **Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
Web: Network Echo Suppressor Enable CLI: 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 this parameter to take effect, a device reset is required.
Web: Echo Canceller Type CLI: 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.

Parameter	Description
Web: Attenuation Intensity CLI: 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.
Web: Max ERL Threshold - DB CLI: 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.
Web: Min Reference Delay x10 msec CLI: 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.
Web: Max Reference Delay x10 msec CLI: 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).
EMS: Echo Canceller Hybrid Loss CLI: 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
EMS: ECN lp Mode CLI: echo-canceller-NLP-mode **[ECNLPMode]**	Defines the echo cancellation Non-Linear Processing (NLP) mode. ▪ **[0]** = (Default) NLP adapts according to echo changes ▪ **[1]** = Disables NLP ▪ **[2]** = Silence output NLP **Note:** This parameter can also be configured in a Tel Profile.
CLI: 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. **Note:** For this parameter to take effect, a device reset is required.
CLI: 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	
Web: Enable Answer Detector **[EnableAnswerDetector]**	Currently, not supported.
Web: Answer Detector Activity Delay CLI: 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.

Parameter	Description
	The valid range is 0 to 1023. The default is 0.
Web: Answer Detector Silence Time **[AnswerDetectorSilenceTime]**	Currently, not supported.
Web: Answer Detector Redirection **[AnswerDetectorRedirection]**	Currently, not supported.
Web: Answer Detector Sensitivity EMS: Sensitivity CLI: answer-detector-sensitivity **[AnswerDetectorSensitivity]**	Defines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0.

67.9.2 Coder Parameters

The coder parameters are described in the table below.

Table 67-38: Coder Parameters

Parameter	Description
Silk Tx Inband FEC CLI: silk-tx-inband-fec [SilkTxInbandFEC]	Enables forward error correction (FEC) for the SILK coder. ▪ [0] Disable (default) ▪ [1] Enable
Silk Max Average Bit Rate CLI: 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. **Note:** The SILK coder is Skype's default audio codec used for Skype-to-Skype calls.
EMS: VBR Coder Header Format CLI: 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.
EMS: VBR Coder Hangover CLI: 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.
Web: AMR Payload Format [AmrOctetAlignedEnable]	Defines the AMR payload format type. ▪ [0] Bandwidth Efficient ▪ [1] Octet Aligned (default)
EMS: AMR Coder Header Format [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.

67.9.3 DTMF Parameters

The dual-tone multi-frequency (DTMF) parameters are described in the table below.

Table 67-39: DTMF Parameters

Parameter	Description
Web/EMS: DTMF Transport Type CLI: 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:** This parameter is automatically updated if the parameters TxDTMFOption or RxDTMFOption are configured.
Web: DTMF Volume (-31 to 0 dB) EMS: DTMF Volume (dBm) CLI: DTMF-volume **[DTMFVolume]**	Defines the DTMF gain control value (in decibels) to the Tel side. The valid range is -31 to 0 dB. The default is -11 dB. **Note:** This parameter can also be configured in a Tel Profile.
Web: DTMF Generation Twist EMS: DTMF Twist Control CLI: 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 this parameter to take effect, a device reset is required.
EMS: DTMF Inter Interval (msec) CLI: inter-digit-interval **[DTMFInterDigitInterval]**	Defines the time (in msec) between generated DTMF digits to the Tel side (if TxDTMFOption = 1, 2 or 3). The valid range is 0 to 32767. The default is 100.
EMS: DTMF Length (msec) **[DTMFDigitLength]**	Defines the time (in msec) for generating DTMF tones to the Tel side (if TxDTMFOption = 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.
EMS: Rx DTMF Relay Hang Over Time (msec) CLI: default-dtmf-signal-duration **[RxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after playing DTMF or MF digits to the Tel side that arrive as Relay from the IP side. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
EMS: Tx DTMF Relay Hang Over Time (msec) CLI: digit-hangover-time-tx **[TxDTMFHangOverTime]**	Defines the Voice Silence time (in msec) after detecting the end of DTMF or MF digits at the Tel side when the DTMF Transport Type is either Relay or Mute. Valid range is 0 to 2,000 msec. The default is 1,000 msec.
Web/EMS: NTE Max Duration CLI: telephony-events-max-duration	Defines the maximum time for sending Named Telephony Events / NTEs (RFC 4733/2833 DTMF relay) to the IP side, regardless of

Parameter	Description
[NTEMaxDuration]	the DTMF signal duration on the TDM 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).

67.9.4 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

Table 67-40: RTP/RTCP and T.38 Parameters

Parameter	Description
Web: Dynamic Jitter Buffer Minimum Delay EMS: Minimal Delay (dB) CLI: 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) or Tel Profiles. For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 314, or in the Tel Profile table, see Configuring Tel Profiles on page 311. **Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
Web: Dynamic Jitter Buffer Optimization Factor EMS: Opt Factor CLI: 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 or Tel Profiles. For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 314, or in the Tel Profile table, see Configuring Tel Profiles on page 311. **Note:** If this functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.
Web/EMS: Analog Signal Transport Type [AnalogSignalTransportType]	Determines the analog signal transport type. ▪ [0] Ignore Analog Signals = (Default) Ignore. ▪ [1] RFC 2833 Analog Signal Relay = Transfer hookflash using RFC 2833. **Note:** This parameter is applicable only to FXS and FXO interfaces.
Web: RTP Redundancy Depth EMS: Redundancy Depth CLI: 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 this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Enable RTP Redundancy Negotiation CLI: rtp-rdcy-nego-enbl [EnableRTPRedundancyNegotiation]	Enables the device to include the RTP redundancy dynamic payload type in the SDP, according to RFC 2198. ▪ [0] Disable (default)

Parameter	Description
	- [1] Enable When enabled, the device includes in the SDP message the RTP payload type "RED" and the payload type configured by the parameter RFC2198PayloadType. `a=rtpmap:<PT> RED/8000` Where <PT> is the payload type as defined by RFC2198PayloadType. The device sends the INVITE message with "a=rtpmap:<PT> RED/8000" and responds with a 18x/200 OK and "a=rtpmap:<PT> RED/8000" in the SDP. **Notes:** - For this feature to be functional, you must also set the parameter RTPRedundancyDepth to 1 (i.e., enabled). - Currently, the negotiation of “RED” payload type is not supported and therefore, it should be configured to the same PT value for both parties.
Web: RFC 2198 Payload Type EMS: Redundancy Payload Type CLI: RTP-redundancy-payload-type [RFC2198PayloadType]	Defines the RTP redundancy packet payload type according to RFC 2198. The range is 96 to 127. The default is 104. **Note:** This parameter is applicable only if the parameter RTPRedundancyDepth is set to 1.
Web: Packing Factor EMS: Packetization Factor [RTPPackingFactor]	N/A. Controlled internally by the device according to the selected coder.
Web/EMS: Basic RTP Packet Interval [BasicRTPPacketInterval]	N/A. Controlled internally by the device according to the selected coder.
Web: RTP Directional Control [RTPDirectionControl]	N/A. Controlled internally by the device according to the selected coder.
Web/EMS: RFC 2833 TX Payload Type CLI: telephony-events-payload-type-tx [RFC2833TxPayloadType]	Defines the Tx RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 127. The default is 96. **Note:** When RFC 2833 payload type negotiation is used (i.e., the parameter TxDTMFOption 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.
Web/EMS: RFC 2833 RX Payload Type CLI: telephony-events-payload-type-rx [RFC2833RxPayloadType]	Defines the Rx RFC 2833 DTMF relay dynamic payload type. The valid range is 96 to 127. The default is 96. **Note:** When RFC 2833 payload type negotiation is used (i.e., the parameter TxDTMFOption 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.

Parameter	Description
[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 this 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 this parameter to 0 or 2.
Web: RTP Base UDP Port EMS: Base UDP Port **[BaseUDPport]**	Global parameter that defines the lower boundary of the UDP port used for RTP, RTCP (RTP port + 1) and T.38 (RTP port + 2). For more information on configuring the UDP port range, see "Configuring RTP Base UDP Port" on page 194. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_RemoteBaseUDPPort) - see "Configuring IP Profiles" on page 314. The range of possible UDP ports is 6,000 to 65,535. The default base UDP port is 6000. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
EMS: No Op Enable CLI: no-operation-enable **[NoOpEnable]**	Enables the transmission of RTP or T.38 No-Op packets. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.
EMS: No Op Interval **[NoOpInterval]**	Defines the time interval in which RTP or T.38 No-Op packets are sent in the case of silence (no RTP/T.38 traffic) when No-Op packet transmission is enabled. The valid range is 20 to 65,000 msec. The default is 10,000. **Note:** To enable No-Op packet transmission, use the NoOpEnable parameter.

Parameter	Description
EMS: No Op Payload Type CLI: no-operation-interval **[RTPNoOpPayloadType]**	Defines the payload type of No-Op packets. The valid range is 96 to 127 (for the range of Dynamic RTP Payload Type for all types of non hard-coded RTP Payload types, refer to RFC 3551). The default is 120. **Note:** When defining this parameter, ensure that it doesn't cause collision with other payload types.
CLI: rtcp-act-mode [RTCPActivationMode]	Disables RTCP traffic when there is no RTP traffic. This feature is useful, for example, to stop RTCP traffic that is typically sent when calls are put on hold (by an INVITE with 'a=inactive' in the SDP). ▪ [0] Active Always = (Default) RTCP is active even during inactive RTP periods, i.e., when the media is in 'recvonly' or 'inactive' mode. ▪ [1] Inactive Only If RTP Inactive = No RTCP is sent when RTP is inactive. **Note:** This parameter is applicable only to Gateway calls (not SBC).
RTP Control Protocol Extended Reports (RTCP XR) Parameters	
Web: Enable RTCP XR EMS: RTCP XR Enable CLI: voice-quality-monitoring-enable **[VQMonEnable]**	Enables voice quality monitoring and RTCP XR, according to RFC 3611. ▪ **[0]** Disable (default) ▪ **[1]** Enable Fully = Calculates voice quality metrics, uses them for QoE calculations, reports them to SEM (if configured), and sends them to remote side using RTCP XR. ▪ **[2]** Enable Calculation Only = Calculates voice quality metrics, uses them for QoE calculations, reports them to SEM (if configured), but does not send them to remote side using RTCP XR. **Note:** For this parameter to take effect, a device reset is required.
Web: Minimum Gap Size EMS: GMin **[VQMonGMin]**	Defines the voice quality monitoring - minimum gap size (number of frames). The default is 16.
Web/EMS: Burst Threshold **[VQMonBurstHR]**	Defines the voice quality monitoring - excessive burst alert threshold. The default is -1 (i.e., no alerts are issued).
Web/EMS: Delay Threshold **[VQMonDelayTHR]**	Defines the voice quality monitoring - excessive delay alert threshold. The default is -1 (i.e., no alerts are issued).
Web: R-Value Delay Threshold EMS: End of Call Rval Delay Threshold **[VQMonEOCRValTHR]**	Defines the voice quality monitoring - end of call low quality alert threshold. The default is -1 (i.e., no alerts are issued).
Web: RTCP XR Packet Interval EMS: Packet Interval CLI: rtcp-interval **[RTCPInterval]**	Defines the time interval (in msec) between adjacent RTCP XR reports. This interval starts from call establishment. Thus, the device can send RTCP XR reports during the call, in addition to at the end of the call. If the duration of the call is

Parameter	Description
	shorter than this interval, RTCP XR is sent only at the end of the call. The valid value range is 0 to 65,535. The default is 5,000.
Web: Disable RTCP XR Interval Randomization EMS: Disable Interval Randomization CLI: disable-RTCP-randomization **[DisableRTCPRandomize]**	Determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval. ▪ **[0]** Disable = (Default) Randomize ▪ **[1]** Enable = No Randomize
EMS: RTCP XR Collection Server Transport Type **[RTCPXRESCTransportType]**	Defines the transport layer used for outgoing SIP dialogs initiated by the device to the RTCP XR Collection Server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Notes:** ▪ When set to [-1], the value of the SIPTransportType parameter is used. ▪ This parameter is applicable only to the Gateway application.
Web: RTCP XR Collection Server EMS: Esc IP CLI: rtcp-xr-coll-srvr **[RTCPXREscIP]**	Defines the IP address of the Event State Compositor (ESC). The device sends RTCP XR reports to this server, using SIP PUBLISH messages, according to Internet-Draft draft-ietf-sipping-rtcp-summary-13. The address can be configured as a numerical IP address or as a domain name. **Note:** This parameter is applicable only to the Gateway application.
Web: Gateway RTCP XR Report Mode EMS: Report Mode CLI: rtcp-xr-rep-mode **[RTCPXRReportMode]**	Determines whether RTCP XR reports are sent to the Event State Compositor (ESC) and defines the interval at which they are sent. ▪ [0] Disable = (Default) RTCP XR reports are not sent. ▪ [1] End Call = RTCP XR reports are sent at the end of each call. ▪ [2] End Call & Periodic = RTCP XR reports are sent at the end of each call and periodically according to the RTCPInterval parameter. **Note:** This parameter is applicable only to the Gateway application.
Web: SBC RTCP XR Report Mode CLI: sbc-rtcpxr-report-mode [SBCRtcpXrReportMode]	Enables the sending of RTCP XR reports of QoE metrics at the end of each call session (i.e., after a SIP BYE). The RTCP XR is sent in the SIP PUBLISH message. ▪ [0] Disable (default) ▪ [1] End of Call **Note:** This parameter is applicable only to the SBC application.

67.10 Gateway and IP-to-IP Parameters

67.10.1 Fax and Modem Parameters

The fax and modem parameters are described in the table below.

Table 67-41: Fax and Modem Parameters

Parameter	Description
Web: Fax Transport Mode EMS: Transport Mode CLI: fax-transport-mode **[FaxTransportMode]**	Determines the fax transport mode used by the device. ▪ **[0]** Disable = transparent mode ▪ **[1]** T.38 Relay (default) ▪ **[2]** Bypass ▪ **[3]** Events Only **Note:** This parameter is overridden by the parameter IsFaxUsed. If the parameter IsFaxUsed is set to 1 (T.38 Relay) or 3 (Fax Fallback), then FaxTransportMode is always set to 1 (T.38 relay).
EMS: V34 Transport Method CLI: V34-fax-transport-type [V34FaxTransportType]	Determines the V.34 fax transport method (whether V34 fax falls back to T.30 or pass over Bypass). ▪ [0] = Transparent ▪ [1] = (Default) Relay ▪ [2] = Bypass ▪ [3] = Transparent with Events **Note:** To configure V34FaxTransportType to 1 (i.e., fax relay), you also need to configure FaxTransportMode to 1 (fax relay).
Web: V.21 Modem Transport Type EMS: V21 Transport CLI: V21-modem-transport-type **[V21ModemTransportType]**	Determines the V.21 modem transport type. ▪ **[0]** Disable = (Default) Transparent. ▪ **[2]** Enable Bypass ▪ **[3]** Events Only = Transparent with Events. **Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 314.
Web: V.22 Modem Transport Type EMS: V22 Transport CLI: V22-modem-transport-type **[V22ModemTransportType]**	Determines the V.22 modem transport type. ▪ **[0]** Disable = Transparent. ▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events. **Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 314.
Web: V.23 Modem Transport Type EMS: V23 Transport CLI: V23-modem-transport-type **[V23ModemTransportType]**	Determines the V.23 modem transport type. ▪ **[0]** Disable = Transparent. ▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events. **Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 314.
Web: V.32 Modem Transport Type	Determines the V.32 modem transport type. ▪ **[0]** Disable = Transparent.

Parameter	Description
EMS: V32 Transport CLI: V32-modem-transport-type **[V32ModemTransportType]**	▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events. **Notes:** ▪ This parameter applies only to V.32 and V.32bis modems. ▪ You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 314.
Web: V.34 Modem Transport Type EMS: V34 Transport CLI: V34-modem-transport-type **[V34ModemTransportType]**	Determines the V.90/V.34 modem transport type. ▪ **[0]** Disable = Transparent. ▪ **[2]** Enable Bypass (default) ▪ **[3]** Events Only = Transparent with Events. **Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_VxxTransportType). For more information, see "Configuring IP Profiles" on page 314.
EMS: Bell Transport Type CLI: bell-modem-transport-type **[BellModemTransportType]**	Determines the Bell modem transport method. ▪ **[0]** = Transparent (default) ▪ **[2]** = Bypass ▪ **[3]** = Transparent with events
Web/EMS: Fax CNG Mode CLI: fax_cng_mode **[FaxCNGMode]**	Determines the device's handling of fax relay upon detection of a fax CNG tone or a V.34/Super G3 V8-CM (Call Menu) signal from originating faxes. ▪ **[0]** Doesn't send T.38 Re-INVITE = (Default) SIP re-INVITE is not sent. ▪ **[1]** Sends on CNG tone = Sends a SIP re-INVITE with T.38 parameters in SDP to the terminating fax upon detection of a fax CNG tone, if the CNGDetectorMode parameter is set to 1. ▪ [2] Sends on CNG or v8-cn = Sends a SIP re-INVITE with T.38 parameters in SDP to the terminating fax upon detection of a fax CNG tone (if the CNGDetectorMode parameter is set to 1) or upon detection of a V8-CM signal. **Notes:** ▪ If this parameter is set to [2] and the CNGDetectorMode parameter is set to [0], the device sends a re-INVITE only if it detects a V8-CM signal from the originating fax. ▪ This feature is applicable only if the IsFaxUsed parameter is set to [1] or [3]. ▪ The device also sends T.38 re-INVITE if the CNGDetectorMode parameter is set to [2], regardless of the FaxCNGMode parameter settings.
Web/EMS: CNG Detector Mode CLI: coder **[CNGDetectorMode]**	Global parameter that enables the detection of the fax calling tone (CNG) and defines the detection method. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_CNGmode). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Fax Detect Timeout Since Connect (msec)	Defines a timeout (in msec) for detecting fax from the Tel side during an established voice call. The interval starts from when the voice call is established. If the device detects a fax tone within the

Parameter	Description
CLI: configure voip > sip-definition general-settings > fax-detect-timeout-since-connect **[FaxDetectTimeoutSinceConnect]**	interval, it ends the voice session and sends a T.38 or VBD re-INVITE message to the IP side and processes the fax. If the interval expires without any received fax event, the device ignores all subsequent fax events during the voice session. The valid value is 0 to 120000. The default is 0. If set to 0, the device can detect fax during the entire voice call.
Web: SIP T.38 Version CLI: sip-t38-ver [SIPT38Version]	Determines the T.38 fax relay version. ▪ [-1] Not Configured = (Default) No T.38 ▪ [0] Version 0 ▪ [3] Version 3 = T.38 Version 3 (V.34 over T.38) **Note:** For a description on V.34 over T.38 fax relay, see V.34 Fax Support on page 184.
Web: Fax Relay Enhanced Redundancy Depth EMS: Enhanced Relay Redundancy Depth CLI: enhanced-redundancy-depth **[FaxRelayEnhancedRedundancyDepth]**	Defines the number of times that control packets are retransmitted when using the T.38 standard. The valid range is 0 to 4. The default is 2.
Web: Fax Relay Redundancy Depth EMS: Relay Redundancy Depth CLI: redundancy-depth **[FaxRelayRedundancyDepth]**	Defines the number of times that each fax relay payload is retransmitted to the network. ▪ **[0]** = (Default) No redundancy ▪ **[1]** = One packet redundancy ▪ **[2]** = Two packet redundancy **Note:** This parameter is applicable only to non-V.21 packets.
Web: Fax Relay Max Rate (bps) EMS: Relay Max Rate CLI: max-rate **[FaxRelayMaxRate]**	Defines the maximum rate (in bps) at which fax relay messages are transmitted (outgoing calls). ▪ **[0]** 2400 = 2.4 kbps ▪ **[1]** 4800 = 4.8 kbps ▪ **[2]** 7200 = 7.2 kbps ▪ **[3]** 9600 = 9.6 kbps ▪ **[4]** 12000 = 12.0 kbps ▪ **[5]** 14400 = 14.4 kbps (default) ▪ [6] 16800bps = 16.8 kbps ▪ [7] 19200bps = 19.2 kbps ▪ [8] 21600bps = 21.6 kbps ▪ [9] 24000bps = 24 kbps ▪ [10] 26400bps = 26.4 kbps ▪ [11] 28800bps = 28.8 kbps ▪ [12] 31200bps = 31.2 kbps ▪ [13] 33600bps = 33.6 kbps **Notes:** ▪ The rate is negotiated between both sides (i.e., the device adapts to the capabilities of the remote side). Negotiation of the T.38 maximum supported fax data rate is provided in SIP's SDP T38MaxBitRate parameter. The negotiated T38MaxBitRate is the minimum rate supported between the local and remote endpoints. ▪ Fax relay rates greater than 14.4 kbps are applicable only to

Parameter	Description
	V.34 / T.38 fax relay. For non-T.38 V.34 supporting devices, configuration greater than 14.4 kbps is truncated to 14.4 kbps.
Web: Fax Relay ECM Enable EMS: Relay ECM Enable CLI: ecm-mode **[FaxRelayECMEnable]**	Enables Error Correction Mode (ECM) mode during fax relay. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Web: Fax/Modem Bypass Coder Type EMS: Coder Type **[FaxModemBypassCoderType]**	Determines the coder used by the device when performing fax/modem bypass. Typically, high-bit-rate coders such as G.711 should be used. ▪ **[0]** G.711Alaw= (Default) G.711 A-law 64 ▪ **[1]** G.711Mulaw = G.711 μ-law
Web: Fax/Modem Bypass Packing Factor EMS: Packetization Period CLI: packing-factor **[FaxModemBypassM]**	Defines the number (20 msec) of coder payloads used to generate a fax/modem bypass packet. The valid range is 1, 2, or 3 coder payloads. The default is 1 coder payload.
CLI: fax-modem-telephony-events-mode **[FaxModemNTEMode]**	Determines whether the device sends RFC 2833 ANS/ANSam events upon detection of fax and/or modem Answer tones (i.e., CED tone). ▪ **[0]** = Disabled (default) ▪ **[1]** = Enabled **Note:** This parameter is applicable only when the fax or modem transport type is set to bypass or Transparent-with-Events.
Web/EMS: Fax Bypass Payload Type CLI: fax-bypass-payload-type **[FaxBypassPayloadType]**	Defines the fax bypass RTP dynamic payload type. The valid range is 96 to 120. The default is 102.
EMS: Modem Bypass Payload Type CLI: modem-bypass-payload-type **[ModemBypassPayloadType]**	Defines the modem bypass dynamic payload type. The range is 0-127. The default is 103.
EMS: Relay Volume (dBm) CLI: volume **[FaxModemRelayVolume]**	Defines the fax gain control. The range is -18 to -3, corresponding to -18 dBm to -3 dBm in 1-dB steps. The default is -6 dBm fax gain control.
Web/EMS: Fax Bypass Output Gain CLI: fax-bypass-output-gain **[FaxBypassOutputGain]**	Defines the fax bypass output gain control. The range is -31 to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).
Web/EMS: Modem Bypass Output Gain **[ModemBypassOutputGain]**	Defines the modem bypass output gain control. The range is -31 dB to +31 dB, in 1-dB steps. The default is 0 (i.e., no gain).

Parameter	Description
EMS: Basic Packet Interval CLI: modem-bypass-output-gain **[FaxModemBypassBasicRTPPacketInterval]**	Defines the basic frame size used during fax/modem bypass sessions. ▪ **[0]** = (Default) Determined internally ▪ **[1]** = 5 msec (not recommended) ▪ **[2]** = 10 msec ▪ **[3]** = 20 msec **Note:** When set to 5 msec (1), the maximum number of simultaneous channels supported is 120.
EMS: Dynamic Jitter Buffer Minimal Delay (dB) CLI: jitter-buffer-minimum-delay **[FaxModemBypasDJBufMinDelay]**	Defines the Jitter Buffer delay (in milliseconds) during fax and modem bypass session. The range is 0 to 150 msec. The default is 40.
EMS: Enable Inband Network Detection CLI: enable-fax-modem-inband-network-detection **[EnableFaxModemInbandNetworkDetection]**	Enables in-band network detection related to fax/modem. ▪ **[0]** = (Default) Disable. ▪ **[1]** = Enable. When this parameter is enabled on Bypass and transparent with events mode (VxxTransportType is set to 2 or 3), a detection of an Answer Tone from the network triggers a switch to bypass mode in addition to the local Fax/Modem tone detections. However, only a high bit-rate coder voice session effectively detects the Answer Tone sent by a remote endpoint. This can be useful when, for example, the payload of voice and bypass is the same, allowing the originator to switch to bypass mode as well.
EMS: NSE Mode CLI: NSE-mode **[NSEMode]**	Global parameter that enables Cisco's compatible fax and modem bypass mode, Named Signaling Event (NSE) packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_NSEMode). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **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.
EMS: NSE Payload Type CLI: NSE-payload-type **[NSEPayloadType]**	Defines the NSE payload type for Cisco Bypass compatible mode. The valid range is 96-127. The default is 105. **Note:** Cisco gateways usually use NSE payload type of 100.
EMS: T38 Use RTP Port **[T38UseRTPPort]**	Defines the port (with relation to RTP port) for sending and receiving T.38 packets. ▪ **[0]** = (Default) Use the RTP port +2 to send/receive T.38 packets. ▪ **[1]** = Use the same port as the RTP port to send/receive T.38 packets. **Notes:** ▪ For this parameter to take effect, you must reset the device. ▪ When the device is configured to use V.152 to negotiate audio and T.38 coders, the UDP port published in SDP for RTP and for T38 must be different. Therefore, set the T38UseRTPPort parameter to 0.

Parameter	Description
Web/EMS: T.38 Max Datagram Size CLI: t38-mx-datagram-sz **[T38MaxDatagramSize]**	Defines the maximum size of a T.38 datagram that the device can receive. This value is included in the outgoing SDP when T.38 is used. The valid range is 120 to 600. The default is 560.
Web/EMS: T38 Fax Max Buffer CLI: t38-fax-mx-buff **[T38FaxMaxBufferSize]**	Defines the maximum size (in bytes) of the device's T.38 buffer. This value is included in the outgoing SDP when T.38 is used for fax relay over IP. The valid range is 500 to 3000. The default is 3000.
Web: Detect Fax on Answer Tone EMS: Enables Detection of FAX on Answer Tone CLI: det-fax-on-ans-tone **[DetFaxOnAnswerTone]**	Determines when the device initiates a T.38 session for fax transmission. ▪ **[0]** Initiate T.38 on Preamble = (Default) The device to which the called fax is connected initiates a T.38 session on receiving HDLC Preamble signal from the fax. ▪ **[1]** Initiate T.38 on CED = The device to which the called fax is connected initiates a T.38 session on receiving a CED answer tone from the fax. This option can only be used to relay fax signals, as the device sends T.38 Re-INVITE on detection of any fax/modem Answer tone (2100 Hz, amplitude modulated 2100 Hz, or 2100 Hz with phase reversals). The modem signal fails when using T.38 for fax relay. **Note:** This parameters is applicable only if the IsFaxUsed parameter is set to 1 (T.38 Relay) or 3 (Fax Fallback).
Web: CED Transfer Mode [CEDTransferMode]	Defines the method for sending fax/modem CED (answering) tones. ▪ [0] Fax Relay or VBD = (Default) The device transfers the CED tone in Relay mode and starts the fax session immediately. ▪ [1] Voice Mode or VBD = The device transfers the CED tone in either Voice or Bypass mode and starts the fax session on V21 preamble. ▪ [2] RFC 4733 Blocking RTP VBD = The device transfers the CED tone in RFC 2833. This is applicable only to V.150.1 modem relay and fax bypass. ▪ [3] RFC 4733 Along with RTP VBD = The device transfers the CED tone in RFC 2833 and bypass, in parallel. For combined V.150.1 modem relay and fax relay, use this option.
Web: T.38 Fax Session CLI: t38-sess-imm-strt **[T38FaxSessionImmediateStart]**	Enables fax transmission of T.38 "no-signal" packets to the terminating fax machine. ▪ **[0]** Disable (default) ▪ **[1]** Immediate Start on Fax = Device activates T.38 fax relay upon receipt of a re-INVITE with T.38 only in the SDP. ▪ **[2]** Immediate Start on Fax & Voice = Device activates T.38 fax relay upon receipt of a re-INVITE with T.38 and audio media in the SDP. This parameter is used for transmission from fax machines connected to the device and located inside a NAT. Generally, the firewall blocks T.38 (and other) packets received from the WAN, unless the device behind 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

Parameter	Description
	originating fax does not wait for an answer, but immediately starts sending T.38 packets to the terminating fax machine. **Note:** To enable No-Op packet transmission, use the NoOpEnable and NoOpInterval parameters.
Web: Profile Number EMS: Allocation Profile [V1501AllocationProfile]	Defines the V.150.1 profile, which determines how many DSP channels support V.150.1. The value range is 0 to 20. The default is 0. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: SSE Payload Type Rx CLI: V1501-SSE-payload-type-rx [V1501SSEPayloadTypeRx]	Defines the V.150.1 (modem relay protocol) State Signaling Event (SSE) payload type Rx. The value range is 96 to 127. The default is 105.
Web/EMS: SSE Redundancy Depth CLI: SSE-redundancy-depth [V1501SSERedundancyDepth]	Defines the SSE redundancy depth. The value range is 1-6. The default is 3.
Web: SPRT Transport Ch.0 Max Payload Size CLI: SPRT-transport-channel0-max-payload-size [V1501SPRTTransportChannel0MaxPayloadSize]	Defines the maximum payload size for V.150.1 SPRT Transport Channel 0. The range is 140 to 256. The default is 140.
Web: SPRT Transport Ch.2 Max Payload Size CLI: SPRT-transport-channel2-max-payload-size [V1501SPRTTransportChannel2MaxPayloadSize]	Defines the maximum payload size for V.150.1 SPRT Transport Channel 2. The range is 132 to 256. The default is 132.
Web: SPRT Transport Ch.2 Max Window Size CLI: SPRT-transport-channel2-max-window-size [V1501SPRTTransportChannel2MaxWindowSize]	Defines the maximum window size of SPRT transport channel 2. The value range is 8 to 32. The default is 8.
Web: SPRT Transport Ch.3 Max Payload Size CLI: SPRT-transport-channel3-max-payload-size [V1501SPRTTransportChannel3MaxPayloadSize]	Defines the maximum payload size for V.150.1 SPRT Transport Channel 3. The range is 140 to 256. The default is 140.

67.10.2 DTMF and Hook-Flash Parameters

The DTMF and hook-flash parameters are described in the table below.

Table 67-42: DTMF and Hook-Flash Parameters

Parameter	Description
Hook-Flash Parameters	

Parameter	Description
Web/EMS: Hook-Flash Code CLI: hook-flash-code **[HookFlashCode]**	For analog interfaces: Defines the digit pattern that when received from the Tel side, indicates a Hook Flash event. For digital interfaces: Defines the digit pattern used by the PBX to indicate a Hook Flash event. When this pattern is detected from the Tel side, the device responds as if a Hook Flash event has occurred and sends a SIP INFO message if the HookFlashOption parameter is set to 1, 5, 6, or 7 (indicating a Hook Flash). If configured and a Hook Flash indication is received from the IP side, the device generates this pattern to the Tel side. The valid range is a 25-character string. The default is a null string. **Note:** This parameter can also be configured in a Tel Profile.

Web/EMS: Hook-Flash Option
CLI: hook-flash-option
[HookFlashOption]
	Defines the hook-flash transport type (i.e., method by which hook-flash is sent and received). For digital interfaces: This feature is applicable only if the HookFlashCode parameter is configured.
	▪ **[0]** Not Supported = (Default) Hook-Flash indication is not sent.
	▪ **[1]** INFO = Sends proprietary INFO message (Broadsoft) with Hook-Flash indication. The device sends the INFO message as follows:
	Content-Type: application/broadsoft; version=1.0
	Content-Length: 17
	event flashhook
	▪ **[4]** RFC 2833 = This option is currently not supported.
	▪ **[5]** INFO (Lucent) = Sends proprietary SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:
	Content-Type: application/hook-flash
	Content-Length: 11
	signal=hf
	▪ **[6]** INFO (NetCentrex) = Sends proprietary SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:
	`Content-Type: application/dtmf-relay`
	`Signal=16`
	Where 16 is the DTMF code for hook flash.
	▪ **[7]** INFO (HUAWEI) = Sends a SIP INFO message with Hook-Flash indication. The device sends the INFO message as follows:
	`Content-Length: 17`
	`Content-Type: application/sscc`
	`event=flashhook`
	Notes:
	▪ The device can interwork DTMF HookFlashCode to SIP INFO messages with Hook Flash indication (for digital interfaces).
	▪ FXO interfaces support only the receipt of RFC 2833 Hook-Flash signals and INFO [1] type.
	▪ FXS interfaces send Hook-Flash signals only if the EnableHold parameter is set to 0.

Parameter	Description
Web: Min. Flash-Hook Detection Period [msec] EMS: Min Flash Hook Time CLI: min-flash-hook-time [MinFlashHookTime]	Defines the minimum time (in msec) for detection of a hook-flash event. Detection is guaranteed for hook-flash periods of at least 60 msec (when setting the minimum time to 25). Hook-flash signals that last a shorter period of time are ignored. The valid range is 25 to 300. The default is 300. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ It's recommended to reduce the detection time by 50 msec from the desired value. For example, if you want to set the value to 200 msec, then enter 150 msec (i.e., 200 minus 50).
Web: Max. Flash-Hook Detection Period [msec] EMS: Flash Hook Period CLI: flash-hook-period [FlashHookPeriod]	Defines the hook-flash period (in msec) for both Tel and IP sides (per device). For the IP side, it defines the hook-flash period that is reported to the IP. For the analog side, it defines the following: ▪ FXS interfaces: ✓ Maximum hook-flash detection period. A longer signal is considered an off-hook or on-hook event. ✓ Hook-flash generation period upon detection of a SIP INFO message containing a hook-flash signal. ▪ FXO interfaces: Hook-flash generation period. The valid range is 25 to 3,000. The default is 700. **Notes:** ▪ For FXO interfaces, a constant of 100 msec must be added to the required hook-flash period. For example, to generate a 450 msec hook-flash, set this parameter to 550. ▪ This parameter can also be configured in a Tel Profile.
DTMF Parameters	
EMS: Use End of DTMF CLI: notify-on-sig-end **[MGCPDTMFDetectionPoint]**	Determines when the detection of DTMF events is notified. ▪ **[0]** = DTMF event is reported at the end of a detected DTMF digit. ▪ **[1]** = (Default) DTMF event is reported at the start of a detected DTMF digit.
Web: Declare RFC 2833 in SDP EMS: Rx DTMF Option CLI: rfc-2833-in-sdp **[RxDTMFOption]**	Global parameter that enables the device to declare the RFC 2833 'telephony-event' parameter in the SDP. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_RxDTMFOption). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web/EMS: 1st Tx DTMF Option / 2nd Tx DTMF Option CLI: configure voip > gw dtmf-and-suppl dtmf-and-dialing > dtmf-options **[TxDTMFOption]**	Defines up to two preferred transmit DTMF negotiation methods. ▪ **[0]** Not Supported = (Default) No negotiation - DTMF digits are sent according to the parameters DTMFTransportType and RFC2833PayloadType. ▪ **[1]** INFO (Nortel) = Sends DTMF digits according to IETF Internet-Draft draft-choudhuri-sip-info-digit-00. ▪ **[2]** NOTIFY = Sends DTMF digits according to IETF Internet-Draft draft-mahy-sipping-signaled-digits-01.

Parameter	Description
	- **[3]** INFO (Cisco) = Sends DTMF digits according to Cisco format. - **[4]** RFC 2833. - **[5]** INFO (Korea) = Sends DTMF digits according to Korea Telecom format. The format of the ini file table parameter is as follows: [TxDTMFOption] FORMAT TxDTMFOption_Index = TxDTMFOption_Type; [\TxDTMFOption] For example: TxDTMFOption 0 = 1; TxDTMFOption 1 = 3; **Notes:** - DTMF negotiation methods are prioritized according to the order of their appearance. - When out-of-band DTMF transfer is used ([1], [2], [3], or [5]), the parameter DTMFTransportType is automatically set to 0 (DTMF digits are erased from the RTP stream). - When RFC 2833 (4) is selected, the device: **a.** Negotiates RFC 2833 payload type using local and remote SDPs. **b.** Sends DTMF packets using RFC 2833 payload type according to the payload type in the received SDP. **c.** Expects to receive RFC 2833 packets with the same payload type as configured by the parameter RFC2833PayloadType. **d.** Removes DTMF digits in transparent mode (as part of the voice stream). - When TxDTMFOption is set to 0, the RFC 2833 payload type is set according to the parameter RFC2833PayloadType for both transmit and receive. - If an ISDN phone user presses digits (e.g., for interactive voice response / IVR applications such as retrieving voice mail messages), ISDN Information messages received by the device for each digit are sent in the voice channel to the IP network as DTMF signals, according to the settings of the TxDTMFOption parameter. - You can also configure this functionality per specific calls, using IP Profiles (IpProfile_FirstTxDtmfOption and IpProfile_SecondTxDtmfOption). For configuring IP Profiles, see "Configuring IP Profiles" on page 314.
[DisableAutoDTMFMute]	Enables the automatic muting of DTMF digits when out-of-band DTMF transmission is used. - **[0]** = (Default) Automatic mute is used. - **[1]** = No automatic mute of in-band DTMF. When this parameter is set to 1, the DTMF transport type is set according to the parameter DTMFTransportType and the DTMF digits aren't muted if out-of-band DTMF mode is selected (TxDTMFOption set to 1, 2 or 3). This enables the sending of DTMF digits in-band (transparent of RFC 2833) in addition to out-of-band DTMF messages. **Note:** Usually this mode is not recommended.

Parameter	Description
Web/EMS: Enable Digit Delivery to IP CLI: digit-delivery-2ip **[EnableDigitDelivery2IP]**	Enables the Digit Delivery feature whereby DTMF digits are sent to the destination IP address after the Tel-to-IP call is answered. ▪ **[0]** Disable (default). ▪ **[1]** Enable = Enable digit delivery to IP. To enable this feature, modify the called number to include at least one 'p' character. The device uses the digits before the 'p' character in the initial INVITE message. After the call is answered, the device waits for the required time (number of 'p' multiplied by 1.5 seconds), and then sends the rest of the DTMF digits using the method chosen (in-band or out-of-band). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The called number can include several 'p' characters (1.5 seconds pause), for example, 1001pp699, 8888p9p300.
Web: Enable Digit Delivery to Tel EMS: Enable Digit Delivery CLI: digit-delivery-2tel **[EnableDigitDelivery]**	Enables the Digit Delivery feature, which sends DTMF digits of the called number to the device's port (analog)/B-channel (digital) (phone line) after the call is answered (i.e., line is off-hooked for FXS, or seized for FXO) for IP-to-Tel calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable For digital interfaces: If the called number in IP-to-Tel call includes the characters 'w' or 'p', the device places a call with the first part of the called number (before 'w' or 'p') and plays DTMF digits after the call is answered. If the character 'w' is used, the device waits for detection of a dial tone before it starts playing DTMF digits. For example, if the called number is '1007766p100', the device places a call with 1007766 as the destination number, then after the call is answered it waits 1.5 seconds ('p') and plays the rest of the number (100) as DTMF digits. Additional examples: 1664wpp102, 66644ppp503, and 7774w100pp200. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For analog interfaces: The called number can include characters 'p' (1.5 seconds pause) and 'd' (detection of dial tone). If character 'd' is used, it must be the first 'digit' in the called number. The character 'p' can be used several times. For example (for FXS/FXO interfaces), the called number can be as follows: d1005, dpp699, p9p300. To add the 'd' and 'p' digits, use the usual number manipulation rules. ▪ For analog interfaces: To use this feature with FXO interfaces, configure the device to operate in one-stage dialing mode. ▪ If this parameter is enabled, it is possible to configure the FXS/FXO interface to wait for dial tone per destination phone number (before or during dialing of destination phone number). Therefore, the parameter IsWaitForDialTone (configurable for the entire device) is ignored. ▪ For analog interfaces: The FXS interface send SIP 200 OK responses only after the DTMF dialing is complete. ▪ This parameter can also be configured in a Tel Profile.
CLI: replace-nb-sign-w-esc	Determines whether to replace the number sign (#) with the

Parameter	Description
[ReplaceNumberSignWithEscapeChar]	escape character (%23) in outgoing SIP messages for Tel-to-IP calls. ▪ [0] Disable (default). ▪ [1] Enable = All number signs #, received in the dialed DTMF digits are replaced in the outgoing SIP Request-URI and To headers with the escape sign %23. **Notes:** ▪ This parameter is applicable only if the parameter IsSpecialDigits is set 1. ▪ This parameter is applicable only to analog interfaces.
Web: Special Digit Representation EMS: Use Digit For Special DTMF CLI: special-digit-rep **[UseDigitForSpecialDTMF]**	Defines the representation for 'special' digits ('*' and '#') that are used for out-of-band DTMF signaling (using SIP INFO/NOTIFY). ▪ **[0]** Special = (Default) Uses the strings '*' and '#'. ▪ **[1]** Numeric = Uses the numerical values 10 and 11.

67.10.3 Digit Collection and Dial Plan Parameters

The digit collection and dial plan parameters are described in the table below.

Table 67-43: Digit Collection and Dial Plan Parameters

Parameter	Description
Web/EMS: Dial Plan Index CLI: dial-plan-index **[DialPlanIndex]**	Defines the Dial Plan index to use in the external Dial Plan file. The Dial Plan file is loaded to the device as a .dat file (converted using the DConvert utility). The Dial Plan index can be defined globally or per Tel Profile. The valid value range is 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1, indicating that no Dial Plan file is used. **Notes:** ▪ If this parameter is configured to select a Dial Plan index, the settings of the parameter DigitMapping are ignored. ▪ If this parameter is configured to select a Dial Plan index from an external Dial Plan file, the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map rules configured by the DigitMapping parameter. ▪ This parameter is applicable also to ISDN with overlap dialing. ▪ For E1 CAS MFC-R2 variants (which don't support terminating digit for the called party number, usually I-15), this parameter and the DigitMapping parameter are ignored. Instead, you can define a Dial Plan template per trunk using the parameter CasTrunkDialPlanName_x (or in the Trunk Settings page). ▪ This parameter can also be configured in a Tel Profile. ▪ For more information on the Dial Plan file, see "Dialing Plans for Digit Collection" on page 606.
CLI: tel2ip-src-nb-map-dial-index **[Tel2IPSourceNumberMappingDi**	Defines the Dial Plan index in the external Dial Plan file for the Tel-to-IP Source Number Mapping feature.

Parameter	Description
alPlanIndex]	The valid value range is 0 to 7, defining the Dial Plan index [Plan x] in the Dial Plan file. The default is -1 (disabled). For more information on this feature, see "Modifying ISDN-to-IP Calling Party Number" on page 613.
Web: Digit Mapping Rules EMS: Digit Map Pat terns CLI: default-dm **[DigitMapping]**	Defines the digit map pattern (used to reduce the dialing period when ISDN overlap dialing for digital interfaces). If the digit string (i.e., dialed number) matches one of the patterns in the digit map, the device stops collecting digits and establishes a call with the collected number. The digit map pattern can contain up to 52 options (rules), each separated by a vertical bar (|). The maximum length of the entire digit pattern is 152 characters. The available notations include the following: ▪ **[n-m]:** Range of numbers (not letters). ▪ **.** (single dot): Repeat digits until next notation (e.g., T). ▪ **x:** Any single digit. ▪ **T:** Dial timeout (configured by the TimeBetweenDigits parameter). ▪ **S:** Short timer (configured by the TimeBetweenDigits parameter; default is two seconds) that can be used when a specific rule is defined after a more general rule. For example, if the digit map is 99|998, then the digit collection is terminated after the first two 9 digits are received. Therefore, the second rule of 998 can never be matched. But when the digit map is 99s|998, then after dialing the first two 9 digits, the device waits another two seconds within which the caller can enter the digit 8. An example of a digit map is shown below: 11xS|00T|[1-7]xxx|8xxxxxxx|#xxxxxxx|*xx|91xxxxxxxxxx|9011x.T In the example above, the last rule can apply to International numbers: 9 for dialing tone, 011 Country Code, and then any number of digits for the local number ('x.'). **Notes:** ▪ For ISDN interfaces, the digit map mechanism is applicable only when ISDN overlap dialing is used (ISDNRxOverlap is set to 1). ▪ If the DialPlanIndex parameter is configured (to select a Dial Plan index), then the device first attempts to locate a matching digit pattern in the Dial Plan file, and if not found, then attempts to locate a matching digit pattern in the Digit Map rules configured by the DigitMapping parameter. ▪ For more information on digit mapping, see "Digit Mapping" on page 422.
Web: Max Digits in Phone Num EMS: Max Digits in Phone Number CLI: mxdig-b4-dialing **[MaxDigits]**	Defines the maximum number of collected destination number digits that can be received (i.e., dialed) from the Tel side (analog) or for digital, when ISDN Tel-to-IP overlap dialing is performed. When the number of collected digits reaches this maximum, the device uses these digits for the called destination number. The valid range is 1 to 49. The default is 5 for analog and 30for digital.

Parameter	Description
	Notes: ▪ Instead of using this parameter, Digit Mapping rules can be configured. ▪ For FXS/FXO interfaces: Dialing ends when any of the following scenarios occur: ✓ Maximum number of digits is dialed ✓ Interdigit Timeout (TimeBetweenDigits) expires ✓ Pound (#) key is pressed ✓ Digit map pattern is matched
Web: Inter Digit Timeout for Overlap Dialing [sec] EMS: Interdigit Timeout (Sec) CLI: time-btwn-dial-digs **[TimeBetweenDigits]**	Analog: Defines the time (in seconds) that the device waits between digits that are dialed by the user. ISDN overlap dialing: Defines the time (in seconds) that the device waits between digits that are received from the PSTN or IP during overlap dialing. When this inter-digit timeout expires, the device uses the collected digits to dial the called destination number. The valid range is 1 to 10. The default is 4.
Web: Enable Special Digits EMS: Use '#' For Dial Termination CLI: special-digits [IsSpecialDigits]	Determines whether the asterisk (*) and pound (#) digits can be used in DTMF. ▪ [0] Disable = Use '*' or '#' to terminate number collection (refer to the parameter UseDigitForSpecialDTMF). (Default.) ▪ [1] Enable = Allows '*' and '#' for telephone numbers dialed by a user or for the endpoint telephone number. **Notes:** ▪ The symbols can always be used as the first digit of a dialed number even if you disable this parameter. ▪ This parameter is applicable only to FXS and FXO interfaces.

67.10.4 Voice Mail Parameters

The voice mail parameters are described in the table below. For more information on the Voice Mail application, refer to the *CPE Configuration Guide for Voice Mail*.

Table 67-44: Voice Mail Parameters

Parameter	Description
Web/EMS: Voice Mail Interface CLI: vm-interface **[VoiceMailInterface]**	Enables the device's Voice Mail application and determines the communication method between the device and PBX. ▪ **[0]** None (default) ▪ **[1]** DTMF ▪ **[2]** SMDI ▪ [3] QSIG ▪ [4] SETUP Only = Applicable only to ISDN. ▪ [5] MATRA/AASTRA QSIG ▪ [6] QSIG SIEMENS = QSIG MWI activate and deactivate messages include Siemens Manufacturer Specific Information (MSI) ▪ [7] IP2IP = The device's IP-to-IP application is used for interworking between an IP Voice Mail server and the device.

Parameter	Description
	This is implemented for sending unsolicited SIP NOTIFY messages received from the Voice Mail server to an IP Group (configured using the NotificationIPGroupID parameter). ▪ [8] ETSI = Euro ISDN, according to ETS 300 745-1 V1.2.4, section 9.5.1.1. Enables MWI interworking from IP to Tel, typically used for BRI phones. ▪ [9] = ISDN PRI trunks set to NI-2. This is used for interworking the SIP Message Waiting Indication (MWI) NOTIFY message to ISDN PRI NI-2 Message Waiting Notification (MWN) that is sent in the ISDN Facility IE message. This option is applicable when the device is connected to a PBX through an ISDN PRI trunk configured to NI-2. **Note:** To disable voice mail per Trunk Group, you can use a Tel Profile with the EnableVoiceMailDelay parameter set to disabled (0). This eliminates the phenomenon of call delay on Trunks not implementing voice mail when voice mail is enabled using this global parameter.
Web: Enable VoiceMail URI EMS: Enable VMURI CLI: voicemail-uri **[EnableVMURI]**	Enables the interworking of target and cause for redirection from Tel to IP and vice versa, according to RFC 4468. ▪ **[0]** Disable (default) ▪ **[1]** Enable Upon receipt of an ISDN Setup message with Redirect values, the device maps the Redirect phone number to the SIP 'target' parameter and the Redirect number reason to the SIP 'cause' parameter in the Request-URI. Redirecting Reason >> SIP Response Code Unknown >> 404 User busy >> 486 No reply >> 408 Deflection >> 487/480 Unconditional >> 302 Others >> 302 If the device receives a Request-URI that includes a 'target' and 'cause' parameter, the 'target' is mapped to the Redirect phone number and the 'cause' is mapped to the Redirect number reason.
[WaitForBusyTime]	Defines the time (in msec) that the device waits to detect busy and/or reorder tones. This feature is used for semi-supervised PBX call transfers (i.e., the LineTransferMode parameter is set to 2). The valid value range is 0 to 20000 (i.e., 20 sec). The default is 2000 (i.e., 2 sec).
Web/EMS: Line Transfer Mode CLI: line-transfer-mode **[LineTransferMode]**	Defines the call transfer method used by the device. This parameter is applicable to FXO call transfer and E1/T1 CAS call transfer if the TrunkTransferMode_x parameter is set to 3 (CAS Normal) or 1 (CAS NFA). ▪ **[0]** None = (Default) IP. ▪ **[1]** Blind = PBX blind transfer: ✓ Analog (FXO): After receiving a SIP REFER message from the IP side, the device (FXO) sends a hook-flash to

Parameter	Description
	the PBX, dials the digits (that are received in the Refer-To header), and then immediately releases the line (i.e., on-hook). The PBX performs the transfer internally. ✓ E1/T1 CAS: When a SIP REFER message is received, the device performs a blind transfer, by performing a CAS wink, waiting a user-defined time (configured by the WaitForDialTime parameter), dialing the Refer-To number, and then releasing the call. The PBX performs the transfer internally. ▪ **[2]** Semi Supervised = PBX semi-supervised transfer: ✓ Analog (FXO): After receiving a SIP REFER message from the IP side, the device sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). If no busy or reorder tones are detected (within the user-defined interval set by the WaitForBusyTime parameter), the device completes the call transfer by releasing the line. If these tones are detected, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected), and generates an additional hook-flash toward the FXO line to restore connection to the original call. ✓ E1/T1 CAS: The device performs a CAS wink, waits a user-defined time (configured by the WaitForDialTime parameter), and then dials the Refer-To number. If during the user-defined interval set by the WaitForBusyTime parameter, no busy or reorder tones are detected, the device completes the call transfer by releasing the line. If during this interval, the device detects these tones, the transfer operation is cancelled, the device sends a SIP NOTIFY message with a failure reason (e.g., 486 if a busy tone is detected), and then generates an additional wink toward the CAS line to restore connection with the original call. ▪ **[3]** Supervised = PBX Supervised transfer: ✓ Analog (FXO): After receiving a SIP REFER message from the IP side, the device sends a hook-flash to the PBX, and then dials the digits (that are received in the Refer-To header). The device waits for connection of the transferred call and then completes the call transfer by releasing the line. If speech is not detected, the transfer is cancelled, the device sends a SIP NOTIFY message with a failure reason in the NOTIFY body (such as 486 if busy tone detected) and generates an additional hook-flash toward the FXO line to restore connection to the original call. ✓ E1/T1 CAS: The device performs a supervised transfer to the PBX. The device performs a CAS wink, waits a user-defined time (configured by the WaitForDialTime parameter), and then dials the Refer-To number. The device completes the call transfer by releasing the line only after detection of the transferred party answer. To enable answer supervision, you also need to do the following: 1) Enable voice detection (i.e., set the EnableVoiceDetection parameter to 1). 2) Set the EnableDSPIPMDetectors parameter to 1.

Parameter	Description
	3) Install the IPMDetector DSP option Feature License Key.
SMDI Parameters	
Web/EMS: Enable SMDI [SMDI]	Enables Simplified Message Desk Interface (SMDI) interface on the device. ▪ [0] Disable = (Default) Normal serial ▪ [1] Enable (Bellcore) ▪ [2] Ericsson MD-110 ▪ [3] NEC (ICS) **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When the RS-232 connection is used for SMDI messages (Serial SMDI), it cannot be used for other applications, for example, to access the Command Line Interface (CLI).
Web/EMS: SMDI Timeout [SMDITimeOut]	Defines the time (in msec) that the device waits for an SMDI Call Status message before or after a Setup message is received. This parameter synchronizes the SMDI and analog CAS interfaces. If the timeout expires and only an SMDI message is received, the SMDI message is dropped. If the timeout expires and only a Setup message is received, the call is established. The valid range is 0 to 10000 (i.e., 10 seconds). The default is 2000.
Message Waiting Indication (MWI) Parameters	
Web: MWI Off Digit Pattern EMS: MWI Off Code CLI: mwi-off-dig-ptrn **[MWIOffCode]**	Defines the digit code used by the device to notify the PBX that there are no messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
Web: MWI On Digit Pattern EMS: MWI On Code CLI: mwi-on-dig-ptrn **[MWIOnCode]**	Defines the digit code used by the device to notify the PBX of messages waiting for a specific extension. This code is added as prefix to the dialed number. The valid range is a 25-character string.
Web: MWI Suffix Pattern EMS: MWI Suffix Code CLI: mwi-suffix-pattern **[MWISuffixCode]**	Defines the digit code used by the device as a suffix for 'MWI On Digit Pattern' and 'MWI Off Digit Pattern'. This suffix is added to the generated DTMF string after the extension number. The valid range is a 25-character string.
Web: MWI Source Number EMS: MWI Source Name CLI: mwi-source-number **[MWISourceNumber]**	Defines the calling party's phone number used in the Q.931 MWI Setup message to PSTN. If not configured, the channel's phone number is used as the calling number.
CLI: mwi-subs-ipgrpid **[MWISubscribeIPGroupID]**	Defines the IP Group ID used when subscribing to an MWI server. The 'The SIP Group Name' field value of the IP Group table is used as the Request-URI host name in the outgoing MWI SIP SUBSCRIBE message. The request is sent to the IP address defined for the Proxy Set that is associated with the IP Group. The Proxy Set's capabilities such as proxy redundancy and load balancing are also applied to the message. For example, if the 'SIP Group Name' field of the IP Group is set to "company.com", the device sends the following SUBSCRIBE

Parameter	Description

	message:
	`SUBSCRIBE sip:company.com...`
	Instead of:
	`SUBSCRIBE sip:10.33.10.10...`
	Note: If this parameter is not configured, the MWI SUBSCRIBE message is sent to the MWI server as defined by the MWIServerIP parameter.

[NotificationIPGroupID]	Defines the IP Group ID to which the device sends SIP NOTIFY MWI messages. **Notes:** ▪ This is used for MWI Interrogation. For more information on the interworking of QSIG MWI to IP, see Message Waiting Indication on page 436. ▪ To determine the handling method of MWI Interrogation messages, use the TrunkGroupSettings_MWIInterrogationType, parameter (in the Trunk Group Settings table).
[MWIQsigMsgCentreldIDPartyNumber]	Defines the Message Centred ID party number used for QSIG MWI messages. If not configured (default), the parameter is not included in MWI (activate and deactivate) QSIG messages. The valid value is a string.
Digit Patterns The following digit pattern parameters apply only to voice mail applications that use the DTMF communication method. For the available pattern syntaxes, refer to the *CPE Configuration Guide for Voice Mail*.	
Web: Forward on Busy Digit Pattern (Internal) EMS: Digit Pattern Forward On Busy CLI: fwd-bsy-dig-ptrn-int **[DigitPatternForwardOnBusy]**	Defines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on No Answer Digit Pattern (Internal) EMS: Digit Pattern Forward On No Answer CLI: fwd-no-ans-dig-pat-int **[DigitPatternForwardOnNoAnswer]**	Defines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (Internal) EMS: Digit Pattern Forward On DND CLI: fwd-dnd-dig-ptrn-int **[DigitPatternForwardOnDND]**	Defines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an internal extension. The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (Internal) EMS: Digit Pattern Forward No Reason CLI: fwd-no-rsn-dig-ptrn-int **[DigitPatternForwardNoReason]**	Defines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an internal extension. The valid range is a 120-character string.

Parameter	Description
Web: Forward on Busy Digit Pattern (External) EMS: VM Digit Pattern On Busy External CLI: fwd-bsy-dig-ptrn-ext **[DigitPatternForwardOnBusyExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on busy' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on No Answer Digit Pattern (External) EMS: VM Digit Pattern On No Answer Ext CLI: fwd-no-ans-dig-pat-ext **[DigitPatternForwardOnNoAnswerExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on no answer' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on Do Not Disturb Digit Pattern (External) EMS: VM Digit Pattern On DND External CLI: fwd-dnd-dig-ptrn-ext **[DigitPatternForwardOnDNDExt]**	Defines the digit pattern used by the PBX to indicate 'call forward on do not disturb' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Forward on No Reason Digit Pattern (External) EMS: VM Digit Pattern No Reason External CLI: fwd-no-rsn-dig-ptrn-ext **[DigitPatternForwardNoReasonExt]**	Defines the digit pattern used by the PBX to indicate 'call forward with no reason' when the original call is received from an external line (not an internal extension). The valid range is a 120-character string.
Web: Internal Call Digit Pattern EMS: Digit Pattern Internal Call CLI: int-call-dig-ptrn **[DigitPatternInternalCall]**	Defines the digit pattern used by the PBX to indicate an internal call. The valid range is a 120-character string.
Web: External Call Digit Pattern EMS: Digit Pattern External Call CLI: ext-call-dig-ptrn **[DigitPatternExternalCall]**	Defines the digit pattern used by the PBX to indicate an external call. The valid range is a 120-character string.
Web: Disconnect Call Digit Pattern EMS: Tel Disconnect Code CLI: disc-call-dig-ptrn **[TelDisconnectCode]**	Defines a digit pattern that when received from the Tel side, indicates the device to disconnect the call. The valid range is a 25-character string.
Web: Digit To Ignore Digit Pattern EMS: Digit To Ignore CLI: dig-to-ignore-dig-pattern **[DigitPatternDigitToIgnore]**	Defines a digit pattern that if received as Src (S) or Redirect (R) numbers is ignored and not added to that number. The valid range is a 25-character string.

67.10.5 Supplementary Services Parameters

This subsection describes the device's supplementary telephony services parameters.

67.10.5.1 Caller ID Parameters

The caller ID parameters are described in the table below.

Table 67-45: Caller ID Parameters

Parameter	Description
Caller ID Permissions Table	
Web: Caller ID Permissions Table EMS: SIP Endpoints > Caller ID CLI: configure voip > gw analoggw enable-caller-id [EnableCallerID]	This table parameter enables (per port) Caller ID generation (for FXS interfaces) and detection (for FXO interfaces). The format of the ini file table parameter is as follows: [EnableCallerID] FORMAT EnableCallerID_Index = EnableCallerID_IsEnabled, EnableCallerID_Module, EnableCallerID_Port; [\EnableCallerID] Where, ▪ Module = Module number, where 1 denotes the module in Slot 1. ▪ Port = Port number, where 1 denotes Port 1 of a module. For example: EnableCallerID 0 = 1,3,1; (caller ID enabled on Port 1 of Module 3) EnableCallerID 1 = 0,3,2; (caller ID disabled on Port 2 of Module 3) For a detailed description of this table, see Configuring Caller ID Permissions on page 483. **Notes:** ▪ The indexing of this parameter starts at 0. ▪ This parameter is applicable only to FXS and FXO interfaces.
Caller Display Information Table	
Web: Caller Display Information Table EMS: SIP Endpoints > Caller ID CLI: configure voip > gw analoggw caller-display-info [CallerDisplayInfo]	This table parameter enables the device to send Caller ID information to the IP side when a call is made. The called party can use this information for caller identification. The information configured in this table is sent in the SIP INVITE message's From header. The format of the ini file table parameter is as follows: [CallerDisplayInfo] FORMAT CallerDisplayInfo_Index = CallerDisplayInfo_DisplayString, CallerDisplayInfo_IsCidRestricted, CallerDisplayInfo_Module, CallerDisplayInfo_Port; [\CallerDisplayInfo] Where, ▪ Module = Module number, where 1 denotes the module in Slot 1. ▪ Port = Port number, where 1 denotes Port 1 of a module. For example:

Parameter	Description
	CallerDisplayInfo 0 = Susan C.,0,1,1; ("Susan C." is sent as the Caller ID for Port 1 of Module 1) CallerDisplayInfo 1 = Mark M.,0,1,2; ("Mark M." is sent as Caller ID for Port 2 of Module 1) For a detailed description of this table, see Configuring Caller Display Information on page 480. **Notes:** ▪ The indexing of this table ini file parameter starts at 0. ▪ This parameter is applicable only to FXS and FXO interfaces.
Web/EMS: Enable Caller ID CLI: enable-caller-id [EnableCallerID]	Global parameter that enables Caller ID. ▪ **[0]** Disable (default) ▪ **[1]** Enable = ✓ FXS: The calling number and display text (from IP) are sent to the device's port. ✓ FXO or CAS: The device detects the Caller ID signal received from the Tel and sends it to the IP in the SIP INVITE message (as the 'Display' element). To configure the Caller ID string per port, see Configuring Caller Display Information on page 480. To enable or disable caller ID generation / detection per port, see Configuring Caller ID Permissions on page 483.
Web: Caller ID Type EMS: Caller id Types CLI: caller-ID-type [CallerIDType]	Determines the standard used for detection (FXO) and generation (FXS) of Caller ID, and detection (FXO) / generation (FXS) of MWI (when specified) signals: ▪ [0] Standard Bellcore = (Default) Caller ID and MWI ▪ [1] Standard ETSI = Caller ID and MWI ▪ [2] Standard NTT ▪ [4] Standard BT = Britain ▪ [16] Standard DTMF Based ETSI ▪ [17] Standard Denmark = Caller ID and MWI ▪ [18] Standard India ▪ [19] Standard Brazil **Notes:** ▪ This parameter is applicable only to FXS and FXO interfaces. ▪ Typically, the Caller ID signals are generated / detected between the first and second rings. However, sometimes the Caller ID is detected before the first ring signal. In such a scenario, set the RingsBeforeCallerID parameter to 0. ▪ Caller ID detection for Britain [4] is not supported on the device’s FXO ports. Only FXS ports can generate the Britain [4] Caller ID. ▪ To select the Bellcore Caller ID sub standard, use the BellcoreCallerIDTypeOneSubStandard parameter. To select the ETSI Caller ID substandard, use the ETSICallerIDTypeOneSubStandard parameter. ▪ To select the Bellcore MWI sub standard, use the BellcoreVMWITypeOneStandard parameter. To select the ETSI MWI sub standard, use the

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>ETSIVMWITypeOneStandard parameter.<ul><li>If you define Caller ID Type as NTT [2], you need to define the NTT DID signaling form (FSK or DTMF) using the NTTDIDSignallingForm parameter.</li></ul></td></tr>
<tr><td>Web: Enable FXS Caller ID Category Digit For Brazil Telecom
CLI: fxs-callid-cat-brazil
[AddCPCPrefix2BrazilCallerID]</td><td>Enables the interworking of Calling Party Category (cpc) code from SIP INVITE messages to FXS Caller ID first digit.<ul><li>[0] Disable (default)</li><li>[1] Enable</li></ul>When this parameter is enabled, the device sends the Caller ID number (calling number) with the cpc code (received in the SIP INVITE message) to the device's FXS port. The cpc code is added as a prefix to the caller ID (after IP-to-Tel calling number manipulation). For example, assuming that the incoming INVITE contains the following From (or P-Asserted-Id) header:<pre>From:&lt;sip:+551137077801;cpc=payphone@10.20.7.35>;tag=53700</pre>The calling number manipulation removes "+55" (leaving 10 digits), and then adds the prefix 7, the cpc code for payphone user. Therefore, the Caller ID number that is sent to the FXS port, in this example is 71137077801.
If the incoming INVITE message doesn't contain the 'cpc' parameter, nothing is added to the Caller ID number.
<table>
<tr><th>CPC Value in Received INVITE</th><th>CPC Code Prefixed to Caller ID (Sent to FXS Endpoint)</th><th>Description</th></tr>
<tr><td>cpc=unknown</td><td>1</td><td>Unknown user</td></tr>
<tr><td>cpc=subscribe</td><td>1</td><td>-</td></tr>
<tr><td>cpc=ordinary</td><td>1</td><td>Ordinary user</td></tr>
<tr><td>cpc=priority</td><td>2</td><td>Pre-paid user</td></tr>
<tr><td>cpc=test</td><td>3</td><td>Test user</td></tr>
<tr><td>cpc=operator</td><td>5</td><td>Operator</td></tr>
<tr><td>cpc=data</td><td>6</td><td>Data call</td></tr>
<tr><td>cpc=payphone</td><td>7</td><td>Payphone user</td></tr>
</table>
<b>Notes:</b><ul><li>This parameter is applicable only to FXS interfaces.</li><li>For this parameter to be enabled, you must also set the parameter EnableCallingPartyCategory to 1.</li></ul></td></tr>
<tr><td>[EnableCallerIDTypeTwo]</td><td>Disables the generation of Caller ID type 2 when the phone is off-hooked. Caller ID type 2 (also known as off-hook Caller ID) is sent to a currently busy telephone to display the caller ID of the waiting call.<ul><li>[0] = Caller ID type 2 isn't played.</li><li>[1] = (Default) Caller ID type 2 is played.</li><li>Note: This parameter is applicable only to FXS interfaces.</li></ul></td></tr>
<tr><td>EMS: Caller ID Timing Mode
CLI: caller-id-timing-mode</td><td>Determines when Caller ID is generated.<ul><li>[0] = (Default) Caller ID is generated between the first two</li></ul></td></tr>
</table>

Parameter	Description
[AnalogCallerIDTimingMode]	rings. - [1] = The device attempts to find an optimized timing to generate the Caller ID according to the selected Caller ID type. **Notes:** - This parameter is applicable only to FXS interfaces. - If this parameter is set to 1 and used with distinctive ringing, the Caller ID signal doesn't change the distinctive ringing timing. - For this parameter to take effect, a device reset is required.
EMS: Bellcore Caller ID Type One Sub Standard CLI: bellcore-callerid-type-one-sub-standard [BellcoreCallerIDTypeOneSubStandard]	Determines the Bellcore Caller ID sub-standard. - [0] = (Default) Between rings. - [1] = Not ring related. **Notes:** - For this parameter to take effect, a device reset is required. - This parameter is applicable only to FXS interfaces.
EMS: ETSI Caller ID Type One Sub Standard CLI: etsi-callerid-type-one-sub-standard [ETSICallerIDTypeOneSubStandard]	Determines the ETSI FSK Caller ID Type 1 sub-standard (FXS only). - [0] = (Default) ETSI between rings. - [1] = ETSI before ring DT_AS. - [2] = ETSI before ring RP_AS. - [3] = ETSI before ring LR_DT_AS. - [4] = ETSI not ring related DT_AS. - [5] = ETSI not ring related RP_AS. - [6] = ETSI not ring related LR_DT_AS. **Notes:** - For this parameter to take effect, a device reset is required. - This parameter is applicable only to FXS interfaces.
Web: Asserted Identity Mode EMS: Asserted ID Mode CLI: asserted-identity-m **[AssertedIdMode]**	Determines whether the SIP header P-Asserted-Identity or P-Preferred-Identity is added to the sent INVITE, 200 OK, or UPDATE request for Caller ID (or privacy). These headers are used to present the calling party's Caller ID, which is composed of a Calling Number and a Calling Name (optional). - **[0]** Disabled = (Default) P-Asserted-Identity and P-Preferred-Identity headers are not added. - **[1]** Add P-Asserted-Identity - **[2]** Add P-Preferred-Identity The used header also depends on the calling Privacy (allowed or restricted). These headers are used together with the Privacy header. If Caller ID is restricted (i.e., P-Asserted-Identity is not sent), the Privacy header includes the value 'id' ('Privacy: id'). Otherwise, for allowed Caller ID, 'Privacy: none' is used. If Caller ID is restricted (received from Tel or configured in the device), the From header is set to <anonymous@anonymous.invalid>. The 200 OK response can contain the connected party CallerID - Connected Number and Connected Name. For example, if the call is answered by the device, the 200 OK response includes the P-Asserted-Identity with Caller ID. The device interworks (in some ISDN variants), the Connected Party number and name

Parameter	Description
	from Q.931 Connect message to SIP 200 OK with the P-Asserted-Identity header. In the opposite direction, if the ISDN device receives a 200 OK with P-Asserted-Identity header, it interworks it to the Connected party number and name in the Q.931 Connect message, including its privacy.
Web/EMS: Use Destination As Connected Number **[UseDestinationAsConnectedNumber]**	Enables the device to include the Called Party Number, from outgoing Tel calls (after number manipulation), in the SIP P-Asserted-Identity header. The device includes the SIP P-Asserted-Identity header in 180 Ringing and 200 OK responses for IP-to-Tel calls. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this feature to function, you also need to enable the device to include the P-Asserted-Identity header in 180/200 OK responses, by setting the AssertedIDMode parameter to **Add P-Asserted-Identity.** ▪ If the received Q.931 Connect message contains a Connected Party Number, this number is used in the P-Asserted-Identity header in 200 OK response. ▪ This parameter is applicable to ISDN, CAS, and/or FXO interfaces.
Web: Caller ID Transport Type EMS: Transport Type CLI: caller-ID-transport-type **[CallerIDTransportType]**	Determines the device's behavior for Caller ID detection. ▪ **[0]** Disable = The caller ID signal is not detected - DTMF digits remain in the voice stream. ▪ **[1]** Relay = (Currently not applicable.) ▪ **[3]** Mute = (Default) The caller ID signal is detected from the Tel side and then erased from the voice stream. **Note:** Caller ID detection is applicable only to FXO interfaces.
Reject Anonymous Calls Per Port Table	
CLI: configure voip > gw analoggw reject-anonymous-calls [RejectAnonymousCallPerPort]	This table parameter determines whether the device rejects incoming anonymous calls per FXS port. If enabled, when a device's FXS interface receives an anonymous call, it rejects the call and responds with a SIP 433 (Anonymity Disallowed) response. The format of the ini file table parameter is as follows: [RejectAnonymousCallPerPort] FORMAT RejectAnonymousCallPerPort_Index = RejectAnonymousCallPerPort_Enable, RejectAnonymousCallPerPort_Port, RejectAnonymousCallPerPort_Module; [\RejectAnonymousCallPerPort] Where, ▪ Enable = accept [0] (default) or reject [1] incoming anonymous calls. ▪ Port = Port number. ▪ Module = Module number. For example: RejectAnonymousCallPerPort 0 = 0,1,1; RejectAnonymousCallPerPort 1 = 1,2,1; **Note:** This parameter is applicable only to FXS interfaces.

67.10.5.2 Call Waiting Parameters

The call waiting parameters are described in the table below.

Table 67-46: Call Waiting Parameters

Parameter	Description
Web/EMS: Enable Call Waiting CLI: call-waiting **[EnableCallWaiting]**	Enables the Call Waiting feature. ▪ **[0]** Disable ▪ **[1]** Enable (Default) For digital interfaces: If enabled and the device initiates a Tel-to-IP call to a destination that is busy, it plays a call waiting ringback tone to the caller. The tone is played only if the destination returns a 182 "Queued" SIP response. For FXS interface: If enabled, when an FXS interface receives a call on a busy endpoint, it responds with a 182 response (and not with a 486 busy). The device plays a call waiting indication signal. When hook-flash is detected, the device switches to the waiting call. The device that initiated the waiting call plays a call waiting ringback tone to the calling party after a 182 response is received. **Notes:** ▪ The device's Call Progress Tones (CPT) file must include a Call Waiting ringback tone (caller side) and a call waiting tone (called side, FXS only). ▪ FXS interfaces: The EnableHold parameter must be enabled on both the calling and the called side. ▪ Analog interfaces: You can use the table parameter CallWaitingPerPort to enable Call Waiting per port. ▪ Analog interfaces: For information on the Call Waiting feature, see Call Waiting on page 435.
EMS: Send 180 For Call Waiting **[Send180ForCallWaiting]**	Determines the SIP response code for indicating Call Waiting. ▪ **[0]** = (Default) Use 182 Queued response to indicate call waiting. ▪ **[1]** = Use 180 Ringing response to indicate call waiting.
Call Waiting Table	
Web: Call Waiting Table EMS: SIP Endpoints > Call Waiting CLI: configure voip > gw analoggw call-waiting [CallWaitingPerPort]	This table parameter configures call waiting per FXS port. The format of this The format of the ini file table parameter format of the ini file table parameter is as follows: [CallWaitingPerPort] FORMAT CallWaitingPerPort_Index = CallWaitingPerPort_IsEnabled, CallWaitingPerPort_Module, CallWaitingPerPort_Port; [\CallWaitingPerPort] For example: CallWaitingPerPort 0 = 0,1,1; (call waiting disabled for Port 1 of Module 1) CallWaitingPerPort 1 = 1,1,2; (call waiting enabled for Port 2 of Module 1) **Notes:** ▪ This parameter is applicable only to FXS ports. ▪ For a detailed description of this table, see Configuring Call Waiting on page 484.

Parameter	Description
Web: Number of Call Waiting Indications EMS: Call Waiting Number of Indications CLI: nb-of-cw-ind [NumberOfWaitingIndications]	Defines the number of call waiting indications that are played to the called telephone that is connected to the device for Call Waiting. The valid range is 1 to 100 indications. The default is 2. **Note:** This parameter is applicable only to FXS ports.
Web: Time Between Call Waiting Indications EMS: Call Waiting Time Between Indications CLI: time-between-cw [TimeBetweenWaitingIndications]	Defines the time (in seconds) between consecutive call waiting indications for call waiting. The valid range is 1 to 100. The default is 10. **Note:** This parameter is applicable only to FXS ports.
Web/EMS: Time Before Waiting Indications CLI: time-b4-cw-ind [TimeBeforeWaitingIndications]	Defines the interval (in seconds) before a call waiting indication is played to the port that is currently in a call. The valid range is 0 to 100. The default time is 0 seconds. **Note:** This parameter is applicable only to FXS ports.
Web/EMS: Waiting Beep Duration CLI: waiting-beep-dur [WaitingBeepDuration]	Defines the duration (in msec) of call waiting indications that are played to the port that is receiving the call. The valid range is 100 to 65535. The default is 300. **Note:** This parameter is applicable only to FXS ports.

EMS: First Call Waiting Tone ID
[FirstCallWaitingToneID]
	Defines the index of the first Call Waiting Tone in the CPT file. This feature enables the called party to distinguish between different call origins (e.g., external versus internal calls).
	There are three ways to use the distinctive call waiting tones:
	▪ Playing the call waiting tone according to the SIP Alert-Info header in the received 180 Ringing SIP response. The value of the Alert-Info header is added to the value of the FirstCallWaitingToneID parameter.
	▪ Playing the call waiting tone according to PriorityIndex in the ToneIndex table parameter.
	▪ Playing the call waiting tone according to the parameter “CallWaitingTone#' of a SIP INFO message.
	The device plays the tone received in the 'play tone CallWaitingTone#' parameter of an INFO message plus the value of this parameter minus 1.
	The valid range is -1 to 1,000. The default is -1 (i.e., not used).
	Notes:
	▪ This parameter is applicable only to analog interfaces.
	▪ It is assumed that all Call Waiting Tones are defined in sequence in the CPT file.
	▪ SIP Alert-Info header examples:
	✓ Alert-Info:<Bellcore-dr2>
	✓ Alert-Info:<http://…/Bellcore-dr2> (where "dr2" defines call waiting tone #2)
	▪ The SIP INFO message is according to Broadsoft's application server definition. Below is an example of such an INFO message:
	`INFO sip:06@192.168.13.2:5060 SIP/2.0`
	`Via:SIP/2.0/UDP 192.168.13.40:5060;branch=z9hG4bK040066422630`
	`From:`

Parameter	Description
	<sip:4505656002@192.168.13.40:5060>;tag=1455352915 To: <sip:06@192.168.13.2:5060> Call-ID:0010-0008@192.168.13.2 CSeq:342168303 INFO Content-Length:28 Content-Type:application/broadsoft play tone CallWaitingTone1

67.10.5.3 Call Forwarding Parameters

The call forwarding parameters are described in the table below.

Table 67-47: Call Forwarding Parameters

Parameter	Description
Web: Enable Call Forward CLI: call-forward **[EnableForward]**	Enables the Call Forwarding feature. ▪ **[0]** Disable ▪ **[1]** Enable (Default) For FXS interfaces, the Call Forward table (FwdInfo parameter) must be defined to use the Call Forward service. The device uses SIP REFER messages for call forwarding. **Notes:** ▪ To use this service, the devices at both ends must support this option. ▪ For the device to respond to SIP 3xx responses with a new SIP request (forwarding the original request), set this parameter to **Enable**.
Call Forwarding Table	
Web: Call Forwarding Table EMS: Analog Gateway Provisioning > Tab: Call Forward CLI: configure voip > gw analoggw call-forward [FwdInfo]	This table parameter configures call forwarding of IP-to-Tel calls (using SIP 302 response) to other device ports or an IP destination, based on the device's port to which the call was originally routed. The format of the ini file table parameter is as follows: [FwdInfo] FORMAT FwdInfo_Index = FwdInfo_Type, FwdInfo_Destination, FwdInfo_NoReplyTime, FwdInfo_Module, FwdInfo_Port; [\FwdInfo] Where, ▪ Module = Module number, where 1 denotes the module in Slot 1. ▪ Port = Port number, where 1 denotes Port 1 of a module. For example: ▪ Below configuration forwards calls originally destined to Port 1 of Module 1 to "1001" upon On Busy: FwdInfo 0 = 1,1001,30,1,1; ▪ Below configuration forwards calls originally destined to Port 2 of Module 1 to an IP address upon On Busy: FwdInfo 1 = 1,2003@10.5.1.1,30,1,2; For a detailed description of this table, see Configuring Call Forward on page 481. **Note:** This parameter is applicable only to FXS and FXO interfaces.
Call Forward Reminder Ring Parameters	

Parameter	Description
Notes: - These parameters are applicable only to FXS interfaces. - For a description of this feature, see Call Forward Reminder Ring on page 433.	
Web/EMS: Enable NRT Subscription CLI: nrt-subscription [EnableNRTSubscription]	Enables endpoint subscription for Ring reminder event notification feature. - [0] Disable (default) - [1] Enable
Web: AS Subscribe IPGroupID CLI: as-subs-ipgroupid [ASSubscribeIPGroupID]	Defines the IP Group ID that contains the Application server for Subscription. The valid value range is 1 to 8. The default is -1 (i.e., not configured).
Web: NRT Retry Subscription Time EMS: NRT Subscription Retry Time [NRTSubscribeRetryTime]	Defines the Retry period (in seconds) for Dialog subscription if a previous request failed. The valid value range is 10 to 7200. The default is 120.
Web/EMS: Call Forward Ring Tone ID CLI: cfe-ring-tone-id [CallForwardRingToneID]	Defines the ringing tone type played when call forward notification is accepted. The valid value range is 1 to 5. The default is 1.

67.10.5.4 Message Waiting Indication Parameters

The message waiting indication (MWI) parameters are described in the table below.

Table 67-48: MWI Parameters

Parameter	Description
Web: Enable MWI EMS: MWI Enable CLI: enable-mwi **[EnableMWI]**	Enables Message Waiting Indication (MWI). - **[0]** Disable (default). - **[1]** Enable **Notes:** - This parameter is applicable only to FXS interfaces. - The device supports only the receipt of SIP MWI NOTIFY messages (the device doesn't generate these messages). - For more information on MWI, see "Message Waiting Indication" on page 436.
Web/EMS: MWI Analog Lamp CLI: mwi-analog-lamp **[MWIAnalogLamp]**	Enables the visual display of MWI. - **[0]** Disable (default). - **[1]** Enable = Enables visual MWI by supplying line voltage of approximately 100 VDC to activate the phone's lamp. **Notes:** - This parameter is applicable only for FXS interfaces. - This parameter can also be configured in a Tel Profile.

Parameter	Description
Web/EMS: MWI Display CLI: enable-mwi **[MWIDisplay]**	Enables sending MWI information to the phone display. ▪ **[0]** Disable = (Default) MWI information isn't sent to display. ▪ **[1]** Enable = The device generates an MWI message (determined by the parameter CallerIDType), which is displayed on the MWI display. **Note:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter can also be configured in a Tel Profile.
Web: Subscribe to MWI EMS: Enable MWI Subscription CLI: subscribe-to-mwi **[EnableMWISubscription]**	Enables subscription to an MWI server. ▪ **[0]** No (default) ▪ **[1]** Yes **Notes:** ▪ To configure the MWI server address, use the MWIServerIP parameter. ▪ To configure whether the device subscribes per endpoint or per the entire device, use the parameter SubscriptionMode.
Web: MWI Server IP Address EMS: MWI Server IP CLI: mwi-srvr-ip-addr **[MWIServerIP]**	Defines the MWI server's IP address. If provided, the device subscribes to this IP address. The MWI server address can be configured as a numerical IP address or as a domain name. If not configured, the Proxy IP address is used instead.
Web/EMS: MWI Server Transport Type CLI: mwi-srvr-transp-type **[MWIServerTransportType]**	Determines the transport layer used for outgoing SIP dialogs initiated by the device to the MWI server. ▪ **[-1]** Not Configured (default) ▪ **[0]** UDP ▪ **[1]** TCP ▪ **[2]** TLS **Note:** When set to ‘Not Configured’, the value of the parameter SIPTransportType is used.
Web: MWI Subscribe Expiration Time EMS: MWI Expiration Time CLI: mwi-subs-expr-time **[MWIExpirationTime]**	Defines the MWI subscription expiration time in seconds. The default is 7200 seconds. The range is 10 to 2,000,000.
Web: MWI Subscribe Retry Time EMS: Subscribe Retry Time CLI: mwi-subs-rtry-time **[SubscribeRetryTime]**	Defines the subscription retry time (in seconds) after last subscription failure. The default is 120 seconds. The range is 10 to 2,000,000.
Web: Subscription Mode CLI: subscription-mode **[SubscriptionMode]**	Determines the method the device uses to subscribe to an MWI server. ▪ **[0]** Per Endpoint = (Default) Each endpoint subscribes separately - typically used for FXS interfaces. ▪ **[1]** Per Gateway = Single subscription for the entire device - typically used for FXO interfaces.
EMS: ETSI VMWI Type One Standard CLI: etsi-vmwi-type-one-standard **[ETSIVMWITypeOneStandard]**	Determines the ETSI Visual Message Waiting Indication (VMWI) Type 1 sub-standard. ▪ **[0]** = (Default) ETSI VMWI between rings ▪ **[1]** = ETSI VMWI before ring DT_AS ▪ **[2]** = ETSI VMWI before ring RP_AS

Parameter	Description
	- **[3]** = ETSI VMWI before ring LR_DT_AS - **[4]** = ETSI VMWI not ring related DT_AS - **[5]** = ETSI VMWI not ring related RP_AS - **[6]** = ETSI VMWI not ring related LR_DT_AS **Note:** For this parameter to take effect, a device reset is required.
EMS: Bellcore VMWI Type One Standard CLI: bellcore-vmwi-type-one-standard **[BellcoreVMWITypeOneStandard]**	Determines the Bellcore VMWI sub-standard. - **[0]** = (Default) Between rings. - **[1]** = Not ring related. **Note:** For this parameter to take effect, a device reset is required.

67.10.5.5 Call Hold Parameters

The call hold parameters are described in the table below.

Table 67-49: Call Hold Parameters

Parameter	Description
Web/EMS: Enable Hold CLI: hold **[EnableHold]**	Global parameter that enables the Call Hold feature (analog interfaces) and interworking of the Hold/Retrieve supplementary service from ISDN PRI to SIP. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableHold). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web/EMS: Hold Format CLI: hold-format **[HoldFormat]**	Defines the format of the SDP in the sent re-INVITE hold request. - **[0]** 0.0.0.0 = (Default) The SDP "c=" field contains the IP address "0.0.0.0" and the "a=inactive" attribute. - **[1]** Send Only = The SDP "c=" field contains the device's IP address and the "a=sendonly" attribute. - **[2]** x.y.z.t = The SDP "c=" field contains the device's IP address and the "a=inactive" attribute. **Notes:** - The device does not send any RTP packets when it is in hold state. - For digital interfaces: This parameter is applicable only to QSIG and Euro ISDN protocols.
Web/EMS:Held Timeout CLI: held-timeout **[HeldTimeout]**	Defines the time interval that the device allows for a call to remain on hold. If a Resume (un-hold Re-INVITE) message is received before the timer expires, the call is renewed. If this timer expires, the call is released (terminated). - **[-1]** = (Default) The call is placed on hold indefinitely until the initiator of the on hold retrieves the call again. - **[0 - 2400]** = Time to wait (in seconds) after which the call is released.
Web: Call Hold Reminder Ring Timeout EMS: CHRR Timeout	Defines the duration (in seconds) that the Call Hold Reminder Ring is played. If a user hangs up while a call is still on hold or there is a call waiting, then the FXS interface immediately rings the extension for the

Parameter	Description
CLI: call-hold-remnd-rng [CHRRTimeout]	duration specified by this parameter. If the user off-hooks the phone, the call becomes active. The valid range is 0 to 600. The default is 30. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This Reminder Ring feature can be disabled using the DisableReminderRing parameter.
CLI: dis-reminder-ring [DisableReminderRing]	Disables the reminder ring, which notifies the FXS user of a call on hold or a waiting call when the phone is returned to on-hook position. ▪ [0] = (Default) The reminder ring feature is active. In other words, if a call is on hold or there is a call waiting and the phone is changed from offhook to onhook, the phone rings (for a duration defined by the CHRRTimeout parameter) to "remind" you of the call hold or call waiting. ▪ [1] = Disables the reminder ring. If a call is on hold or there is a call waiting and the phone is changed from offhook to onhook, the call is released (and the device sends a SIP BYE to the IP). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ This parameter is typically used for MLPP, allowing preemption to clear held calls.
CLI: dtmf-during-hold **[PlayDTMFduringHold]**	Determines whether the device sends DTMF signals (or DTMF SIP INFO message) when a call is on hold. ▪ **[0]** = (Default) Disable. ▪ **[1]** = Enable **-** If the call is on hold, the device stops playing the Held tone (if it is played) and sends DTMF: ✓ To Tel side: plays DTMF digits according to the received SIP INFO message(s). (The stopped held tone is not played again.) ✓ To IP side: sends DTMF SIP INFO messages to an IP destination if it detects DTMF digits from the Tel side.

67.10.5.6 Call Transfer Parameters

The call transfer parameters are described in the table below.

Table 67-50: Call Transfer Parameters

Parameter	Description
Web/EMS: Enable Transfer CLI: enable-transfer **[EnableTransfer]**	Enables the Call Transfer feature. ▪ **[0]** Disable ▪ **[1]** Enable = (Default) The device responds to a REFER message with the Referred-To header to initiate a call transfer. For analog interfaces: If the transfer service is enabled, the user can activate Transfer using hook-flash signaling. If this service is enabled, the remote party performs the call transfer. **Notes:** ▪ To use call transfer, the devices at both ends must support this option. ▪ To use call transfer, set the parameter EnableHold to 1.

Parameter	Description
Web: Transfer Prefix EMS: Logical Prefix For Transferred Call CLI: transfer-prefix **[xferPrefix]**	Defines the string that is added as a prefix to the transferred/forwarded called number when the REFER/3xx message is received. **Notes:** - The number manipulation rules apply to the user part of the Refer-To and/or Contact URI before it is sent in the INVITE message. - This parameter can be used to apply different manipulation rules to differentiate transferred/forwarded (only for analog interfaces) number from the originally dialed number.
Web: Transfer Prefix IP 2 Tel CLI: xfer-prefix-ip2tel **[XferPrefixIP2Tel]**	Defines the prefix that is added to the destination number received in the SIP Refer-To header (for IP-to-Tel calls). This parameter is applicable to FXO/CAS blind transfer modes, i.e., LineTransferMode = 1, 2 or 3, and TrunkTransferMode = 1 or 3 (for CAS). The valid range is a string of up to 9 characters. By default, no value is defined. **Note:** This parameter is also applicable to ISDN Blind Transfer, according to AT&T Toll Free Transfer Connect Service (TR 50075) “Courtesy Transfer-Human-No Data”. To support this transfer mode, you need to configure the parameter XferPrefixIP2Tel to "*8" and the parameter TrunkTransferMode to 5.
Web/EMS: Enable Semi-Attended Transfer CLI: semi-att-transfer **[EnableSemiAttendedTransfer]**	Determines the device behavior when Transfer is initiated while in Alerting state. - **[0]** Disable = (Default) Send REFER with the Replaces header. - **[1]** Enable = Send CANCEL, and after a 487 response is received, send REFER without the Replaces header.
Web: Blind EMS: Blind Transfer CLI: blind-transfer **[KeyBlindTransfer]**	Defines the keypad sequence to activate blind transfer for established Tel-to-IP calls. The Tel user can perform blind transfer by dialing the KeyBlindTransfer digits, followed by a transferee destination number. After the KeyBlindTransfer DTMF digits sequence is dialed, the current call is put on hold (using a Re-INVITE message), a dial tone is played to the channel, and then the phone number collection starts. After the destination phone number is collected, it is sent to the transferee in a SIP REFER request in a Refer-To header. The call is then terminated and a confirmation tone is played to the channel. If the phone number collection fails due to a mismatch, a reorder tone is played to the channel. **Note:** For FXS/FXO interfaces, it is possible to configure whether the KeyBlindTransfer code is added as a prefix to the dialed destination number, by using the parameter KeyBlindTransferAddPrefix.
EMS: Blind Transfer Add Prefix CLI: blind-xfer-add-prefix [KeyBlindTransferAddPrefix]	Determines whether the device adds the Blind Transfer code (defined by the KeyBlindTransfer parameter) to the dialed destination number. - [0] Disable (default) - [1] Enable **Note:** This parameter is applicable only to FXO and FXS

Parameter	Description
	interfaces.
EMS: Blind Transfer Disconnect Timeout CLI: blind-xfer-disc-tmo **[BlindTransferDisconnectTimeout]**	Defines the duration (in milliseconds) for which the device waits for a disconnection from the Tel side after the Blind Transfer Code (KeyBlindTransfer) has been identified. When this timer expires, a SIP REFER message is sent toward the IP side. If this parameter is set to 0, the REFER message is immediately sent. The valid value range is 0 to 1,000,000. The default is 0.
Web: QSIG Path Replacement Mode CLI: qsig-path-replacement-md [QSIGPathReplacementMode]	Enables QSIG transfer for IP-to-Tel and Tel-to-IP calls. ▪ [0] IP2QSIGTransfer = (Default) Enables IP-to-QSIG transfer. ▪ [1] QSIG2IPTransfer = Enables QSIG-to-IP transfer.
CLI: replace-tel2ip-calnum-to [ReplaceTel2IPCallingNumTimeout]	Defines the maximum duration (timeout) to wait between call Setup and Facility with Redirecting Number for replacing the calling number (for Tel-to-IP calls). The valid value range is 0 to 10,000 msec. The default is 0. The interworking of the received Setup message to a SIP INVITE is suspended when this parameter is set to any value greater than 0. This means that the redirecting number in the Setup message is not checked. When a subsequent Facility with Call Transfer Complete/Update is received with a non-empty Redirection Number, the Calling Number is replaced with the received redirect number in the sent INVITE message. If the timeout expires, the device sends the INVITE without changing the calling number. **Notes:** ▪ The suspension of the INVITE message occurs for all calls. ▪ This parameter is applicable to QSIG.
Web: Call Transfer using re-INVITEs CLI: enable-call-transfer-using-reinvites [EnableCallTransferUsingReinvites]	Enables call transfer using re-INVITEs. ▪ [0] Disable = (Default) Call transfer is done using REFER messages. ▪ [1] Enable = Call transfer is done by sending re-INVITE messages (instead of REFER). **Notes:** ▪ The device uses two DSP channels per transferred call. Thus, to use this feature, you also need to configure the maximum number of available DSP channels, using the MediaChannels parameter. ▪ This parameter is applicable only to FXS interfaces.

Parameter	Description
Web: IP2IP Transfer Mode CLI: ip2ip-transfer-mode [IP2IPTransfermode]	Determines the interworking of incoming mid-call SIP REFER messages to outgoing REFER messages, for calls pertaining to the IP-to-IP application. ▪ [0] Disable = (Default) Upon receipt of a REFER message, the device sends an INVITE to the refer-to destination with or without the Replaces header. ▪ [1] Enable = Upon receipt of a REFER message, the device forwards the REFER message and all relevant SIP messages from and to the transferor, to the target destination during call transfer. For consultation call transfer, the REFER message contains a 'replaces' parameter in the Refer-To header. In this case, the outgoing REFER also contains a 'replaces' parameter in the Refer-To header. **Notes:** ▪ This parameter is applicable to blind and consultation call transfers. ▪ This parameter is applicable only to the IP-to-IP application.

67.10.5.7 Multi-Line Extensions and Supplementary Services Parameters

The multi-line extensions and supplementary services parameters are described in the table below.

Table 67-51: Multi-line Extensions and Supplementary Services Parameters

Parameter	Description
Supplementary Services Table	
Web: Supplementary Services Table EMS: Digital Gateway Provisioning > ISDN Supplementary Services CLI: configure voip/gw digitalgw isdn-supp-serv **[ISDNSuppServ]**	This table parameter defines phone extension numbers per FXS/BRI port and configures various supplementary services per endpoint. The format of the ini file table parameter is as follows: [ISDNSuppServ] FORMAT ISDNSuppServ_Index = ISDNSuppServ_PhoneNumber, ISDNSuppServ_Module, ISDNSuppServ_Port, ISDNSuppServ_UserId, ISDNSuppServ_UserPassword, ISDNSuppServ_CallerID, ISDNSuppServ_IsPresentationRestricted, ISDNSuppServ_IsCallerIDEnabled; [\ISDNSuppServ] For a detailed description of this table, see "Configuring Multi-Line Extensions and Supplementary Services" on page 465.

67.10.5.8 Three-Way Conferencing Parameters

The three-way conferencing parameters are described in the table below.

Table 67-52: Three-Way Conferencing Parameters

Parameter	Description
Web: Enable 3-Way Conference EMS: Enable 3 Way	Enables the 3-Way Conference feature.

Parameter	Description
CLI: enable-3w-conf **[Enable3WayConference]**	▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: 3-Way Conference Mode EMS: 3 Way Mode CLI: 3w-conf-mode **[3WayConferenceMode]**	Defines the mode of operation for three-way conferencing. ▪ **[0]** AudioCodes Media Server = (Default) The conference-initiating INVITE sent by the device, uses the ConferenceID concatenated with a unique identifier as the Request-URI. This same Request-URI is set as the Refer-To header value in the REFER messages that are sent to the two remote parties. This conference mode is used when operating with AudioCodes IPMedia conferencing server. ▪ **[1]** Non-AudioCodes Media Server = The conference-initiating INVITE sent by the device, uses only the ConferenceID as the Request-URI. The Conference server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. This Conference URI is then included by the device in the Refer-To header value in the REFER messages sent by the device to the remote parties. The remote parties join the conference by sending INVITE messages to the conference using this conference URI. ▪ **[2]** On Board = On-board, three-way conference. The conference is established on the device without the need of an external Conference server. You can limit the number of simultaneous, on-board 3-way conference calls, by using the MaxInBoardConferenceCalls parameter. ▪ **[3]** Huawei Media Server = The conference is managed by an external, third-party Conferencing server. The conference-initiating INVITE sent by the device, uses only the ConferenceID as the Request-URI. The Conferencing server sets the Contact header of the 200 OK response to the actual unique identifier (Conference URI) to be used by the participants. The Conference URI is included in the URI of the REFER with a Replaces header sent by the device to the Conferencing server. The Conferencing server then sends an INVITE with a Replaces header to the remote participants. **Notes:** ▪ This parameter is applicable only to FXS and BRI interfaces. ▪ Three-way conferencing using an external conference server is supported only by FXS interfaces. ▪ When using an external Conferencing server, a conference call with up to six participants can be established.
Web: Max. 3-Way Conference EMS: Max In Board Calls **[MaxInBoardConferenceCalls]**	Defines the maximum number of simultaneous, on-board three-way conference calls. The valid range is 0 to 5. The default is 2. **Notes:** ▪ For enabling on-board, three-way conferencing, use the 3WayConferenceMode parameter. ▪ This parameter is applicable only to FXS and BRI interfaces.
Web: Establish Conference Code EMS: Establish Code CLI: estb-conf-code	Defines the DTMF digit pattern, which upon detection generates the conference call when three-way conferencing is enabled

Parameter	Description
[ConferenceCode]	(Enable3WayConference is set to 1). The valid range is a 25-character string. The default is "!" (Hook-Flash). **Note:** If the FlashKeysSequenceStyle parameter is set to 1 or 2, the setting of the ConferenceCode parameter is overridden.
Web/EMS: Conference ID CLI: conf-id **[ConferenceID]**	Defines the Conference Identification string. The valid value is a string of up to 16 characters. The default is "conf". The device uses this identifier in the Conference-initiating INVITE that is sent to the media server when the Enable3WayConference parameter is set to 1.
Web: Use Different RTP port After Hold CLI: configure voip > sip-definition advanced-settings > dfrnt-port-after-hold **[UseDifferentRTPportAfterHold]**	Enables the use of different RTP ports for the two calls involved in a three-way conference call made by the FXS endpoint in the initial outgoing INVITE requests. ▪ **[0]** Disable = (Default) The FXS endpoint makes the first and second calls on the same RTP port in the initial outgoing INVITE request. If a three-way conference is then made, the device sends a re-INVITE to the held call to retrieve it and to change the RTP port to a different port number. For example: A first calls B on port 6000 and places B on hold. A then calls C, also on port 6000. The device sends a re-INVITE to the held call to retrieve it and changes the port to 6010. ▪ **[1]** Enable = The FXS endpoint makes the first and second calls on different RTP ports in the initial outgoing INVITE request. If a three-way conference is then made, the device sends a re-INVITE to the held call to retrieve it, without changing the port of the held call. For example: A first calls B on port 6000 and places B on hold. A then calls C on port 6010. The device sends a re-INVITE to the held call to retrieve it (without changing the port, i.e., remains 6010). **Notes:** ▪ When this feature is enabled and only one RTP port is available, only one call can be made by the FXS endpoint, as there is no free RTP port for a second call. ▪ When this feature is enabled and you are using the Call Forking feature, every forked call is sent with a different RTP port. As the device can fork a call to up to 10 destinations, the device requires at least 10 free RTP ports. ▪ This parameter is applicable only to FXS interfaces.

67.10.5.9 MLPP and Emergency Call Parameters

The Multilevel Precedence and Preemption (MLPP) and emergency E911 call parameters are described in the table below.

Table 67-53: MLPP and Emergency E911 Call Parameters

Parameter	Description
Web/EMS: Call Priority Mode	Defines priority call handling, for all calls.

Parameter	Description
CLI: call-prio-mode **[CallPriorityMode]**	▪ **[0]** Disable (default). ▪ **[1]** MLPP = MLPP Priority Call handling is enabled. MLPP prioritizes call handling whereby the relative importance of various kinds of communications is strictly defined, allowing higher precedence communication at the expense of lower precedence communications. Higher priority calls override less priority calls when, for example, congestion occurs in a network. ▪ **[2]** Emergency = Preemption of IP-to-Tel E911 emergency calls. If the device receives an E911 call and there are unavailable channels to receive the call, the device terminates one of the channel calls and sends the E911 call to that channel. The preemption is done only on a channel pertaining to the same Trunk Group for which the E911 call was initially destined and if the channel select mode (configured by the ChannelSelectMode parameter) is set to other than By Dest Phone Number (0). The preemption is done only if the incoming IP-to-Tel call is identified as an emergency call. The device identifies emergency calls by one of the following: ✓ The destination number of the IP call matches one of the numbers defined by the EmergencyNumbers parameter. (For E911, you must define this parameter with the value "911".) ✓ The incoming SIP INVITE message contains the “emergency” value in the Priority header. **Notes:** ▪ This parameter is applicable to FXS/FXO, CAS, and ISDN. ▪ For FXO interfaces, the preemption is done only on existing IP-to-Tel calls. In other words, if all the current FXO channels are busy with calls that were initiated by the FXO (i.e., Tel-to-IP calls), new incoming emergency IP-to-Tel calls are dropped. ▪ MLPP and Emergency services can also be configured in a Tel Profile. ▪ For more information, see "Pre-empting Existing Call for E911 IP-to-Tel Call" on page 448.
Emergency E911 Parameters	
[E911Gateway]	Enables Enhanced 9-1-1(E9-1-1) support for ELIN handling in Microsoft Lync Server 2010 environment. ▪ [0] = Disable (default) ▪ [1] = Enable ▪ [2] = Location-based manipulations
[E911CallbackTimeout]	Defines the maximum interval within which the PSAP can use the ELIN to call back the E9-1-1 caller. This interval starts from when the initial call established with the PSAP is terminated. The valid range is 1 to 60 (minutes). The default is 30.
Web: Emergency Special Release Cause CLI: emrg-spcl-rel-cse [EmergencySpecialReleaseCause]	Enables the device to send a SIP 503 "Service Unavailable" response if an emergency call cannot be established (i.e., rejected). This can occur, for example, due to the PSTN (for example, the destination is busy or not found) or ELIN Gateway

Parameter	Description
	(for example, lack of resources or an internal error). ▪ [0] = Disable (default) ▪ [1] = Enable
EMS: Enable 911 PSAP [Enable911PSAP]	Enables the support for the E911 DID protocol, according to the Bellcore GR-350-CORE standard. This protocol defines signaling between E911 Tandem Switches and the PSAP, using analog loop-start lines. The FXO device can be installed instead of an E911 switch, connected directly to PSAP DID loop-start lines. ▪ [0] = Disable (default) ▪ [1] = Enable **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ This parameter can also be configured in a Tel Profile.
Web/EMS: Emergency Numbers CLI: emerg-nbs **[EmergencyNumbers]**	Defines a list of “emergency” numbers. For FXS: When one of these numbers is dialed, the outgoing INVITE message includes the SIP Priority and Resource-Priority headers. If the user places the phone on-hook, the call is not disconnected. Instead, a Hold Re-INVITE request is sent to the remote party. Only if the remote party disconnects the call (i.e., a BYE is received) or a timer expires (set by the EmergencyRegretTimeout parameter) is the call terminated. For FXO, CAS, and ISDN: These emergency numbers are used for the preemption of E911 IP-to-Tel calls when there are unavailable or busy channels. In this scenario, the device terminates one of the busy channels and sends the emergency call to this channel. This feature is enabled by setting the CallPriorityMode parameter to 2 (“Emergency”). For a description of this feature, see "Pre-empting Existing Call for E911 IP-to-Tel Call" on page 448. The list can include up to four different numbers, where each number can be up to four digits long. Example: EmergencyNumbers = ‘100’,’911’,’112’
Web: Emergency Calls Regret Timeout EMS: Emergency Regret Timeout CLI: emerg-calls-regrt-t-out [EmergencyRegretTimeout]	Defines the time (in minutes) that the device waits before tearing-down an emergency call (defined by the parameter EmergencyNumbers). Until this time expires, an emergency call can only be disconnected by the remote party, typically, by a Public Safety Answering Point (PSAP). The valid range is 1 to 30. The default is 10. **Note:** This parameter is applicable only to FXS interfaces.
Multilevel Precedence and Preemption (MLPP) Parameters	
Web: MLPP Default Namespace EMS: Default Name Space CLI: mlpp-dflt-namespace [MLPPDefaultNamespace]	Determines the namespace used for MLPP calls received from the ISDN side without a Precedence IE and destined for an Application server. This value is used in the Resource-Priority header of the outgoing SIP INVITE request. ▪ [1] DSN (default) ▪ [2] DOD ▪ [3] DRSN ▪ [5] UC ▪ [7] CUC

Parameter	Description
	Note: If the ISDN message contains a Precedence IE, the device automatically interworks the "network identity" digits in the IE to the network domain subfield in the Resource-Priority header. For more information, see Multilevel Precedence and Preemption on page 461.
[ResourcePriorityNetworkDomains]	Defines up to 32 user-defined MLPP network domain names (namespaces). This value is used in the AS-SIP Resource-Priority header of the outgoing SIP INVITE request. This parameter is used in combination with the MLPPDefaultNamespace parameter, where you need to enter the table row index as its value. This parameter is also used for mapping the Resource-Priority field value of the SIP Resource-Priority header to the ISDN PRI Precedence Level IE. The mapping is configured by the field, EnableIp2TelInterworking: ▪ Disabled: The network-domain field in the Resource-Priority header is set to "0 1 0 0" (i.e., "routine") in the Precedence Level field. ▪ Enabled: The network-domain field in the Resource-Priority header is set in the Precedence Level field according to Table 5.3.2.12-4 (Mapping of RPH r-priority Field to PRI Precedence Level Value). The domain name can be a string of up to 10 characters. The format of this table ini file parameter is as follows: FORMAT ResourcePriorityNetworkDomains_Index = ResourcePriorityNetworkDomains_Name, ResourcePriorityNetworkDomains_EnableIp2TelInterworking; ResourcePriorityNetworkDomains 1 = dsn, 0; ResourcePriorityNetworkDomains 2 = dod, 0; ResourcePriorityNetworkDomains 3 = drsn, 0; ResourcePriorityNetworkDomains 5 = uc, 1; ResourcePriorityNetworkDomains 7 = cuc, 0; [\ResourcePriorityNetworkDomains] **Notes:** ▪ Indices 1, 2, 3, 5, and 7 cannot be modified and are defined for DSN, DOD, DRSN, UC, and CUC, respectively. ▪ If the MLPPDefaultNamespace parameter is set to -1, interworking from PSTN NI digits is done automatically.
Web/EMS: Default Call Priority CLI: dflt-call-prio [SIPDefaultCallPriority]	Determines the default call priority for MLPP calls. ▪ [0] 0 = (Default) ROUTINE ▪ [2] 2 = PRIORITY ▪ [4] 4 = IMMEDIATE ▪ [6] 6 = FLASH ▪ [8] 8 = FLASH-OVERRIDE ▪ [9] 9 = FLASH-OVERRIDE-OVERRIDE If the incoming SIP INVITE request doesn't contain a valid priority value in the SIP Resource-Priority header, the default value is used in the Precedence IE (after translation to the relevant ISDN Precedence value) of the outgoing PRI Setup message. If the incoming PRI Setup message doesn't contain a valid

Parameter	Description
	Precedence Level value, the default value is used in the Resource-Priority header of the outgoing SIP INVITE request. In this scenario, the character string is sent without translation to a numerical value.
Web: MLPP DiffServ EMS: Diff Serv CLI: mlpp-diffserv **[MLPPDiffserv]**	Defines the DiffServ value (differentiated services code point/DSCP) used in IP packets containing SIP messages that are related to MLPP calls. This parameter defines DiffServ for incoming and outgoing MLPP calls with the Resource-Priority header. The valid range is 0 to 63. The default is 50.
Web/EMS: Preemption Tone Duration CLI: preemp-tone-dur [PreemptionToneDuration]	Defines the duration (in seconds) in which the device plays a preemption tone to the Tel and IP sides if a call is preempted. The valid range is 0 to 60. The default is 3. **Note:** If set to 0, no preemption tone is played.
Web: MLPP Normalized Service Domain EMS: Normalized Service Domain CLI: mlpp-norm-ser-dmn [MLPPNormalizedServiceDomain]	Defines the MLPP normalized service domain string. If the device receives an MLPP ISDN incoming call, it uses the parameter (if different from ‘FFFFFF’) as a Service domain in the SIP Resource-Priority header in outgoing INVITE messages. If the parameter is configured to ‘FFFFFF’, the Resource-Priority header is set to the MLPP Service Domain obtained from the Precedence IE. The valid value is 6 hexadecimal digits. The default is ‘000000’. **Note:** This parameter is applicable only to the MLPP NI-2 ISDN variant with CallPriorityMode set to 1.
CLI: mlpp-nwrk-id [MLPPNetworkIdentifier]	Defines the MLPP network identifier (i.e., International prefix or Telephone Country Code/TCC) for IP-to-ISDN calls, according to the UCR 2008 and ITU Q.955 specifications. The valid range is 1 to 999. The default is 1 (i.e., USA). The MLPP network identifier is sent in the Facility IE of the ISDN Setup message. For example: ▪ MLPPNetworkIdentifier set to default (i.e., USA, 1): PlaceCall- MLPPNetworkID:0100 MlppServiceDomain:123abc, MlppPrecLevel:5 Fac(1c): 91 a1 15 02 01 05 02 01 19 30 0d 0a 01 05 0a 01 01 04 05 01 00 12 3a bc ▪ MLPPNetworkIdentifier set to 490: PlaceCall- MLPPNetworkID:9004 MlppServiceDomain:123abc, MlppPrecLevel:5 Fac(1c): 91 a1 15 02 01 0a 02 01 19 30 0d 0a 01 05 0a 01 01 04 05 90 04 12 3a bc
Web: MLPP Default Service Domain EMS: Default Service Domain CLI: mlpp-dflt-srv-domain [MLPPDefaultServiceDomain]	Defines the MLPP default service domain string. If the device receives a non-MLPP ISDN incoming call (without a Precedence IE), it uses the parameter (if different than “FFFFFF”) as a Service domain in the SIP Resource-Priority header in outgoing (Tel-to-IP calls) INVITE messages. This parameter is used in conjunction with the parameter SIPDefaultCallPriority. If MLPPDefaultServiceDomain is set to 'FFFFFF', the device interworks the non-MLPP ISDN call to non-MLPP SIP call, and the outgoing INVITE does not contain the Resource-Priority header. The valid value is a 6 hexadecimal digits. The default is

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>"000000".
<b>Note:</b> This parameter is applicable only to the MLPP NI-2 ISDN variant with CallPriorityMode set to 1.</td></tr>
<tr><td>Web/EMS: Precedence Ringing Type
CLI: precedence-ringing
[PrecedenceRingingType]</td><td>Defines the index of the Precedence Ringing tone in the Call Progress Tones (CPT) file. This tone is used when the parameter CallPriorityMode is set to 1 and a Precedence call is received from the IP side.
The valid range is -1 to 16. The default is -1 (i.e., plays standard ringing tone).
<b>Note:</b> This parameter is applicable only to analog interfaces.</td></tr>
<tr><td>EMS: E911 MLPP Behavior
CLI: e911-mlpp-bhvr
[E911MLPPBehavior]</td><td>Defines the E911 (or Emergency Telecommunication Services/ETS) MLPP Preemption mode:
▪ [0] = (Default) Standard Mode - ETS calls have the highest priority and preempt any MLPP call.
▪ [1] = Treat as routine mode - ETS calls are handled as routine calls.
<b>Note:</b> This parameter is applicable only to analog interfaces.</td></tr>
<tr><td>CLI: resource-prio-req
<b>[RPRequired]</b></td><td>Determines whether the SIP resource-priority tag is added in the SIP Require header of the INVITE message for Tel-to-IP calls.
▪ <b>[0]</b> Disable = Excludes the SIP resource-priority tag from the SIP Require header.
▪ <b>[1]</b> Enable = (Default) Adds the SIP resource-priority tag in the SIP Require header.
<b>Note:</b> This parameter is applicable only to MLPP priority call handling (i.e., only when the CallPriorityMode parameter is set to 1).</td></tr>
<tr><td colspan="2"><b>Multiple Differentiated Services Code Points (DSCP) per MLPP Call Priority Level (Precedence) Parameters</b>
The MLPP service allows placement of priority calls, where properly validated users can preempt (terminate) lower-priority phone calls with higher-priority calls. For each MLPP call priority level, the DSCP can be set to a value from 0 to 63. The Resource Priority value in the Resource-Priority SIP header can be one of the following:
<table>
<tr><th>MLPP Precedence Level</th><th>Precedence Level in Resource-Priority SIP Header</th></tr>
<tr><td>0 (lowest)</td><td>routine</td></tr>
<tr><td>2</td><td>priority</td></tr>
<tr><td>4</td><td>immediate</td></tr>
<tr><td>6</td><td>flash</td></tr>
<tr><td>8</td><td>flash-override</td></tr>
<tr><td>9 (highest)</td><td>flash-override-override</td></tr>
</table></td></tr>
<tr><td>Web/EMS: RTP DSCP for MLPP Routine
CLI: dscp-4-mlpp-rtn
<b>[MLPPRoutineRTPDSCP]</b></td><td>Defines the RTP DSCP for MLPP Routine precedence call level.
The valid range is -1 to 63. The default is -1.
<b>Note:</b> If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.</td></tr>
<tr><td>Web/EMS: RTP DSCP for MLPP Priority</td><td>Defines the RTP DSCP for MLPP Priority precedence call level.</td></tr>
</table>

Parameter	Description
CLI: dscp-4-mlpp-prio **[MLPPPriorityRTPDSCP]**	The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/EMS: RTP DSCP for MLPP Immediate CLI: dscp-4-mlpp-immed **[MLPPImmediateRTPDSCP]**	Defines the RTP DSCP for MLPP Immediate precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/EMS: RTP DSCP for MLPP Flash CLI: dscp-4-mlpp-flsh **[MLPPFlashRTPDSCP]**	Defines the RTP DSCP for MLPP Flash precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/EMS: RTP DSCP for MLPP Flash Override CLI: dscp-4-mlpp-flsh-ov **[MLPPFlashOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.
Web/EMS: RTP DSCP for MLPP Flash-Override-Override CLI: dscp-4-mlpp-flsh-ov-ov **[MLPPFlashOverOverRTPDSCP]**	Defines the RTP DSCP for MLPP Flash-Override-Override precedence call level. The valid range is -1 to 63. The default is -1. **Note:** If set to -1, the DiffServ value is taken from the global parameter PremiumServiceClassMediaDiffServ or as defined in IP Profiles per call.

67.10.5.10 Call Cut-Through Parameters

The call cut-through parameters are described in the table below.

Table 67-54: Call Cut-Through Parameters

Parameter	Description
Web: Enable Calls Cut Through EMS: Cut Through CLI: calls-cut-through [CutThrough]	Enables FXS endpoints to receive incoming IP calls while the port is in off-hook state. ▪ [0] Disable (default) ▪ [1] Enable If enabled, the FXS interface answers the call and 'cuts through' the voice channel if there is no other active call on the port, even if the port is in off-hook state. When the call is terminated (by the remote IP party), the device plays a reorder tone for a user-defined time (configured by the CutThroughTimeForReorderTone parameter) and is then ready to answer the next incoming call without on-hooking the phone. The waiting call is automatically answered by the device when the current call is terminated (configured by setting the parameter EnableCallWaiting to 1).

Parameter	Description
	Note: This feature is applicable only to FXS interfaces.
CLI: cut-through-anable [DigitalCutThrough]	Enables PSTN CAS channels/endpoints to receive incoming IP calls even if the B-channels are in off-hook state. ▪ [0] Disabled (default) ▪ [1] Enabled When enabled, this feature operates as follows: 1 A Tel-to-IP call is established (connected) by the device for a B-channel. 2 The device receives a SIP BYE (i.e., IP side ends the call) and plays a reorder tone to the PSTN side for the duration set by the CutThroughTimeForReOrderTone parameter. The device releases the call towards the IP side (sends a SIP 200 OK). 3 The PSTN side, for whatever reason, remains off-hook. 4 If a new IP call is received for this B-channel after the reorder tone has ended, the device “cuts through” the channel and connects the call immediately (despite the B-channel being in physical off-hook state) without playing a ring tone. If an IP call is received while the reorder tone is played, the device rejects the call. **Notes:** ▪ If this parameter is disabled and the PSTN side remains in off-hook state after the IP call ends the call, the device releases the call after 60 seconds. ▪ A special CAS table can be used to report call status events (Active/Idle) to the PSTN side during Cut Through mode. ▪ This feature can also be configured in a Tel Profile and therefore, assigned to specific B-channels that use specific CAS tables.

67.10.5.11 Automatic Dialing Parameters

The automatic dialing upon off-hook parameters are described in the table below.

Table 67-55: Automatic Dialing Parameters

Parameter	Description
Automatic Dialing Table	
Web: Automatic Dialing Table EMS: Analog Gateway Provisioning >	This table parameter defines telephone numbers that are automatically dialed when a specific FXS or FXO port is off-hooked. The format of the ini file table parameter is as follows: [TargetOfChannel]

Parameter	Description
Automatic dialing CLI: configure voip/gw analoggw automatic-dialing **[TargetOfChannel]**	FORMAT TargetOfChannel_Index = TargetOfChannel_Destination, TargetOfChannel_Type, TargetOfChannel_Module, TargetOfChannel_Port, TargetOfChannel_HotLineToneDuration; [\TargetOfChannel] For example, the below configuration defines automatic dialing of phone number 911 when the phone connected to Port 1 of Module 1 is off-hooked for over 10 seconds: TargetOfChannel 0 = 911, 1, 1, 1 ,10; **Notes:** ▪ The first index of this table ini file parameter is 0. ▪ TargetOfChannel_Module is the module number, where 1 denotes the module in Slot 1. ▪ TargetOfChannel_Port is the port number, where 1 denotes Port 1 on the module. ▪ This is parameter is applicable only to FXS and FXO interfaces. ▪ For a detailed description of this table, see "Configuring Automatic Dialing" on page 478.

67.10.5.12 Direct Inward Dialing Parameters

The Direct Inward Dialing (DID) parameters are described in the table below.

Table 67-56: DID Parameters

Parameter	Description
Web/EMS: DID Wink CLI: did-wink-enbl **[EnableDIDWink]**	Enables Direct Inward Dialing (DID) using Wink-Start signaling, typically used for signaling between an E-911 switch and the PSAP. ▪ **[0]** Disable (default) ▪ **[1]** Single = The device can be used for connection to EIA/TIA-464B DID Loop Start lines. Both FXO (detection) and FXS (generation) are supported: ✓ The FXO interface dials DTMF (or MF) digits upon detection of a Wink signal, instead of a dial tone. ✓ The FXS interface generates a Wink signal upon detection of an off-hook state, instead of playing a dial tone. For example: (Wink) KP I(I) xxx-xxxx ST (Off Hook) Where: ✓ I = one or two information digits ✓ x = ANI **Note:** The FXO interface generates such MF digits when the Enable911PSAP parameter is set to 1. ▪ **[2]** Double Wink = Double-wink signaling. This is applicable to FXS interfaces only. The FXS interface generates the first Wink upon detection of an off-hook state in the line. The second Wink is generated after a user-defined interval (configured by the TimeBetweenDIDWinks parameter) after which the DTMF/MF digits are collected by the device. Digits that arrive between the first and second Wink are ignored as they contain the same number. For example: (Wink) KP 911 ST (Wink) KP I(I) xxx-xxxx ST (Off Hook) ▪ **[3]** Wink & Polarity= ✓ **FXS:** The FXS interface generates the first Wink after it detects

Parameter	Description
	an off-hook state. A polarity change from normal to reversed is generated after a user-defined time (configured by the TimeBetweenDIDWinks parameter). DTMF/MF digits are collected by the device only after this polarity change. Digits that arrive between the first Wink and the polarity change are ignored as they always contain the same number. In this mode, the FXS interface does not generate a polarity change to normal if the Tel-to-IP call is answered by an IP party. Polarity reverts to normal when the call is released. For example: (Wink) KP 911 ST (Polarity) KP I(I) xxx-xxxx ST (Off Hook) ✓ **FXO:** For IP-to-Tel calls: 1) Upon incoming INVITE message, the FXO interface goes off-hook (seizes the line). 2) Upon detection of a Wink signal from the Tel side (instead of a dial tone), the FXO interface dials the digits, "KP911ST" (denotes *911#). 3) The FXO interface waits for polarity reversal change from normal to reverse for an interval of 2,000 msec. 4) Upon detection of a polarity reversal change, the FXO interface dials the DTMF (or MF) digits of the calling party (number that dialed 911) in the format "KP<ANI>ST" (*ANI#), where ANI is the calling number from the INVITE. If no polarity reversal, the FXO goes idle. For example: (Wink) KP911ST (Polarity Change) KP02963700ST Note: The Enable911PSAP parameter must be set to 1. **Notes:** ▪ The EnableReversalPolarity and PolarityReversalType parameters must be set to 1 for FXS interfaces. ▪ This parameter can also be configured in a Tel Profile.
[TimeBetweenDIDWinks]	Defines the interval (in msec) for wink signaling: ▪ Double-wink signaling [2]: interval between the first and second wink ▪ Wink and Polarity signaling [3]: interval between wink and polarity change The valid range is 100 to 2000. The default is 1000. **Note:** See the EnableDIDWink parameter for configuring the wink signaling type.
Web/EMS: Delay Before DID Wink CLI: delay-b4-did-wink **[DelayBeforeDIDWink]**	Defines the time interval (in msec) between the detection of the off-hook and the generation of the DID Wink. The valid range is 0 to 1,000. The default is 0. **Note:** This parameter is applicable only to FXS interfaces.
EMS: NTT DID Signalling Form CLI: NTT-DID-signaling-form **[NTTDIDSignallingForm]**	Determines the type of DID signaling support for NTT (Japan) modem: DTMF- or Frequency Shift Keying (FSK)-based signaling. The devices can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. ▪ **[0]** = (Default) FSK-based signaling ▪ **[1]** = DTMF-based signaling **Note:** This parameter is applicable only to FXS interfaces.
EMS: Enable DID **[EnableDID]**	This table parameter enables support for Japan NTT 'Modem' DID. FXS interfaces can be connected to Japan's NTT PBX using 'Modem' DID lines. These DID lines are used to deliver a called number to the PBX. The DID signal can be sent alone or combined with an NTT Caller ID

Parameter	Description
	signal. The format of the ini file table parameter is as follows: [EnableDID] FORMAT EnableDID_**Index** = EnableDID_**IsEnable**, EnableDID_Port, EnableDID_Module; [\EnableDID] Where, ▪ IsEnable = Enables [1] or disables [0] (default) Japan NTT Modem DID support. ▪ Port = Port number. ▪ Module = Module number. For example: EnableDID 0 = 1,1,2; (DID is enabled on Port 1 of Module 2) **Note:** This parameter is applicable only to FXS interfaces.
CLI: wink-time [WinkTime]	Defines the time (in msec) elapsed between two consecutive polarity reversals. This parameter can be used for DID signaling, for example, E911 lines to the Public Safety Answering Point (PSAP), according to the Bellcore GR-350-CORE standard (refer to the ini file parameter Enable911PSAP). The valid range is 0 to 4,294,967,295. The default is 200. **Notes:** ▪ This parameter is applicable to FXS and FXO interfaces. ▪ For this parameter to take effect, a device reset is required.

67.10.5.13 ISDN BRI Parameters

The automatic dialing upon off-hook parameters are described in the table below.

Table 67-57: Automatic Dialing Parameters

Parameter	Description
BRI-to-SIP Supplementary Services Codes for Call Forward **Note:** Upon receipt of an ISDN Facility message for call forward from the BRI phone, the device sends a SIP INVITE to the softswitch with a user-defined code in the SIP To header, representing the reason for the call forward. For more information on BRI call forwarding, see "BRI Call Forwarding" on page 434.	
Web/EMS: Call Forward Unconditional **[SuppServCodeCFU]**	Defines the prefix code for activating Call Forward Unconditional sent to the softswitch. The valid value is a string. By default, no value is defined. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Web/EMS: Call Forward Unconditional Deactivation **[SuppServCodeCFUDeact]**	Defines the prefix code for deactivating Call Forward Unconditional Deactivation sent to the softswitch. The valid value is a string. By default, no value is defined. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Web: Call Forward on Busy EMS: Code Call Forward on Busy **[SuppServCodeCFB]**	Defines the prefix code for activating Call Forward on Busy sent to the softswitch. The valid value is a string. By default, no value is defined. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').

Parameter	Description
Web: Call Forward on Busy Deactivation EMS: Code Call Forward on Busy Deactivation **[SuppServCodeCFBDeact]**	Defines the prefix code for deactivating Call Forward on Busy Deactivation sent to the softswitch. The valid value is a string. By default, no value is defined. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Web: Call Forward on No Reply EMS: Code Call Forward on No Reply **[SuppServCodeCFNR]**	Defines the prefix code for activating Call Forward on No Reply sent to the softswitch. The valid value is a string. By default, no value is defined. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').
Call Forward on No Reply Deactivation EMS: Code Call Forward on No Reply Deactivation **[SuppServCodeCFNRDeact]**	Defines the prefix code for deactivating Call Forward on No Reply Deactivation sent to the softswitch. The valid value is a string. By default, no value is defined. **Note:** The string must be enclosed in single apostrophe (e.g., '*72').

67.10.6 PSTN Parameters

This subsection describes the device's PSTN parameters.

67.10.6.1 General Parameters

The general PSTN parameters are described in the table below.

Table 67-58: General PSTN Parameters

Parameter	Description
Web: Trunk Name CLI: name (config-voip > interface <e1|t1|bri> name **[DigitalPortInfo_x]**	Defines an arbitrary name for a trunk. This can be used to help you easily identify the trunk. The valid value is a string of up to 40 characters. By default, the value is undefined. **Note:** For the ini file parameter, the *x* denotes the trunk number.
Web/EMS: Protocol Type CLI: protocol **[ProtocolType]**	Defines the PSTN protocol for all the Trunks. To configure the protocol type for a specific Trunk, use the *ini* file parameter ProtocolType_x: ▪ **[0]** NONE ▪ [1] E1 EURO ISDN = ISDN PRI Pan-European (CTR4) protocol ▪ [2] T1 CAS = Common T1 robbed bits protocols including E&M wink start, E&M immediate start, E&M delay dial/start and loop-start and ground start. ▪ [3] T1 RAW CAS ▪ [4] T1 TRANSPARENT = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 24 of all trunks are mapped to DSP channels. ▪ [5] E1 TRANSPARENT 31 = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 31 of each trunk are mapped to DSP channels. ▪ [6] E1 TRANSPARENT 30 = Transparent protocol, where no signaling is provided by the device. Timeslots 1 to 31, excluding time slot 16 of all trunks are mapped to DSP channels.

Parameter	Description
	- [7] E1 MFCR2 = Common E1 MFC/R2 CAS protocols (including line signaling and compelled register signaling). - [8] E1 CAS = Common E1 CAS protocols (including line signaling and MF/DTMF address transfer). - [9] E1 RAW CAS - [10] T1 NI2 ISDN = National ISDN 2 PRI protocol - [11] T1 4ESS ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 4ESS switch. - [12] T1 5ESS 9 ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-9 switch. - [13] T1 5ESS 10 ISDN = ISDN PRI protocol for the Lucent™/AT&T™ 5ESS-10 switch. - [14] T1 DMS100 ISDN = ISDN PRI protocol for the Nortel™ DMS switch. - [15] J1 TRANSPARENT - [16] T1 NTT ISDN = ISDN PRI protocol for the Japan - Nippon Telegraph Telephone (known also as INS 1500). - [17] E1 AUSTEL ISDN = ISDN PRI protocol for the Australian Telecom. - [18] E1 HKT ISDN = ISDN PRI (E1) protocol for the Hong Kong - HKT. - [19] E1 KOR ISDN = ISDN PRI protocol for Korean Operator (similar to ETSI). - [20] T1 HKT ISDN = ISDN PRI (T1) protocol for the Hong Kong - HKT. - [21] E1 QSIG = ECMA 143 QSIG over E1 - [22] E1 TNZ = ISDN PRI protocol for Telecom New Zealand (similar to ETSI) - [23] T1 QSIG = ECMA 143 QSIG over T1 - [30] E1 FRENCH VN6 ISDN = France Telecom VN6 - [31] E1 FRENCH VN3 ISDN = France Telecom VN3 - [34] T1 EURO ISDN =ISDN PRI protocol for Euro over T1 - [35] T1 DMS100 Meridian ISDN = ISDN PRI protocol for the Nortel™ DMS Meridian switch - [36] T1 NI1 ISDN = National ISDN 1 PRI protocol - [40] E1 NI2 ISDN = National ISDN 2 PRI protocol over E1 - [50] BRI EURO ISDN = Euro ISDN over BRI - [51] BRI N12 ISDN - [52] BRI DMS 100 ISDN - [53] BRI 5ESS 10 ISDN - [54] BRI QSIG = QSIG over BRI - [55] BRI VN6 = VN6 over BRI - [56] BRI NTT = BRI ISDN Japan (Nippon Telegraph) - [57] BRI IUA **Notes:** - All PRI trunks must be configured as the same line type (either E1 or T1). The device can support different variants of CAS and PRI protocols on different E1/T1 spans (no more than four simultaneous PRI variants). - BRI trunks can operate together with E1 or T1 trunks.
[ProtocolType_x]	Defines the protocol type for a specific trunk ID (where x denotes

Parameter	Description
	the Trunk ID and 0 is the first trunk). For more information, see the ProtocolType parameter.
[ISDNTimerT310]	Defines the T310 override timer for DMS, Euro ISDN, and ISDN NI-2 variants. An ISDN timer is started when a Q.931 Call Proceeding message is received. The timer is stopped when a Q.931 Alerting, Connect, or Disconnect message is received from the other end. If no ISDN Alerting, Progress, or Connect message is received within the duration of T310 timer, the call clears. The valid value range is 0 to 600 seconds. The default is 0 (i.e., use the default timer value according to the protocol's specifications). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When both the parameters ISDNDmsTimerT310 and ISDNTimerT310 are configured, the value of the parameter ISDNTimerT310 prevails.
[ISDNDMSTimerT310]	Defines the override T310 timer for the DMS-100 ISDN variant. T310 defines the timeout between the receipt of a Proceeding message and the receipt of an Alerting/Connect message. The valid range is 10 to 30. The default is 10 (seconds). **Notes:** ▪ Instead of configuring this parameter, it is recommended to use the parameter ISDNTimerT310. ▪ This parameter is applicable only to Nortel DMS and Nortel MERIDIAN PRI variants (ProtocolType = 14 and 35).
[ISDNTimerT301]	Defines the override T301 timer (in seconds). The T301 timer is started when a Q.931 Alert message is received. The timer is stopped when a Q.931 Connect/Disconnect message is received from the other side. If no Connect or Disconnect message is received within the duration of T301, the call is cleared. The valid range is 0 to 2400. The default is 0 (i.e., the default T301 timer value - 180 seconds - is used). If set to any other value than 0, it overrides the timer with this value. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to the QSIG variant.
[ISDNJapanNTTTimerT3JA]	Defines the T3_JA timer (in seconds). This parameter overrides the internal PSTN T301 timeout on the Users Side (TE side). If an outgoing call from the device to ISDN is not answered during this timeout, the call is released. The valid range is 10 to 240. The default is 50. **Notes:** ▪ This timer is also affected by the parameter PSTNAlertTimeout. ▪ This parameter is applicable only to the Japan NTT PRI variant (ProtocolType = 16).
Web/EMS: Trace Level **[TraceLevel]**	Defines the trace level: ▪ **[0]** No Trace (default) ▪ **[1]** Full ISDN Trace ▪ **[2]** Layer 3 ISDN Trace

Parameter	Description
	▪ **[3]** Only ISDN Q.931 Messages Trace ▪ **[4]** Layer 3 ISDN No Duplication Trace
Web/EMS: Framing Method CLI: framing [FramingMethod]	Determines the physical framing method for the trunk. ▪ [0] Extended Super Frame = (Default) Depends on protocol type: ✓ E1: E1 CRC4 MultiFrame Format extended G.706B (same as c) ✓ T1: T1 Extended Super Frame with CRC6 (same as D) ▪ [1] Super Frame = T1 SuperFrame Format (as B). ▪ [a] E1 FRAMING DDF = E1 DoubleFrame Format - CRC4 is forced to off ▪ [b] E1 FRAMING MFF CRC4 = E1 CRC4 MultiFrame Format - CRC4 is always on ▪ [c] E1 FRAMING MFF CRC4 EXT = E1 CRC4 MultiFrame Format extended G.706B - auto negotiation is on. If the negotiation fails, it changes automatically to CRC4 off (ddf) ▪ [A] T1 FRAMING F4 = T1 4-Frame multiframe. ▪ [B] T1 FRAMING F12 = T1 12-Frame multiframe (D4). ▪ [C] T1 FRAMING ESF = T1 Extended SuperFrame without CRC6 ▪ [D] T1 FRAMING ESF CRC6 = T1 Extended SuperFrame with CRC6 ▪ [E] T1 FRAMING F72 = T1 72-Frame multiframe (SLC96) ▪ [F] T1 FRAMING ESF CRC6 J2 = J1 Extended SuperFrame with CRC6 (Japan) **Note:** This parameter is not configurable for BRI interfaces; the device automatically sets the BRI framing method. If the TerminationSide parameter is set to USER_TERMINATION_SIDE (0), ClockMaster is automatically set to 0.; if the TerminationSide parameter is set to NETWORK_TERMINATION_SIDE (1), ClockMaster is automatically set to 1.
[FramingMethod_x]	Same as the description for parameter FramingMethod, but for a specific trunk ID (where x denotes the Trunk ID and 0 is the first Trunk).
Web/EMS: Clock Master CLI: clock-master [ClockMaster]	Determines the Tx clock source of the E1/T1 line. ▪ [0] Recovered = (Default) Generate the clock according to the Rx of the E1/T1 line. ▪ [1] Generated = Generate the clock according to the internal TDM bus. **Notes:** ▪ The source of the internal TDM bus clock is determined by the parameter TDMBusClockSource. ▪ This parameter is applicable only to E1/T1 interfaces.
[ClockMaster_x]	Same as the description for parameter ClockMaster, but for a specific Trunk ID (where x denotes the Trunk ID and 0 is the first Trunk). **Note:** This parameter is applicable only to E1/T1 interfaces.

Parameter	Description
Web/EMS: Line Code CLI: line-code [LineCode]	Selects B8ZS or AMI for T1 spans, and HDB3 or AMI for E1 spans. ▪ [0] B8ZS = (Default) B8ZS line code (for T1 trunks only). ▪ [1] AMI = AMI line code. ▪ [2] HDB3 = HDB3 line code (for E1 trunks only). **Note:** This parameter is not configurable for BRI interfaces; the device automatically uses the Modified Alternate Mark Invert (MAMI) line code.
[LineCode_x]	Same as the description for parameter LineCode, but for a specific trunk ID (where 0 denotes the first trunk).
[AdminState]	Defines the administrative state for all trunks. ▪ **[0]** = Lock the trunk; stops trunk traffic to configure the trunk protocol type. ▪ **[1]** = Shutting down (read only). ▪ **[2]** = (Default) Unlock the trunk; enables trunk traffic. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ When the device is locked from the Web interface, this parameter changes to 0. ▪ To define the administrative state per trunk, use the TrunkAdministrativeState parameter.
[TrunkAdministrativeState_x]	Defines the administrative state per trunk, where *x* denotes the trunk number. ▪ **[0]** = Lock the trunk; stops trunk traffic to configure the trunk protocol type. ▪ **[1]** = shutting down (read only). ▪ **[2]** = (Default) Unlock the trunk; enables trunk traffic.
Web/EMS: Line Build Out Loss CLI: line-build-out-loss [LineBuildOut.Loss]	Defines the line build out loss for the selected T1 trunk. ▪ [0] 0 dB (default) ▪ [1] -7.5 dB ▪ [2] -15 dB ▪ [3] -22.5 dB **Note:** This parameter is applicable only to T1 trunks.
[TDMHairPinning]	Defines static TDM hair-pinning (cross-connection) performed at initialization. The connection is between trunks with an option to exclude a single B-channel in each trunk. Format example: T0-T1/B3,T2-T3,T4-T5/B2. **Note:** For this parameter to take effect, a device reset is required.
Web: Enable TDM Tunneling EMS: TDM Over IP CLI: tdm-tunneling [EnableTDMoverIP]	Enables TDM tunneling. ▪ [0] Disable (default) ▪ [1] Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ TDM Tunneling is applicable only to E1/T1 interfaces (not BRI). ▪ For an overview on TDM tunneling, see TDM Tunneling on page 355.

Parameter	Description
CLI: iso8859-charset **[ISO8859CharacterSet]**	Defines the ISO 8859 character set type (languages) for representing the alphanumeric string of the calling name (caller ID) in the forwarded message, for IP-to-Tel and Tel-to-IP calls. ▪ **[0]** No Accented = Proprietary method where incoming INVITE messages with any accented characters (e.g., ב, י, ם, ףּ, and □), which are represented in a 2-byte unicode character, are translated to Latin-only, which are normal one-byte ASCII characters (a, e, i, o, and u, respectively). ▪ **[1]** Western European (Default) ▪ **[2]** Central European ▪ **[3]** South European ▪ **[4]** North European ▪ **[5]** Cyrillic ▪ **[6]** Arabic ▪ **[7]** Hebrew ▪ **[8]** Turkish

67.10.6.2 TDM Bus and Clock Timing Parameters

The TDM Bus parameters are described in the table below.

Table 67-59: TDM Bus and Clock Timing Parameters

Parameter	Description
TDM Bus Parameters	
Web/EMS: PCM Law Select **[PCMLawSelect]**	Determines the type of pulse-code modulation (PCM) companding algorithm law in input and output TDM bus. ▪ **[1]** Alaw ▪ **[3]** MuLaw The default value is automatically selected according to the Protocol Type of the selected trunk: E1 defaults to ALaw, T1 defaults to MuLaw. If the Protocol Type is set to NONE, the default is MuLaw. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ Typically, A-Law is used for E1 spans and Mu-Law for T1/J1 spans.
Web/EMS: Idle PCM Pattern CLI: idle-pcm-pattern **[IdlePCMPattern]**	Defines the PCM Pattern that is applied to the E1/T1 timeslot (B-channel) when the channel is idle. The range is 0 to 255. The default is set internally according to the Law select 1 (0xFF for Mu-Law; 0x55 for A-law). **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: Idle ABCD Pattern **[IdleABCDPattern]**	Defines the ABCD (CAS) Pattern that is applied to the CAS signaling bus when the channel is idle. The valid range is 0x0 to 0xF. The default is -1 (i.e., default pattern is 0000). **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only when using PSTN interface

Parameter	Description
	with CAS protocols.
Web/EMS: TDM Bus Clock Source **[TDMBusClockSource]**	Defines the clock source to which the device synchronizes. ▪ **[1]** Internal = (Default) Generate clock from local source. ▪ **[4]** Network = Recover clock from PSTN line. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: TDM Bus Local Reference **[TDMBusLocalReference]**	Defines the physical Trunk ID from which the device recovers (receives) its clock synchronization. The range is 0 to the maximum number of Trunks. The default is 0. **Note:** This parameter is applicable only if the parameter TDMBusClockSource is set to 4 and the parameter TDMBusPSTNAutoClockEnable is set to 0.
Web/EMS: TDM Bus Enable Fallback **[TDMBusEnableFallback]**	Defines the automatic fallback of the clock. ▪ **[0]** Manual (default) ▪ **[1]** Auto Non-Revertive ▪ **[2]** Auto Revertive
Web: TDM Bus Fallback Clock Source EMS: TDM Bus Fallback Clock **[TDMBusFallbackClock]**	Determines the fallback clock source on which the device synchronizes in the event of a clock failure. ▪ **[4]** Network (default) ▪ **[8]** H.110_A ▪ **[9]** H.110_B ▪ **[10]** NetReference1 ▪ **[11]** NetReference2
Web/EMS: TDM Bus Net Reference Speed **[TDMBusNetrefSpeed]**	Defines the NetRef frequency (for both generation and synchronization). ▪ **[0]** 8 kHz (default) ▪ **[1]** 1.544 MHz ▪ **[2]** 2.048 MHz
Web: TDM Bus PSTN Auto FallBack Clock EMS: TDM Bus Auto Fall Back Enable **[TDMBusPSTNAutoClockEnable]**	Enables the PSTN trunk Auto-Fallback Clock feature. ▪ **[0]** Disable = (Default) Recovers the clock from the E1/T1 line defined by the parameter TDMBusLocalReference. ▪ **[1]** Enable = Recovers the clock from any connected synchronized slave E1/T1 line. If this trunk loses its synchronization, the device attempts to recover the clock from the next trunk. Note that initially, the device attempts to recover the clock from the trunk defined by the parameter TDMBusLocalReference. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only if the TDMBusClockSource parameter is set to 4.
Web: TDM Bus PSTN Auto Clock Reverting EMS: TDM Bus Auto Fall Back Reverting Enable **[TDMBusPSTNAutoClockRevertingEnable]**	Enables the PSTN trunk Auto-Fallback Reverting feature. If enabled and a trunk returning to service has an AutoClockTrunkPriority parameter value that is higher than the priority of the local reference trunk (set in the TDMBusLocalReference parameter), the local reference reverts to the trunk with the higher priority that has returned to service for the device's clock source.

Parameter	Description
	- **[0]** Disable (default) - **[1]** Enable **Notes:** - For this parameter to take effect, a device reset is required. - This parameter is applicable only when the TDMBusPSTNAutoClockEnable parameter is set to 1.
Web: Auto Clock Trunk Priority EMS: Auto Trunk Priority CLI: clock-priority **[AutoClockTrunkPriority]**	Defines the trunk priority for auto-clock fallback (per trunk parameter). The valid range is 0 to 100, where 0 (default) is the highest priority and 100 indicates that the device does not perform a fallback to the trunk (typically, used to mark untrusted source of clock). **Note:** Fallback is enabled when the TDMBusPSTNAutoClockEnable parameter is set to 1.

67.10.6.3 CAS Parameters

The Common Channel Associated (CAS) parameters are described in the table below. Note that CAS is not applicable to BRI interfaces.

Table 67-60: CAS Parameters

Parameter	Description
Web: CAS Transport Type EMS: CAS Relay Transport Mode CLI: CAS-transport-type **[CASTransportType]**	Determines the ABCD signaling transport type over IP. - **[0]** CAS Events Only = (Default) Disable CAS relay. - **[1]** CAS RFC2833 Relay = Enable CAS relay mode using RFC 2833. The CAS relay mode can be used with the TDM tunneling feature to enable tunneling over IP for both voice and CAS signaling bearers.
[CASAddressingDelimiters]	Enables the addition of delimiters to the received address or received ANI digits string. - **[0]** = (default) Disable. The address and ANI strings remain without delimiters. - **[1]** = Enable. Delimiters such as '*', '#', and 'ST' are added to the received address or received ANI digits string.
CLI: cas-delimiters-types **[CASDelimitersPaddingUsage]**	Defines the digits string delimiter padding usage per trunk. - **[0]** = (Default) Default address string padding: '*XXX#' (where XXX is the digit string that begins with '*' and ends with '#', when using padding). - **[1]** = Special use of asterisks delimiters: '*XXX*YYY*' (where XXX is the address, YYY is the source phone number, and '*' is the only delimiter padding). **Note:** For this parameter to take effect, a device reset is required.
Web: CAS Table per Trunk EMS: Trunk CAS Table Index CLI: cas-table-index **[CASTableIndex_x]**	Defines the CAS protocol per trunk from a list of CAS protocols defined by the parameter CASFileName_x. For example, the below configuration specifies Trunks 0 and 1 to use the E&M Winkstart CAS (E_M_WinkTable.dat) protocol, and Trunks 2 and 3 to use the E&M Immediate Start CAS (E_M_ImmediateTable.dat) protocol:

Parameter	Description

	`CASFileName_0 = 'E_M_WinkTable.dat'`
	`CASFileName_1 = 'E_M_ImmediateTable.dat'`
	`CASTableIndex_0 = 0`
	`CASTableIndex_1 = 0`
	`CASTableIndex_2 = 1`
	`CASTableIndex_3 = 1`
	Notes:
	▪ You can define CAS tables per B-channel using the parameter CASChannelIndex.
	▪ The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.

Web: Dial Plan
EMS: Dial Plan Name
CLI: cas-dial-plan-name
[CASTrunkDialPlanName_x]
	Defines the CAS Dial Plan name per trunk.
	The range is up to 11 characters.
	For example, the below configures E1_MFCR2 trunk with a single protocol (Trunk 5):
	`ProtocolType_5 = 7`
	`CASFileName_0='R2_Korea_CP_ANI.dat'`
	`CASTableIndex_5 = 0`
	`DialPlanFileName = 'DialPlan_USA.dat'`
	`CASTrunkDialPlanName_5 = 'AT_T'`
	`Note:` `The` *`x`* `in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.`

[CASFileName_x]	Defines the CAS file name (e.g., 'E_M_WinkTable.dat') that defines the CAS protocol, where x denotes the CAS file ID (0-7). It is possible to define up to eight different CAS files by repeating this parameter. Each CAS file can be associated with one or more of the device's trunks, using the parameter CASTableIndex_x. **Note:** For this parameter to take effect, a device reset is required.

Web/EMS: CAS Table per Channel
CLI: cas-channel-index
[CASChannelIndex]
	Defines the loaded CAS protocol table index per B-channel pertaining to a CAS trunk. This parameter is assigned a string value and can be set in one of the following two formats:
	▪ **CAS table per channel:** Each channel is separated by a comma and the value entered denotes the CAS table index used for that channel. The syntax is <CAS index>,<CAS index> (e.g., "1,2,1,2…"). For this format, 31 indices must be defined for E1 trunks (including dummy for B-channel 16), or 24 indices for T1 trunks. Below is an example for configuring a T1 CAS trunk (Trunk 5) with several CAS variants:
	`ProtocolType_5 = 7`
	`CASFILENAME_0='E_M_FGBWinkTable.dat'`
	`CASFILENAME_1='E_M_FGDWinkTable.dat'`
	`CASFILENAME_2='E_M_WinkTable.txt'`
	`CasChannelIndex_5 =`
	`‘0,0,0,1,1,1,2,2,2,0,0,0,1,1,1,0,1,2,0,2,1,2,2`
	`,2’`
	`CASDelimitersPaddingUsage_5 = 1`
	▪ **CAS table per channel group:** Each channel group is separated by a colon and each channel is separated by a comma. The syntax is <x-y channel range>:<CAS table index>, (e.g., "1-10:1,11-31:3"). Every B-channel (including 16 for E1) must belong to a channel group. Below is an example for configuring an E1 CAS trunk (Trunk 5) with several CAS

Parameter	Description

	variants:
	`ProtocolType_5 = 8`
	`CASFILENAME_2='E1_R2D'`
	`CASFILENAME_7= E_M_ImmediateTable_A-Bit.txt'`
	`CasChannelIndex_5 = ‘1-10:2,11-20:7,21-31:2’`
	Notes:
	▪ To configure this parameter, the trunk must first be stopped.
	▪ Only one of these formats can be implemented; not both.
	▪ When this parameter is not configured, a single CAS table for the entire trunk is used, configured by the parameter CASTableIndex.

[CASTablesNum]	Defines how many CAS protocol configurations files are loaded. The valid range is 1 to 8. **Note:** For this parameter to take effect, a device reset is required.
CAS State Machines Parameters **Note:** For configuring the CAS State Machine table using the Web interface, see "Configuring CAS State Machines" on page 352. The CAS state machine can be configured only through the Web-based management tool.	
Web: Generate Digit On Time **[CASStateMachineGenerateDigitOnTime]**	Generates digit on-time (in msec). The value must be a positive value. The default is -1.
Web: Generate Inter Digit Time **[CASStateMachineGenerateInterDigitTime]**	Generates digit off-time (in msec). The value must be a positive value. The default is -1.
Web: DTMF Max Detection Time **[CASStateMachineDTMFMaxOnDetectionTime]**	Detects digit maximum on time (according to DSP detection information event) in msec units. The value must be a positive value. The default is -1.
Web: DTMF Min Detection Time **[CASStateMachineDTMFMinOnDetectionTime]**	Detects digit minimum on time (according to DSP detection information event) in msec units. The digit time length must be longer than this value to receive a detection. Any number may be used, but the value must be less than CasStateMachineDTMFMaxOnDetectionTime. The value must be a positive value. The default is -1.
Web: MAX Incoming Address Digits **[CASStateMachineMaxNumOfIncomingAddressDigits]**	Defines the limitation for the maximum address digits that need to be collected. After reaching this number of digits, the collection of address digits is stopped. The value must be an integer. The default is -1.
Web: MAX Incoming ANI Digits **[CASStateMachineMaxNumOfIncomingANIDigits]**	Defines the limitation for the maximum ANI digits that need to be collected. After reaching this number of digits, the collection of ANI digits is stopped. The value must be an integer. The default is -1.
Web: Collect ANI **[CASStateMachineCollectANI]**	In some cases, when the state machine handles the ANI collection (not related to MFCR2), you can enable the state machine to collect ANI or discard ANI. ▪ **[0]** No = Don't collect ANI. ▪ **[1]** Yes = Collect ANI. ▪ **[-1]** Default = Default value.

Parameter	Description
Web: Digit Signaling System **[CASStateMachineDigitSignalingSystem]**	Defines which Signaling System to use in both directions (detection\generation). ▪ **[0]** DTMF = DTMF signaling. ▪ **[1]** MF = (Default) MF signaling. ▪ **[-1]** Default = Default value.

67.10.6.4 ISDN Parameters

The ISDN parameters are described in the table below.

Table 67-61: ISDN Parameters

Parameter	Description
Web: ISDN Termination Side EMS: Termination Side CLI: isdn-termination-side **[TerminationSide]**	Determines the ISDN termination side. ▪ [0] User side = (Default) ISDN User Termination Equipment (TE) side. ▪ [1] Network side = ISDN Network Termination (NT) side. **Note:** Select 'User side' when the PSTN or PBX side is configured as 'Network side' and vice versa. If you don't know the device's ISDN termination side, choose 'User side'. If the D-channel alarm is indicated, choose 'Network Side'. The BRI module supports the ITU-T I.430 standard, which defines the ISDN-BRI layer 1 specification. The BRI and PRI ports are configured similarly, using this parameter. When an NT port is active, it drives a 38-V line and sends an INFO1 signal (as defined in ITU-T I.430 Table 4) on the data line to synchronize to a TE port that might be connected to it. To stop the voltage and the INFO1 signal on the line, stop the trunk using the Stop Trunk button.
[TerminationSide_x]	Same as the description for parameter TerminationSide, but for a specific trunk ID (where *x* denotes the Trunk ID and 0 is the first Trunk).
Web: Enable ignoring ISDN Disconnect with PI CLI: ign-isdn-disc-w-pi **[KeepISDNCallOnDisconnectWithPI]**	Enables the device to ignore ISDN Disconnect messages with PI 1 or 8. ▪ **[1]** = The call (in connected state) is not released if a Q.931 Disconnect with PI (PI = 1 or 8) message is received during the call. ▪ **[0]** = (Default) The call is disconnected.
Web: PI For Setup Message CLI: pi-4-setup-msg **[PIForSetupMsg]**	Determines whether and which Progress Indicator (PI) information element (IE) is added to the sent ISDN Setup message. Some ISDN protocols such as NI-2 or Euro ISDN can optionally contain PI = 1 or PI = 3 in the Setup message. ▪ **[0]** = PI is not added (default). ▪ **[1]** = PI 1 is added to a sent ISDN Setup message - call is not end-to-end ISDN. ▪ **[3]** = PI 3 is added to a sent ISDN Setup message - calling equipment is not ISDN.
Web/EMS: B-channel Negotiation CLI: b-ch-negotiation	Defines the ISDN B-channel negotiation mode. ▪ **[0]** Preferred

Parameter	Description
[BchannelNegotiation]	▪ **[1]** Exclusive (default) ▪ **[2]** Any **Notes:** ▪ For some ISDN variants, when 'Any' (2) is selected, the Setup message excludes the Channel Identification IE. ▪ The Any' (2) option is applicable only if the following conditions are met: ✓ The parameter TerminationSide is set to 0 ('User side'). ✓ The PSTN protocol type (ProtocolType) is configured as Euro ISDN.
ISDN Flexible Behavior Parameters ISDN protocol is implemented in different switches/PBXs by different vendors. Several implementations may vary slightly from the specification. Therefore, to provide a flexible interface that supports these ISDN variants, the ISDN behavior parameters can be used.	
Web/EMS: Incoming Calls Behavior CLI: isdn-bits-incoming-calls-behavior **[ISDNInCallsBehavior]**	Determines the bit-field used to determine several behavior options that influence how the ISDN Stack INCOMING calls behave. ▪ **[32]** DATA CONN RS = The device automatically sends a Q.931 Connect (answer) message on incoming Tel calls (Q.931 Setup). ▪ **[64]** VOICE CONN RS = The device sends a Connect (answer) message on incoming Tel calls. ▪ **[2048]** CHAN ID IN FIRST RS = (Default) The device sends Channel ID in the first response to an incoming Q.931 Call Setup message. Otherwise, the Channel ID is sent only if the device requires changing the proposed Channel ID. ▪ **[4096]** USER SETUP ACK = The Setup Ack message is sent by the SIP Gateway application layer and not automatically by the PSTN stack. By default, this bit is set. ▪ **[8192]** CHAN ID IN CALL PROC = The device sends Channel ID in a Q.931 Call Proceeding message. ▪ **[65536]** PROGR IND IN SETUP ACK = The device includes Progress Indicator (PI=8) in Setup Ack message if an empty called number is received in an incoming Setup message. This option is applicable to the overlap dialing mode. The device also plays a dial tone (for TimeForDialTone) until the next called number digits are received. By default, this is bit is set. ▪ **[2147483648]** USER SCREEN INDICATOR = When the device receives two Calling Number IE's in the Setup message, the device, by default, uses only one of the numbers according to the following: ✓ Network provided, Network provided - the first calling number is used ✓ Network provided, User provided: the first one is used ✓ User provided, Network provided: the second one is used ✓ User provided, user provided: the first one is used When this bit is configured, the device behaves as

Parameter	Description
	follows: ✓ Network provided, Network provided: the first calling number is used ✓ Network provided, User provided: the second one is used ✓ User provided, Network provided: the first one is used ✓ User provided, user provided: the first one is used **Note:** When using the *ini* file to configure the device to support several ISDNInCallsBehavior features, enter a summation of the individual feature values. For example, to support both [2048] and [65536] features, set ISDNInCallsBehavior = 67584 (i.e., 2048 + 65536).
[ISDNInCallsBehavior_x]	Same as the description for the parameter ISDNInCallsBehavior, but per trunk (i.e., where *x* denotes the Trunk ID).
Web/EMS: Q.931 Layer Response Behavior CLI: isdn-bits-ns-behavior **[ISDNIBehavior]**	Bit-field used to determine several behavior options that influence the behaviour of the Q.931 protocol. ▪ **[0]** = Disable (default). ▪ **[1]** NO STATUS ON UNKNOWN IE = Q.931 Status message isn't sent if Q.931 received message contains an unknown/unrecognized IE. By default, the Status message is sent. Note: This value is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[2]** NO STATUS ON INV OP IE = Q.931 Status message isn't sent if an optional IE with invalid content is received. By default, the Status message is sent. **Note:** This option is applicable only to ISDN variants in which sending of Status message is optional. ▪ **[4]** ACCEPT UNKNOWN FAC IE = Accepts unknown/unrecognized Facility IE. Otherwise, the Q.931 message that contains the unknown Facility IE is rejected (default). **Note:** This option is applicable only to ISDN variants where a complete ASN1 decoding is performed on Facility IE. ▪ **[128]** SEND USER CONNECT ACK = The Connect ACK message is sent in response to received Q.931 Connect; otherwise, the Connect ACK is not sent. **Note:** This option is applicable only to Euro ISDN User side outgoing calls. ▪ **[512]** EXPLICIT INTERFACE ID = Enables configuration of T1 NFAS Interface ID (refer to the parameter ISDNNFASInterfaceID_x). **Note:** This value is applicable only to 4/5ESS, DMS, NI-2 and HKT variants. ▪ **[2048]** ALWAYS EXPLICIT = Always set the Channel Identification IE to explicit Interface ID, even if the B-channel is on the same trunk as the D-channel. **Note:** This value is applicable only to 4/5ESS, DMS and NI-2 variants. ▪ **[32768]** ACCEPT MU LAW =Mu-Law is also accepted in

Parameter	Description
	ETSI. ▪ **[65536]** EXPLICIT PRES SCREENING = The calling party number (octet 3a) is always present even when presentation and screening are at their default. **Note:** This option is applicable only to ETSI, NI-2, and 5ESS. ▪ **[131072]** STATUS INCOMPATIBLE STATE = Clears the call on receipt of Q.931 Status with incompatible state. Otherwise, no action is taken (default). ▪ **[262144]** STATUS ERROR CAUSE = Clear call on receipt of Status according to cause value. ▪ **[524288]** ACCEPT A LAW =A-Law is also accepted in 5ESS. ▪ **[2097152]** RESTART INDICATION = Upon receipt of a Restart message, acEV_PSTN_RESTART_CONFIRM is generated. ▪ **[4194304]** FORCED RESTART = On data link (re)initialization, send RESTART if there is no call. ▪ **[67108864]** NS ACCEPT ANY CAUSE = Accept any Q.850 Cause IE from ISDN. **Note:** This option is applicable only to Euro ISDN. ▪ [134217728] NS_BRI_DL_ALWAYS_UP (0x08000000) = By default, the BRI D-channel goes down if there are no active calls. If this option is configured, the BRI D-channel is always up and synchronized. ▪ **[536870912]** = Alcatel coding for redirect number and display name is accepted by the device. Note: This option is applicable only to QSIG (and relevant for specific Alcatel PBXs such as OXE). ▪ **[1073741824]** QSI ENCODE INTEGER = If this bit is set, INTEGER ASN.1 type is used in operator coding (compliant to new ECMA standards); otherwise, OBJECT IDENTIFIER ASN.1 type is used. **Note:** This option is applicable only to QSIG. ▪ **[2147483648]** 5ESS National Mode For Bch Maintenance = Use the National mode of AT&T 5ESS for B-channel maintenance. **Notes:** ▪ To configure the device to support several ISDNIBehavior features, enter a summation of the individual feature values. For example, to support both [512] and [2048] features, set the parameter ISDNIBehavior is set to 2560 (i.e., 512 + 2048). ▪ When configuring in the Web interface, to select the options click the arrow button and then for each required option select 1 to enable. ▪ For BRI terminal endpoint identifier (TEI) configuration, instead of using the ISDNIBehavior parameter, use the following parameters: BriTEIConfigP2P_x, BriTEIConfigP2MP_x, BriTEIAssignTrigger_x, and BriTEIRemoveTrigger_x.
[ISDNIBehavior_x]	Same as the description for parameter ISDNIBehavior, but for a specific trunk ID.

Parameter	Description
Web: General Call Control Behavior EMS: General CC Behavior CLI: isdn-bits-cc-behavior **[ISDNGeneralCCBehavior]**	Bit-field for determining several general CC behavior options. To select the options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** = Data calls with interworking indication use 64 kbps B-channels (physical only). ▪ **[8]** REVERSE CHAN ALLOC ALGO = Channel ID allocation algorithm. ▪ **[16]** = The device clears down the call if it receives a NOTIFY message specifying 'User-Suspended'. A NOTIFY (User-Suspended) message is used by some networks (e.g., in Italy or Denmark) to indicate that the remote user has cleared the call, especially in the case of a long distance voice call. ▪ **[32]** CHAN ID 16 ALLOWED = Applies only to ETSI E1 lines (30B+D). Enables handling the differences between the newer QSIG standard (ETS 300-172) and other ETSI-based standards (ETS 300-102 and ETS 300-403) in the conversion of B-channel ID values into timeslot values: ✓ In 'regular ETSI' standards, the timeslot is identical to the B-channel ID value, and the range for both is 1 to 15 and 17 to 31. The D-channel is identified as channel-id #16 and carried into the timeslot #16. ✓ In newer QSIG standards, the channel-id range is 1 to 30, but the timeslot range is still 1 to 15 and 17 to 31. The D-channel is not identified as channel-id #16, but is still carried into the timeslot #16. When this bit is set, the channel ID #16 is considered as a valid B-channel ID, but timeslot values are converted to reflect the range 1 to 15 and 17 to 31. This is the new QSIG mode of operation. When this bit is not set (default), the channel_id #16 is not allowed, as for all ETSI-like standards. ▪ **[64]** USE T1 PRI = PRI interface type is forced to T1. ▪ **[128]** USE E1 PRI = PRI interface type is forced to E1. ▪ **[256]** START WITH B CHAN OOS = B-channels start in the Out-Of-Service state (OOS). ▪ **[512]** CHAN ALLOC LOWEST = CC allocates B-channels starting from the lowest available B-channel id. ▪ **[1024]** CHAN ALLOC HIGHEST = CC allocates B-channels starting from the highest available B-channel id. ▪ **[16384]** CC_TRANSPARENT_UUI bit: The UUI-protocol implementation of CC is disabled allowing the application to freely send UUI elements in any primitive, regardless of the UUI-protocol requirements (UUI Implicit Service 1). This allows more flexible application control on the UUI. When this bit is not set (default behavior), CC implements the UUI-protocol as specified in the ETS 300-403 standards for Implicit Service 1. ▪ **[65536]** GTD5 TBCT = CC implements the VERIZON-GTD-5 Switch variant of the TBCT Supplementary Service, as specified in FSD 01-02-40AG Feature Specification Document from Verizon. Otherwise, TBCT is implemented as specified in GR-2865-CORE

Parameter	Description
	specification (default behavior). **Note:** When using the *ini* file to configure the device to support several ISDNGeneralCCBehavior features, add the individual feature values. For example, to support both [16] and [32] features, set ISDNGeneralCCBehavior = 48 (i.e., 16 + 32).
Web/EMS: Outgoing Calls Behavior CLI: isdn-bits-outgoing-calls-behavior **[ISDNOutCallsBehavior]**	Determines several behaviour options (bit fields) that influence the behaviour of the ISDN Stack outgoing calls. To select options, click the arrow button, and then for each required option, select 1 to enable. The default is 0 (i.e., disable). ▪ **[2]** USER SENDING COMPLETE =The default behavior of the device (when this bit is not set) is to automatically generate the Sending-Complete IE in the Setup message. This behavior is used when overlap dialing is not needed. When overlap dialing is needed, set this bit and the behavior is changed to suit the scenario, i.e., Sending-Complete IE is added when required in the Setup message for Enblock mode or in the last Digit with Overlap mode. ▪ **[16]** USE MU LAW = The device sends G.711-m-Law in outgoing voice calls. When disabled, the device sends G.711-A-Law in outgoing voice calls. **Note:** This option is applicable only to the Korean variant. ▪ **[128]** DIAL WITH KEYPAD = The device uses the Keypad IE to store the called number digits instead of the CALLED_NB IE. **Note:** This option is applicable only to the Korean variant (Korean network). This is useful for Korean switches that don't accept the CALLED_NB IE. ▪ **[256]** STORE CHAN ID IN SETUP = The device forces the sending of a Channel-Id IE in an outgoing Setup message even if it's not required by the standard (i.e., optional) and no Channel-Id has been specified in the establishment request. This is useful for improving required compatibility with switches. On BRI lines, the Channel-Id IE indicates 'any channel'. On PRI lines it indicates an unused channel ID, preferred only. ▪ **[572]** USE A LAW = The device sends G.711 A-Law in outgoing voice calls. When disabled, the device sends the default G.711-Law in outgoing voice calls. **Note:** This option is applicable only to the E10 variant. ▪ **[1024]** = Numbering plan/type for T1 IP-to-Tel calling numbers are defined according to the manipulation tables or according to the RPID header (default). Otherwise, the plan/type for T1 calls are set according to the length of the calling number. ▪ **[2048]** = The device accepts any IA5 character in the called_nb and calling_nb strings and sends any IA5 character in the called_nb, and is not restricted to extended digits only (i.e., 0-9,*,#). ▪ **[16384]** DLCI REVERSED OPTION = Behavior bit used in the IUA interface groups to indicate that the reversed format of the DLCI field must be used. **Note:** When using the *ini* file to configure the device to

Parameter	Description
	support several ISDNOutCallsBehavior features, add the individual feature values. For example, to support both [2] and [16] features, set ISDNOutCallsBehavior = 18 (i.e., 2 + 16).
[ISDNOutCallsBehavior_x]	Same as the description for parameter ISDNOutCallsBehavior, but for a specific trunk ID.
Web: ISDN NS Behaviour 2 CLI: isdn-bits-ns-extension-behavior **[ISDNNSBehaviour2]**	Bit-field to determine several behavior options that influence the behavior of the Q.931 protocol. ▪ **[8]** NS BEHAVIOUR2 ANY UUI = Any User to User Information Element (UUIE) is accepted for any protocol discriminator. This is useful for interoperability with non-standard switches. ▪ **[16]** NS BEHAVIOUR2 DISPLAY = The Display IE is accepted even if it is not defined in the QSIG ISDN protocol standard. This is applicable only when configuration is QSI. ▪ **[64]** NS BEHAVIOUR2 FAC REJECT = When this bit is set, the device answers with a Facility IE message with the Reject component on receipt of Facility IE with unknown/invalid Invoke component. This bit is implemented in QSIG and ETSI variants.
[PSTNExtendedParams]	Determines the bit map for special PSTN behavior parameters: ▪ **[0]** = (Default) Applicable for NI-2 ISDN and QSIG "Networking Extensions". This bit (i.e., bit #0) is responsible for the Invoke ID size: ✓ If this bit is not set (default), then the Invoke ID size is always one byte, with a value of 01 to 7f. ✓ If this bit is set, then the Invoke ID size is one or two bytes according to the Invoke ID value. ▪ **[2]** = Applicable to the ROSE format (according to the old QSIG specifications). This bit (i.e., bit #1) is responsible for the QSIG octet 3. According to the ECMA-165 new version, octet 3 in all QSIG supplementary services Facility messages should be 0x9F = Networking Extensions. However, according to the old version, the value should be 0x91 = ROSE: ✓ If this bit is not set (default): 0x9F = Networking Extensions. ✓ If this bit is set: 0x91 = ROSE. ▪ **[3]** = Use options [0] and [2] above. **Note:** For this parameter to take effect, a device reset is required.
BRI Parameters	
BRI Layer 2 Mode CLI: isdn-layer2-mode [BriLayer2Mode]	Defines Point-to-Point (P2P) or Point-to-Multipoint (P2MP) mode for BRI ports. ▪ [0] Point to Point (default) ▪ [1] Point to Multipoint = Must be configured for Network side.
CLI: tei-config-p2p (config-voice >	Defines the BRI terminal endpoint identifier (TEI) when in

Parameter	Description
interface bri <module/port>) [BriTEIConfigP2P_x]	point-to-point (P2P) mode. The valid value is 0 to 63, 127. The default is 0. ▪ Network Side: ✓ 0-63: Static TEI is accepted. ✓ 127: Any possible TEI is accepted. Dynamic TEI allocation is supported. ▪ User Side: ✓ 0-63: Static TEI is used. ✓ 127: Dynamic TEI allocation is supported (TEI request procedure initiated). **Note:** The value 127 replaces the previous configuration requirement to set the ISDNIBehavior parameter to NS EXPLICIT INTERFACE ID (1).
CLI: tei-config-p2mp (config-voice > interface bri <module/port>) [BriTEIConfigP2MP_x]	Defines the BRI TEI when in point-to-multipoint (P2MP) mode. The valid value is 0 to 63, 127. The default is 127. ▪ Network Side: Not applicable - In network side in P2MP configuration, any TEI must be accepted. ▪ User Side: ✓ 0-63: Static TEI is used. ✓ 127: Dynamic TEI allocation is supported (TEI request procedure initiated).
CLI: tei-assign-trigger (config-voice > interface bri <module/port>) [BriTEIAssignTrigger_x]	Defines when to start the TEI assignment procedure. The valid values are (bit-field parameter): ▪ Bit#0: LAYER1_ACTIVATION ▪ Bit#1: BRI_PORT_CONFIG ▪ Bit#2: CALL_ESTABLISH The default is 0x02. **Note:** The parameter is applicable only to the User side (for Dynamic TEI).
CLI: tei-remove-trigger (config-voice > interface bri <module/port>) [BriTEIRemoveTrigger_x]	Defines the following: ▪ Network Side: When to "forget" all existing TEIs and wait for the User side to start a new TEI assignment procedure. This is also applicable to static TEI. ▪ User Side: When to start a new TEI assignment verification procedure. The valid values are (bit-field parameter): ▪ Bit#0: LAYER1_DEACTIVATION ▪ Bit#1: BRI_DL_RELEASED ▪ Bit#2: TEI_0_P2MP_NET_SIDE The default is 0x00.
NFAS Parameters (**Note:** These parameters are applicable only to PRI interfaces.)	
Web: NFAS Group Number EMS: Group Number CLI: isdn-nfas-group-number [NFASGroupNumber_x]	Defines the ISDN Non-Facility Associated Signaling (NFAS) group number (NFAS member), per trunk. ▪ [0] = (Default) Non-NFAS trunk. ▪ [1] to [12] = NFAS group number. Trunks that belong to the same NFAS group have the same

Parameter	Description
	number. With NFAS, you can use a single D-channel to control multiple PRI interfaces. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only to T1 ISDN protocols. ▪ The x in the ini file parameter name denotes the trunk number, where 0 is Trunk 1. ▪ For more information on NFAS, see ISDN Non-Facility Associated Signaling (NFAS) on page 360.
Web/EMS: D-channel Configuration CLI: isdn-nfas-dchannel-type [DChConfig_x]	Defines primary, backup (optional), and B-channels only, per trunk. ▪ [0] PRIMARY= (Default) Primary Trunk - contains a D-channel that is used for signaling. ▪ [1] BACKUP = Backup Trunk - contains a backup D-channel that is used if the primary D-channel fails. ▪ [2] NFAS = NFAS Trunk - contains only 24 B-channels, without a signaling D-channel. **Notes:** ▪ This parameter is applicable only to T1 ISDN protocols. ▪ The x in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
Web: NFAS Interface ID EMS: ISDN NFAS Interface ID CLI: isdn-nfas-interface-id [ISDNNFASInterfaceID_x]	Defines a different Interface ID per T1 trunk. The valid range is 0 to 100. The default interface ID equals the trunk's ID. **Notes:** ▪ To set the NFAS interface ID, configure ISDNIBehavior_x to include '512' feature per T1 trunk. ▪ The x in the ini file parameter name denotes the trunk number, where 0 is Trunk 1. ▪ For more information on NFAS, see ISDN Non-Facility Associated Signaling (NFAS) on page 360.

67.10.7 ISDN and CAS Interworking Parameters

The ISDN and CAS interworking parameters are described in the table below.

Table 67-62: ISDN and CAS Interworking Parameters

Parameter	Description
ISDN Parameters	
Web: Send Local Time To ISDN Connect **[SendLocalTimeToISDNConnect]**	Determines the device's handling of the date and time sent in the ISDN Connect message (Date / Time IE) upon receipt of SIP 200 OK messages. ▪ **[0]** Disable = (Default) If the SIP 200 OK includes the Date header, the device sends its value in the ISDN Connect Date / Time IE. If the 200 OK does not include this header, it does not add the Date / Time IE to the sent ISDN Connect message. ▪ **[1]** Enable = If the SIP 200 OK includes the Date header, the

Parameter	Description
	device sends its value (i.e. date and time) in the ISDN Connect Date / Time IE. If the 200 OK does not include this header, the device uses its internal, local date and time for the Date / Time IE, which it adds to the sent ISDN Connect message. ▪ **[2]** Always Send Local Date and Time = The device always sends its local date and time (obtained from its internal clock) to PBXs in ISDN Q.931 Connect messages (Date / Time IE). It does this regardless of whether or not the incoming SIP 200 OK includes the Date header. If the SIP 200 OK includes the Date header, the device ignores its value. **Notes:** ▪ This feature is applicable only to Tel-to-IP calls. ▪ For IP-to-Tel calls, this parameter is not applicable. Only if the incoming ISDN Connect message contains the Date / Time IE does the device add the Date header to the sent SIP 200 OK message.
Web/EMS: Min Routing Overlap Digits CLI: min-dg-b4-routing **[MinOverlapDigitsForRouting]**	Defines the minimum number of overlap digits to collect (for ISDN overlap dialing) before sending the first SIP message for routing Tel-to-IP calls. The valid value range is 0 to 49. The default is 1. **Note:** This parameter is applicable when the ISDNRxOverlap parameter is set to [2].
Web/EMS: ISDN Overlap IP to Tel Dialing CLI: isdn-tx-overlap **[ISDNTxOverlap]**	Enables ISDN overlap dialing for IP-to-Tel calls. This feature is part of ISDN-to-SIP overlap dialing according to RFC 3578. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, for each received INVITE of the same dialog session, the device sends an ISDN Setup (and subsequent ISDN Info Q.931 messages) with the collected digits to the Tel side. For all subsequent INVITEs received, the device sends a SIP 484 Address Incomplete response in order to maintain the current dialog session and receive additional digits from subsequent INVITEs. **Note:** When IP-to-Tel overlap dialing is enabled, to send ISDN Setup messages without the Sending Complete IE, the ISDNOutCallsBehavior parameter must be set to USER SENDING COMPLETE (2).
Web: Enable Receiving of Overlap Dialing CLI: ovrlp-rcving-type **[ISDNRxOverlap_x]**	Determines the receiving (Rx) type of ISDN overlap dialing for Tel-to-IP calls, per trunk. ▪ **[0]** None = (Default) Disabled. ▪ **[1]** Local receiving = ISDN Overlap Dialing - the complete number is sent in the INVITE Request-URI user part. The device receives ISDN called number that is sent in the 'Overlap' mode. The ISDN Setup message is sent to IP only after the number (including the Sending Complete IE) is fully received (via Setup and/or subsequent Info Q.931 messages). In other words, the device waits until it has received all the ISDN signaling messages containing parts of the called number, and only then it sends a SIP INVITE with the entire called number in the Request-URI. ▪ **[2]** Through SIP = Interworking of ISDN Overlap Dialing to SIP, based on RFC 3578. The device interworks ISDN to SIP by sending digits each time they are received (from Setup and subsequent Info Q.931 messages) to the IP, using subsequent

Parameter	Description
	SIP INVITE messages. **Notes:** - When option [2] is configured, you can define the minimum number of overlap digits to collect before sending the first SIP message for routing the call, using the MinOverlapDigitsForRouting parameter. - When option [2] is configured, even if SIP 4xx responses are received during this ISDN overlap receiving, the device does not release the call. - The MaxDigits parameter can be used to limit the length of the collected number for ISDN overlap dialing (if Sending Complete is not received). - If a digit map pattern is defined (using the DigitMapping or DialPlanIndex parameters), the device collects digits until a match is found (e.g., for closed numbering schemes) or until a timer expires (e.g., for open numbering schemes). If a match is found (or the timer expires), the digit collection process is terminated even if Sending Complete is not received. - For enabling ISDN overlap dialing for IP-to-Tel calls, use the ISDNTxOverlap parameter. - The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1. - For more information on ISDN overlap dialing, see "ISDN Overlap Dialing" on page 362.
CLI: ovrlp-rcving-type **[ISDNRxOverlap]**	Same as the description for parameter ISDNRxOverlap_x, but for all trunks.
Web/EMS: Mute DTMF In Overlap **[MuteDTMFInOverlap]**	Enables the muting of in-band DTMF detection until the device receives the complete destination number from the ISDN (for Tel-to-IP calls). In other words, the device does not accept DTMF digits received in the voice stream from the PSTN, but only accepts digits from ISDN Info messages. - **[0]** Don't Mute (default). - **[1]** Mute DTMF in Overlap Dialing = The device ignores in-band DTMF digits received during ISDN overlap dialing (disables the DTMF in-band detector). **Note:** This parameter is applicable to ISDN Overlap mode only when dialed numbers are sent using Q.931 Information messages.
[ConnectedNumberType]	Defines the Numbering Type of the ISDN Q.931 Connected Number IE that the device sends in the Connect message to the ISDN (for Tel-to-IP calls). This is interworked from the P-Asserted-Identity header in SIP 200 OK. The default is [0] (i.e., unknown).
[ConnectedNumberPlan]	Defines the Numbering Plan of the ISDN Q.931 Connected Number IE that the device sends in the Connect message to the ISDN (for Tel-to-IP calls). This is interworked from the P-Asserted-Identity header in SIP 200 OK. The default is [0] (i.e., unknown).
Web/EMS: Enable ISDN Tunneling Tel to IP CLI: isdn-tnl-tel2ip **[EnableISDNTunnelingTel2IP**	Enables ISDN Tunneling. - **[0]** Disable (default). - **[1]** Using Header = Enable ISDN Tunneling from ISDN PRI to

Parameter	Description
]	SIP using a proprietary SIP header. - **[2]** Using Body = Enable ISDN Tunneling from ISDN PRI to SIP using a dedicated message body. When ISDN Tunneling is enabled, the device sends all ISDN PRI messages using the correlated SIP messages. The ISDN Setup message is tunneled using SIP INVITE, all mid-call messages are tunneled using SIP INFO, and ISDN Disconnect/Release message is tunneled using SIP BYE messages. The raw data from the ISDN is inserted into a proprietary SIP header (X-ISDNTunnelingInfo) or a dedicated message body (application/isdn) in the SIP messages. **Notes:** - For this feature to function, you must set the parameter ISDNDuplicateQ931BuffMode to 128 (i.e., duplicate all messages). - ISDN tunneling is applicable for all ISDN variants as well as QSIG.
Web/EMS: Enable ISDN Tunneling IP to Tel CLI: isdn-tnl-ip2tel **[EnableISDNTunnelingIP2Tel]**	Enables ISDN Tunneling for IP-to-Tel calls. - **[0]** Disable (default) - **[1]** Enable ISDN Tunneling from IP to ISDN When ISDN Tunneling is enabled, the device extracts raw data received in the proprietary SIP header, x-isdntunnelinginfo, or a dedicated message body (application/isdn) in the SIP message and then sends the data in an ISDN message to the PSTN. If the raw data in this SIP header is suffixed with the string "ADDE", then the raw data is extracted and added as Informational Elements (IE) in the outgoing Q.931 message. The tunneling of the x-isdntunnelinginfo SIP header with IEs is converted from INVITE, 180, and 200 OK SIP messages to Q.931 SETUP, ALERT, and CONNECT respectively. For example, if the following SIP header is received, `x-isdntunnelinginfo: ADDE1C269FAA 06 800100820100A10F020136 0201F0A00702010102021F69` then it is added as an IE to the outgoing Q.931 message as 1C269FAA 06 800100820100A10F020136 0201F0A00702010102021F69, where, for example, "1C269F" is a 26 byte length Facility IE. **Note:** This feature is similar to that of the AddIEinSetup parameter. If both parameters are configured, the x-isdntunneling parameter takes precedence.
Web/EMS: Enable QSIG Tunneling CLI: qsig-tunneling **[EnableQSIGTunneling]**	Global parameter that enables QSIG tunneling-over-SIP for all calls. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableQSIGTunneling). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **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.
[QSIGTunnelingMode]	Defines the format of encapsulated QSIG message data in the SIP message MIME body. - **[0]** = (Default) ASCII presentation of Q.931 QSIG message. - **[1]** = Binary encoding of Q.931 QSIG message (according to ECMA-355, RFC 3204, and RFC 2025).

Parameter	Description
	Note: This parameter is applicable only if the QSIG Tunneling feature is enabled (using the EnableQSIGTunneling parameter).
Web: Enable Hold to ISDN EMS: Enable Hold 2 ISDN CLI: hold-to-isdn **[EnableHold2ISDN]**	Enables SIP-to-ISDN interworking of the Hold/Retrieve supplementary service. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ This parameter is applicable to Euro ISDN variants - from TE (user) to NT (network). ▪ This parameter is applicable to QSIG BRI. ▪ If the parameter is disabled, the device plays a held tone to the Tel side when a SIP request with 0.0.0.0 or "inactive" in SDP is received. An appropriate CPT file with the held tone should be used.
EMS: Duplicate Q931 Buff Mode **[ISDNDuplicateQ931BuffMode]**	Determines the activation/deactivation of delivering raw Q.931 messages. ▪ **[0]** = (Default) ISDN messages aren't duplicated. ▪ **[128]** = All ISDN messages are duplicated. **Note:** For this parameter to take effect, a device reset is required.
Web/EMS: ISDN SubAddress Format CLI: isdn-subaddr-frmt **[ISDNSubAddressFormat]**	Determines the encoding format of the SIP Tel URI parameter 'isub', which carries the encoding type of ISDN subaddresses. This is used to identify different remote ISDN entities under the same phone number (ISDN Calling and Called numbers) for interworking between ISDN and SIP networks. ▪ **[0]** = (Default) ASCII - IA5 format that allows up to 20 digits. Indicates that the 'isub' parameter value needs to be encoded using ASCII characters. ▪ **[1]** = BCD (Binary Coded Decimal) - allows up to 40 characters (digits and letters). Indicates that the 'isub' parameter value needs to be encoded using BCD when translated to an ISDN message. ▪ **[2]** = User Specified For IP-to-Tel calls, if the incoming SIP INVITE message includes subaddress values in the 'isub' parameter for the Called Number (in the Request-URI) and/or the Calling Number (in the From header), these values are mapped to the outgoing ISDN Setup message. If the incoming ISDN Setup message includes 'subaddress' values for the Called Number and/or the Calling Number, these values are mapped to the outgoing SIP INVITE message's 'isub' parameter in accordance with RFC 4715.
[IgnoreISDNSubaddress]	Determines whether the device ignores the Subaddress from the incoming ISDN Called and Calling numbers when sending to IP. ▪ **[0]** = (Default) If an incoming ISDN Q.931 Setup message contains a Called/Calling Number Subaddress, the Subaddress is interworked to the SIP 'isub' parameter according to RFC. ▪ **[1]** = The device removes the ISDN Subaddress and does not include the 'isub' parameter in the Request-URI and does not process INVITEs with this parameter.
[ISUBNumberOfDigits]	Defines the number of digits (from the end) that the device takes from the called number (received from the IP) for the isub number

Parameter	Description
	(in the sent ISDN Setup message). This feature is only applicable for IP-to-ISDN calls. The valid value range is 0 to 36. The default is 0. This feature operates as follows: 1 If an isub parameter is received in the Request-URI, for example, INVITE sip:9565645;**isub**=1234@host.domain:user=phone SIP/2.0 then the isub value is sent in the ISDN Setup message as the destination subaddress. 2 If the isub parameter is not received in the user part of the Request-URI, the device searches for it in the URI parameters of the To header, for example, To: "Alex" <sip: 9565645@host.domain;**isub**=1234> If present, the isub value is sent in the ISDN Setup message as the destination subaddress. 3 If the isub parameter is not present in the Request-URI header nor To header, the device does the following: ✓ If the called number (that appears in the user part of the Request-URI) starts with zero (0), for example, INVITE sip:**0**5694564@host.domain:user=phone SIP/2.0 then the device maps this called number to the destination number of the ISDN Setup message, and the destination subaddress in this ISDN Setup message remains empty. ✓ If the called number (that appears in the user part of the Request-URI) does not start with zero, for example, INVITE sip:5694564@host.domain:user=phone SIP/2.0 then the device maps this called number to the destination number of the ISDN Setup message, and the destination subaddress in this ISDN Setup message then contains *y* digits from the end of the called number. The *y* number of digits can be configured using the ISUBNumberOfDigits parameter. The default value of ISUBNumberOfDigits is 0, thus, if this parameter is not configured, and 1) and 2) scenarios (described above) have not provided an isub value, the subaddress remains empty.
Web: Default Cause Mapping From ISDN to SIP CLI: dflt-cse-map-isdn2sip **[DefaultCauseMapISDN2IP]**	Defines a single default ISDN release cause that is used (in ISDN-to-IP calls) instead of all received release causes, except when the following Q.931 cause values are received: Normal Call Clearing (16), User Busy (17), No User Responding (18), or No Answer from User (19). The range is any valid Q.931 release cause (0 to 127). The default is 0 (i.e., not configured - static mapping is used).
Web/EMS: Enable Calling Party Category CLI: ni2-cpc [EnableCallingPartyCategory]	Enables the mapping of the calling party category (CPC) between the incoming PSTN message and outgoing SIP message, and vice versa (i.e., for IP-to-Tel and Tel-to-IP calls). The CPC characterizes the station used to originate a call (e.g., a payphone or an operator). ▪ [0] Disable = (Default) CPC is not relayed between SIP and PSTN. ▪ [1] Enable The CPC is denoted in the PSTN message as follows: ▪ ISDN PRI NI-2: In the Originating Line Information (OLI) Information Element (IE) of the ISDN Setup message.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>
<ul><li>MFC-R2: ANI II digits. The device supports the Brazilian and Argentinian variants. This regional support is configured using the CallingPartyCategoryMode.</li></ul>
<p>The CPC is denoted in the SIP INVITE message using the 'cpc=' parameter in the From or P-Asserted-Identity headers. For example, the 'cpc=' parameter in the below INVITE message is set to "payphone":</p>
<pre>
INVITE sip:bob@biloxi.example.com SIP/2.0
To: "Bob" <sip:bob@biloxi.example.com>
From: <tel:+17005554141;cpc=payphone>;tag=1928301774
</pre>
<p>The table below shows the mapping of CPC between SIP and PSTN:</p>
<table>
<tr><th rowspan="2">SIP CPC</th><th rowspan="2">NI-2 PRI</th><th colspan="2">MFC-R2</th></tr>
<tr><th>Argentina</th><th>Brazil</th></tr>
<tr><td>ordinary</td><td>23</td><td>II-1</td><td>II-1</td></tr>
<tr><td>priority</td><td>n/a</td><td>II-2</td><td>II-2</td></tr>
<tr><td>data</td><td>n/a</td><td>II-6</td><td>II-6</td></tr>
<tr><td>test</td><td>n/a</td><td>II-3</td><td>II-3</td></tr>
<tr><td>operator</td><td>35</td><td>II-5</td><td>II-5</td></tr>
<tr><td>payphone</td><td>70</td><td>II-4</td><td>II-7</td></tr>
<tr><td>unknown</td><td>n/a</td><td>II-1</td><td>II-1</td></tr>
<tr><td>subscriber</td><td>23</td><td>n/a</td><td>II-1</td></tr>
<tr><td>cellular</td><td>61</td><td>II-13</td><td>n/a</td></tr>
<tr><td>locutorio</td><td>n/a</td><td>II-11</td><td>n/a</td></tr>
<tr><td>servicio-publico</td><td>n/a</td><td>II-12</td><td>n/a</td></tr>
<tr><td>red-privada-virtual / private-virtual-network</td><td>n/a</td><td>II-14</td><td>n/a</td></tr>
<tr><td>linea-especial / special-operator-handling-required</td><td>n/a</td><td>II-15</td><td>n/a</td></tr>
<tr><td>operadora-con-intervencion / telco-operator-handled-call</td><td>n/a</td><td>II-5</td><td>n/a</td></tr>
<tr><td>prison</td><td>29</td><td>n/a</td><td>n/a</td></tr>
<tr><td>hotel</td><td>66</td><td>n/a</td><td>n/a</td></tr>
<tr><td>cellular-roaming</td><td>63</td><td>n/a</td><td>n/a</td></tr>
</table>
<p>Note: This feature is applicable only to the NI-2 PRI and E1 MFC-R2 variants.</p>
</td></tr>
<tr><td>Web: Calling Party Category Mode
CLI: cpc-mode
[CallingPartyCategoryMode]</td><td>
<p>Defines the regional Calling Party Category (CPC) mapping variant between SIP and PSTN for MFC-R2.</p>
<ul><li>[0] None (default)</li><li>[1] Brazil R2</li><li>[2] Argentina R2</li></ul>
<p><b>Notes:</b></p>
<ul><li>To enable CPC mapping, set the EnableCallingPartyCategory parameter to 1.</li></ul>
</td></tr>
</table>

Parameter	Description
	▪ This parameter is applicable only to the E1 MFC-R2 variant.

CLI: usr2usr-hdr-frmt
[UserToUserHeaderFormat]
	Defines the interworking between the SIP INVITE's User-to-User header and the ISDN User-to-User (UU) IE data.
	▪ **[0]** = (Default) SIP header format: X-UserToUser.
	▪ **[1]** = SIP header format: User-to-User with Protocol Discriminator (pd) attribute (according to IETF Internet-Draft draft-johnston-sipping-cc-uui-04). For example:
	`User-to-User=3030373435313734313635353b313233343b3834;pd=4`
	▪ **[2]** = SIP header format: User-to-User with encoding=hex at the end and pd embedded as the first byte (according to IETF Internet-Draft draft-johnston-sipping-cc-uui-03). For example:
	`User-to-User=043030373435313734313635353b313233343b3834;encoding=hex`
	where "04" at the beginning of this message is the pd.
	▪ **[3]** = Interworks the SIP User-to-User header containing text format to ISDN UUIE in hexadecimal format, and vice versa. For example:
	SIP Header in text format:
	`User-to-User=01800213027b712a;NULL;4582166;`
	Translated to hexadecimal in the ISDN UUIE:
	`303138303032313330323762373132613b4e554c4c3b343538323136363b`
	The Protocol Discriminator (pd) used in UUIE is "04" (IUA characters).
	Note: This parameter is applicable for Tel-to-IP and IP-to-Tel calls.

Web/EMS: Remove CLI when Restricted CLI: rmv-cli-when-restr **[RemoveCLIWhenRestricted]**	Determines (for IP-to-Tel calls) whether the Calling Number and Calling Name IEs are removed from the ISDN Setup message if the presentation is set to Restricted. ▪ **[0]** No = (Default) IE's are not removed. ▪ **[1]** Yes = IE's are removed.
Web/EMS: Remove Calling Name CLI: rmv-calling-name **[RemoveCallingName]**	Enables the device to remove the Calling Name from SIP-to-ISDN calls for all trunks. ▪ **[0]** Disable = (Default) Does not remove Calling Name. ▪ **[1]** Enable = Removes Calling Name. **Note:** Some PSTN switches / PBXs may not be configured to support the receipt of the “Calling Name” information. These switches might respond to an ISDN Setup message (including the Calling Name) with an ISDN "REQUESTED_FAC_NOT_SUBSCRIBED" failure. This parameter can be set to Enable (1) to remove the “Calling Name” from SIP-to-ISDN calls and allow the call to proceed.
Web: Remove Calling Name EMS: Remove Calling Name For Trunk Mode **[RemoveCallingNameForTrunk_x]**	Enables the device to remove the Calling Name for SIP-to-ISDN calls, per trunk. ▪ **[-1]** Use Global Parameter = (Default) Settings of the global parameter RemoveCallingName are used. ▪ **[0]** Disable = Does not remove Calling Name. ▪ **[1]** Enable = Remove Calling Name.

Parameter	Description
	Note: The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
Web/EMS: Progress Indicator to ISDN CLI: pi-to-isdn **[ProgressIndicator2ISDN_x]**	Determines the Progress Indicator (PI) to ISDN per trunk. ▪ **[-1]** Not Configured = (Default) The PI in ISDN messages is set according to the parameter PlayRBTone2Tel. ▪ **[0]** No PI = PI is not sent to ISDN. ▪ **[1]** PI = 1; **[8]** PI = 8: The PI value is sent to PSTN in Q.931/Proceeding and Alerting messages. Typically, the PSTN/PBX cuts through the audio channel without playing local ringback tone, enabling the originating party to hear remote Call Progress Tones or network announcements. **Note:** The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
Web: Set PI in Rx Disconnect Message EMS: Set PI For Disconnect Msg CLI: pi-in-rx-disc-msg **[PIForDisconnectMsg_x]**	Defines the device's behavior per trunk when a Disconnect message is received from the ISDN before a Connect message is received. ▪ **[-1]** Not Configured = (Default) Sends a 183 SIP response according to the received progress indicator (PI) in the ISDN Disconnect message. If PI = 1 or 8, the device sends a 183 response, enabling the PSTN to play a voice announcement to the IP side. If there isn't a PI in the Disconnect message, the call is released. ▪ **[0]** No PI = Doesn't send a 183 response to IP. The call is released. ▪ **[1]** PI = 1; **[8]** PI = 8: Sends a 183 response to IP. **Note:** The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
EMS: Connect On Progress Ind **[ConnectOnProgressInd]**	Enables the play of announcements from IP to Tel without the need to answer the Tel-to-IP call. It can be used with PSTN networks that don't support the opening of a TDM channel before an ISDN Connect message is received. ▪ **[0]** = (Default) Connect message isn't sent after SIP 183 Session Progress message is received. ▪ **[1]** = Connect message is sent after SIP 183 Session Progress message is received.
Web: Local ISDN Ringback Tone Source EMS: Local ISDN RB Source CLI: local-isdn-rbt-src **[LocalISDNRBSource_x]**	Determines whether the ringback tone is played to the ISDN by the PBX/PSTN or by the device, per trunk. ▪ **[0]** PBX = (Default) PBX/PSTN plays the ringback tone. ▪ **[1]** Gateway = The device plays the ringback tone. **Notes:** ▪ This parameter is used together with the PlayRBTone2Trunk parameter. ▪ The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
Web/EMS: PSTN Alert Timeout CLI: pstn-alrt-timeout **[TrunkPSTNAlertTimeout_x]**	Defines the Alert Timeout (ISDN T301 timer) in seconds for outgoing calls to PSTN, per trunk. This timer is used between the time that an ISDN Setup message is sent to the Tel side (IP-to-Tel call establishment) and a Connect message is received. If Alerting is received, the timer is restarted. The range is 1 to 600. The default is 180.

Parameter	Description
	Note: The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
Web: B-Channel Negotiation EMS: B-Channel Negotiation For Trunk Mode **[BChannelNegotiationForTrunk_x]**	Determines the ISDN B-channel negotiation mode, per trunk. ▪ **[-1]** Not Configured = (Default) Use per device configuration of the BChannelNegotiation parameter. ▪ **[0]** Preferred. ▪ **[1]** Exclusive. ▪ **[2]** Any. **Notes:** ▪ The option **Any** is only applicable if TerminationSide is set to 0 (i.e., User side). ▪ The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
CLI: snd-isdn-ser-aftr-restart **[SendISDNServiceAfterRestart]**	Enables the device to send an ISDN SERVice message per trunk upon device reset. The messsage (transmitted on the trunk's D-channel) indicates the availability of the trunk's B-channels (i.e., trunk in service). ▪ **[0]** = Disable (default) ▪ **[0]** = Enable
EMS: Support Redirect InFacility **[SupportRedirectInFacility]**	Determines whether the Redirect Number is retrieved from the Facility IE. ▪ **[0]** = (Default) Not supported. ▪ **[1]** = Supports partial retrieval of Redirect Number (number only) from the Facility IE in ISDN Setup messages. This is applicable to Redirect Number according to ECMA-173 Call Diversion Supplementary Services. **Note:** To enable this feature, the parameter ISDNDuplicateQ931BuffMode must be set to 1.
CLI: call-re-rte-mode **[CallReroutingMode]**	Determines whether ISDN call rerouting (call forward) is performed by the PSTN instead of by the SIP side. This call forwarding is based on Call Deflection for Euro ISDN (ETS-300-207-1) and QSIG (ETSI TS 102 393). ▪ **[0]** Disable (default). ▪ **[1]** Enable = Enables ISDN call rerouting. When the device sends the INVITE message to the remote SIP entity and receives a SIP 302 response with a Contact header containing a URI host name that is the same as the device's IP address, the device sends a Facility message with a Call Rerouting invoke method to the ISDN and waits for the PSTN side to disconnect the call. **Note:** When this parameter is enabled, ensure that you configure in the Inbound IP Routing table (PSTNPrefix *ini* file parameter) a rule to route the redirected call (using the user part from the 302 Contact header) to the same Trunk Group from where the incoming Tel-to-IP call was received.
EMS: Enable CIC **[EnableCIC]**	Determines whether the Carrier Identification Code (CIC) is relayed to ISDN. ▪ **[0]** = (Default) Do not relay the Carrier Identification Code (CIC) to ISDN. ▪ **[1]** = CIC is relayed to the ISDN in Transit Network Selection (TNS) IE.

Parameter	Description
	If enabled, the CIC code (received in an INVITE Request-URI) is included in a TNS IE in the ISDN Setup message. For example: INVITE sip:555666;cic=2345@100.2.3.4 sip/2.0. **Notes:** ▪ This feature is supported only for SIP-to-ISDN calls. ▪ The parameter AddCicAsPrefix can be used to add the CIC as a prefix to the destination phone number for routing IP-to-Tel calls.
Web/EMS: AoC Support **[EnableAOC]**	Enables the interworking of ISDN Advice of Charge (AOC) messages to SIP. ▪ **[0]** Disable (default) ▪ **[1]** Enable For more information on AOC, see "Advice of Charge Services for Euro ISDN" on page 468.
Web: IPMedia Detectors EMS: DSP Detectors Enable CLI: IPM-detectors-enable **[EnableDSPIPMDetectors]**	Enables the device's DSP detectors. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable **Notes:** ▪ For this 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 621. ▪ When enabled (1), the number of available channels is reduced.
Web: Add IE in SETUP EMS: IE To Be Added In Q.931 Setup CLI: add-ie-in-setup **[AddIEinSetup]**	Global parameter that defines an optional Information Element (IE) data (in hex format) to add to ISDN Setup messages. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AddIEInSetup). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Trunk Groups to Send IE EMS: List Of Trunk Groups To Send IE CLI: trkgrps-to-snd-ie **[SendIEonTG]**	Defines Trunk Group IDs (up to 50 characters) from where the optional ISDN IE (defined by the parameter AddIEinSetup) is sent. For example: '1,2,4,10,12,6'. **Notes:** ▪ You can configure different IE data for Trunk Groups by defining this parameter for different IP Profile IDs (using the parameter IPProfile), and then assigning the required IP Profile ID in the Inbound IP Routing table (PSTNPrefix). ▪ When IP Profiles are used for configuring different IE data for Trunk Groups, this parameter is ignored.
Web: Enable User-to-User IE for Tel to IP EMS: Enable UUI Tel 2 Ip CLI: uui-ie-for-tel2ip **[EnableUUITel2IP]**	Enables transfer of User-to-User (UU) IE from ISDN PRI to SIP. ▪ **[0]** Disable (default) ▪ **[1]** Enable The device supports the following ISDN PRI-to-SIP interworking: Setup to SIP INVITE, Connect to SIP 200 OK, User Information to SIP INFO, Alerting to SIP 18x response, and Disconnect to SIP BYE response messages.

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td><b>Note:</b> The interworking of ISDN User-to-User IE to SIP INFO is applicable only to the Euro ISDN, QSIG, and 4ESS PRI variants.</td></tr>
<tr><td>Web: Enable User-to-User IE for IP to Tel
EMS: Enable UUI Ip 2 Tel
CLI: uui-ie-for-ip2tel
<b>[EnableUUIIP2Tel]</b></td><td>Enables interworking of SIP user-to-user information (UUI) to User-to-User IE in ISDN Q.931 messages.
▪ <b>[0]</b> Disable = (Default) Received UUI is not sent in ISDN message.
▪ <b>[1]</b> Enable = The device interworks UUI from SIP to ISDN messages. The device supports the following SIP-to-ISDN interworking of UUI:
✓ SIP INVITE to Q.931 Setup
✓ SIP REFER to Q.931 Setup
✓ SIP 200 OK to Q.931 Connect
✓ SIP INFO to Q.931 User Information
✓ SIP 18x to Q.931 Alerting
✓ SIP BYE to Q.931 Disconnect
<b>Notes:</b>
▪ The interworking of ISDN User-to-User IE to SIP INFO is applicable only to the Euro ISDN, QSIG, and 4ESS PRI variants.
▪ To interwork the UUIE header from SIP-to-ISDN messages with the 4ESS ISDN variant, the ISDNGeneralCCBehavior parameter must be set to 16384.</td></tr>
<tr><td><b>[Enable911LocationIdIP2Tel]</b></td><td>Enables interworking of Emergency Location Identification from SIP to PRI.
▪ <b>[0]</b> = Disabled (default)
▪ <b>[1]</b> = Enabled
When enabled, the From header received in the SIP INVITE is translated into the following ISDN IE's:
▪ Emergency Call Control.
▪ Generic Information - to carry the Location Identification Number information.
▪ Generic Information - to carry the Calling Geodetic Location information.
<b>Note:</b> This capability is applicable only to the NI-2 ISDN variant.</td></tr>
<tr><td>CLI: early-answer-timeout
<b>[EarlyAnswerTimeout]</b></td><td>Global parameter that defines the duration (in seconds) that the device waits for an ISDN Connect message from the called party (Tel side), started from when it sends a Setup message. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EarlyAnswerTimeout). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314.
<b>Note:</b> 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.</td></tr>
<tr><td>Web/EMS: Trunk Transfer Mode
CLI: trk-xfer-mode-type
<b>[TrunkTransferMode]</b></td><td>Determines the trunk transfer method (for all trunks) when a SIP REFER message is received. The transfer method depends on the Trunk's PSTN protocol (configured by the parameter ProtocolType) and is applicable only when one of these protocols are used:
<table>
<tr><th>PSTN Protocol</th><th>Transfer Method (Described Below)</th></tr>
<tr><td>E1 Euro ISDN [1]</td><td>ECT [2] or InBand [5]</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Parameter</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td>E1 QSIG [21],
T1 QSIG [23]</td><td>Single Step Transfer [4], Path Replacement Transfer [2], or InBand [5]</td></tr>
<tr><td>T1 NI2 ISDN [10],
T1 4ESS ISDN [11],
T1 5ESS 9 ISDN [12]</td><td>TBCT [2] or InBand [5]</td></tr>
<tr><td>T1 DMS-100 ISDN [14]</td><td>RTL [2] or InBand [5]</td></tr>
<tr><td>T1 RAW CAS [3], T1 CAS [2], E1 CAS [8], E1 RAW CAS [9]</td><td>[1] CAS NFA DMS-100 or [3] CAS Normal transfer</td></tr>
<tr><td>T1 DMS-100 Meridian ISDN [35]</td><td>RTL [2] or InBand [5]</td></tr>
</table>

The valid values of this parameter are described below:

- **[0]** = Not supported (default).
- **[1]** = Supports CAS NFA DMS-100 transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, waits for an acknowledged Wink from the remote side, dials the Refer-to number to the switch, and then releases the call.
  Note: A specific NFA CAS table is required.
- **[2]** = Supports ISDN (PRI/BRI) transfer - Release Link Trunk (RLT) (DMS-100), Two B Channel Transfer (TBCT) (NI2), Explicit Call Transfer (ECT) (EURO ISDN), and Path Replacement (QSIG). When a SIP REFER message is received, the device performs a transfer by sending Facility messages to the PBX with the necessary information on the call's legs to be connected. The different ISDN variants use slightly different methods (using Facility messages) to perform the transfer.
  Notes:
  - ✓ For RLT ISDN transfer, the parameter SendISDNTransferOnConnect must be set to 1.
  - ✓ The parameter SendISDNTransferOnConnect can be used to define if the TBCT/ECT transfer is performed after receipt of Alerting or Connect messages. For RLT, the transfer is always done after receipt of Connect (SendISDNTransferOnConnect is set to 1).
  - ✓ This transfer can be performed between B-channels from different trunks or Trunk Groups, by using the parameter EnableTransferAcrossTrunkGroups.
  - ✓ The device initiates the ECT process after receiving a SIP REFER message only for trunks that are configured to User side.
- **[3]** = Supports CAS Normal transfer. When a SIP REFER message is received, the device performs a Blind Transfer by executing a CAS Wink, dialing the Refer-to number to the switch, and then releasing the call.
- **[4]** = Supports QSIG Single Step transfer (PRI/BRI):
  IP-to-Tel: When a SIP REFER message is received, the device performs a transfer by sending a Facility message to the PBX, initiating Single Step transfer. Once a success return result is received, the transfer is completed.
  Tel-to-IP: When a Facility message initiating Single Step transfer
</td></tr>
</table>

Parameter	Description
	is received from the PBX, a SIP REFER message is sent to the IP side. ▪ **[5]** = IP-to-Tel Blind Transfer mode supported for ISDN (PRI/BRI) protocols and implemented according to AT&T Toll Free Transfer Connect Service (TR 50075) “Courtesy Transfer-Human-No Data”. When the device receives a SIP REFER message, it performs a blind transfer by first dialing the DTMF digits (transfer prefix) defined by the parameter XferPrefixIP2Tel (configured to "*8" for AT&T service), and then (after 500 msec) the device dials the DTMF of the number (referred) from the Refer-To header sip:URI userpart. If the hostpart of the Refer-To sip:URI contains the device's IP address, and if the Trunk Group selected according to the IP to Tel Routing table is the same Trunk Group as the original call, then the device performs the in-band DTMF transfer; otherwise, the device sends the INVITE according to regular transfer rules. After completing the in-band transfer, the device waits for the ISDN Disconnect message. If the Disconnect message is received during the first 5 seconds, the device sends a SIP NOTIFY with 200 OK message; otherwise, the device sends a NOTIFY with 4xx message. ▪ [6] = Supports AT&T toll free out-of-band blind transfer for trunks configured with the 4ESS ISDN protocol. AT&T courtesy transfer is a supplementary service which enables a user (e.g., user "A") to transform an established call between it and user "B" into a new call between users "B" and "C", whereby user "A" does not have a call established with user "C" prior to call transfer. The device handles this feature as follows: ✓ IP-to-Tel (user side): When a SIP REFER message is received, the device initiates a transfer by sending a Facility message to the PBX. ✓ Tel-to-IP (network side): When a Facility message initiating an out-of-band blind transfer is received from the PBX, the device sends a SIP REFER message to the IP side (if the EnableNetworkISDNTransfer parameter is set to 1). **Note:** For configuring trunk transfer mode per trunk, use the parameter TrunkTransferMode_x.
[TrunkTransferMode_x]	Determines the trunk transfer mode per trunk (where *x* denotes the Trunk number). For configuring trunk transfer mode for all trunks and for a description of the parameter options, refer to the parameter TrunkTransferMode.
[EnableTransferAcrossTrunkGroups]	Determines whether the device allows ISDN ECT, RLT or TBCT IP-to-Tel call transfers between B-channels of different Trunk Groups. ▪ **[0]** = (Default) Disable - ISDN call transfer is only between B-channels of the same Trunk Group. ▪ **[1]** = Enable - the device performs ISDN transfer between any two PSTN calls (between any Trunk Group) handled by the device. **Note:** The ISDN transfer also requires that you configure the parameter TrunkTransferMode_x to 2.
Web: ISDN Transfer Capabilities EMS: Transfer Capability To ISDN	Defines the IP-to-ISDN Transfer Capability of the Bearer Capability IE in ISDN Setup messages, per trunk. ▪ **[-1]** Not Configured

Parameter	Description
CLI: isdn-xfer-cab **[ISDNTransferCapability_x]**	▪ **[0]** Audio 3.1 (default) ▪ **[1]** Speech ▪ **[2]** Data **Notes:** ▪ If this parameter is not configured or is set to -1, Audio 3.1 capability is used. ▪ The Audio 7 option is currently not supported. ▪ The *x* in the ini file parameter name denotes the trunk number, where 0 is Trunk 1.
[TransferCapabilityForDataCalls]	Defines the ISDN Transfer Capability for data calls. ▪ **[0]** = (Default) ISDN Transfer Capability for data calls is 64k unrestricted (data). ▪ **[1]** = ISDN Transfer Capability for data calls is determined according to the ISDNTransferCapability parameter.
Web: ISDN Transfer On Connect EMS: Send ISDN Transfer On Connect CLI: isdn-trsfr-on-conn **[SendISDNTransferOnConnect]**	This parameter is used for the ECT/TBCT/RLT/Path Replacement ISDN transfer methods. Usually, the device requests the PBX to connect an incoming and outgoing call. This parameter determines if the outgoing call (from the device to the PBX) must be connected before the transfer is initiated. ▪ **[0]** Alert = (Default) Enables ISDN Transfer if the outgoing call is in Alerting or Connect state. ▪ **[1]** Connect = Enables ISDN Transfer only if the outgoing call is in Connect state. **Note:** For RLT ISDN transfer (TrunkTransferMode = 2 and ProtocolType = 14 DMS-100), this parameter must be set to 1.
[ISDNTransferCompleteTimeout]	Defines the timeout (in seconds) for determining ISDN call transfer (ECT, RLT, or TBCT) failure. If the device does not receive any response to an ISDN transfer attempt within this user-defined time, the device identifies this as an ISDN transfer failure and subsequently performs a hairpin TDM connection or sends a SIP NOTIFY message with a SIP 603 response (depending whether hairpin is enabled or disabled, using the parameter DisableFallbackTransferToTDM). The valid range is 1 to 10. The default is 4.
Web/EMS: Enable Network ISDN Transfer CLI: network-isdn-xfer **[EnableNetworkISDNTransfer]**	Determines whether the device allows interworking of network-side received ECT/TBCT Facility messages (NI-2 TBCT - Two B-channel Transfer and ETSI ECT - Explicit Call Transfer) to SIP REFER. ▪ **[0]** Disable = Rejects ISDN transfer requests. ▪ **[1]** Enable = (Default) The device sends a SIP REFER message to the remote call party if ECT/TBCT Facility messages are received from the ISDN side (e.g., from a PBX).
[DisableFallbackTransferToTDM]	Enables "hairpin" TDM transfer upon ISDN (ECT, RLT, or TBCT) call transfer failure. When this feature is enabled and an ISDN call transfer failure occurs, the device sends a SIP NOTIFY message with a SIP 603 Decline response. ▪ **[0]** = (Default) The device performs a hairpin TDM transfer upon ISDN call transfer. ▪ **[1]** = Hairpin TDM transfer is disabled.
Web: Enable QSIG Transfer	Determines whether the device interworks QSIG Facility messages

Parameter	Description
Update CLI: qsig-xfer-update **[EnableQSIGTransferUpdate]**	with CallTranferComplete or CallTransferUpdate invoke application protocol data units (APDU) to SIP UPDATE messages with P-Asserted-Identity and optional Privacy headers. This feature is supported for IP-to-Tel and Tel-to-IP calls. ▪ **[0]** Disable = (Default) Ignores QSIG Facility messages with CallTranferComplete or CallTransferUpdate invokes. ▪ **[1]** Enable For example, assume A and C are PBX call parties and B is the SIP IP phone: 1 A calls B; B answers the call. 2 A places B on hold and calls C; C answers the call. 3 A performs a call transfer (the transfer is done internally by the PBX); B and C are connected to one another. In the above example, the PBX updates B that it is now talking with C. The PBX updates this by sending a QSIG Facility message with CallTranferComplete invoke APDU. The device interworks this message to a SIP UPDATE message containing a P-Asserted-Identity header with the number and name derived from the QSIG CallTranferComplete RedirectionNumber and RedirectionName. **Notes:** ▪ For IP-to-Tel calls, the RedirectionNumber and RedirectionName in the CallTRansferComplete invoke is derived from the P-Asserted-Identity and Privacy headers in the received SIP INFO message. ▪ To include the P-Asserted-Identity header in outgoing SIP UPDATE messages, set the AssertedIDMode parameter to **Add P-Asserted-Identity.**

EMS: CAS Detection Of Hook Flash
CLI: is-cas-sndhook-flsh
[CASSendHookFlash]
	Enables sending Wink signal toward CAS trunks.
	▪ **[0]** = Disable (default)
	▪ **[1]** = Enable
	If the device receives a mid-call SIP INFO message with flashhook event body (as shown below) and this parameter is set to 1, the device generates a wink signal toward the CAS trunk. The CAS wink signal is done by changing the A bit from 1 to 0, and then back to 1 for 450 msec.
	`INFO sip:4505656002@192.168.13.40:5060 SIP/2.0`
	`Via: SIP/2.0/UDP 192.168.13.2:5060`
	`From: <sip:06@192.168.13.2:5060>`
	`To:`
	`<sip:4505656002@192.168.13.40:5060>;tag=13287879`
	`6-1040067870294`
	`Call-ID: 0010-0016-D69A7DA8-1@192.168.13.2`
	`CSeq:2 INFO`
	`Content-Type: application/broadsoft`
	`Content-Length: 17`
	`event flashhook`
	Note: This parameter is applicable only to T1 CAS protocols.

Release Cause Mapping from ISDN to SIP Table	
Web: Release Cause Mapping Table EMS: ISDN to SIP Cause Mapping	This table parameter maps ISDN Q.850 Release Causes to SIP responses. The format of the ini file table parameter is as follows: [CauseMapISDN2SIP] FORMAT CauseMapISDN2SIP_Index =

Parameter	Description
CLI: configure voip > gw manipulations CauseMapIsdn2Sip **[CauseMapISDN2SIP]**	CauseMapISDN2SIP_IsdnReleaseCause, CauseMapISDN2SIP_SipResponse; [\CauseMapISDN2SIP]
Release Cause Mapping from SIP to ISDN Table	
Web: Release Cause Mapping Table EMS: SIP to ISDN Cause Mapping CLI: configure voip > gw manipulations CauseMapSip2Isdn **[CauseMapSIP2ISDN]**	This table parameter maps SIP responses to Q.850 Release Causes. The format of the ini file table parameter is as follows: [CauseMapSIP2ISDN] FORMAT CauseMapSIP2ISDN_Index = CauseMapSIP2ISDN_SipResponse, CauseMapSIP2ISDN_IsdnReleaseCause; [\CauseMapSIP2ISDN]

67.10.8 Answer and Disconnect Supervision Parameters

The answer and disconnect supervision parameters are described in the table below.

Table 67-63: Answer and Disconnect Parameters

Parameter	Description
Web: Wait before PSTN Release-Ack CLI: wait-befor-pstn-rel-ack [TimeToWaitForPstnReleaseAck]	Defines a timeout (in milliseconds) that the device waits for the receipt of an ISDN Q.931 Release message from the PSTN side before releasing the channel. The Release ACK is typically sent by the PSTN in response to the device's Disconnect message to end the call. If the timeout expires and a Release message has not yet been received, the device releases the call channel. The valid value is 1 to 360,000. The default is 6,000. **Note:** This parameter is applicable only to Digital interfaces.
Web: Answer Supervision EMS: Enable Voice Detection CLI: answer-supervision **[EnableVoiceDetection]**	Enables the sending of SIP 200 OK upon detection of speech, fax, or modem. ▪ **[1]** Yes = The device sends a SIP 200 OK (in response to an INVITE message) when speech, fax, or modem is detected (from the Tel side, for analog interfaces). ▪ **[0]** No = (Default) The device sends a SIP 200 OK only after it completes dialing (to the Tel side, for analog interfaces). Typically, this feature is used only when early media (enabled using the EnableEarlyMedia parameter) is used to establish the voice path before the call is answered. **Notes:** ▪ FXO interfaces: This feature is applicable only to one-stage dialing (FXO). ▪ Digital interfaces: To activate this feature, set the EnableDSPIPMDetectors parameter to 1. ▪ Digital interfaces: This feature is applicable only when the protocol type is CAS.
Web/EMS: Max Call Duration (min)	Defines the maximum duration (in minutes) of a call. If this

Parameter	Description
CLI: mx-call-duration **[MaxCallDuration]**	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).
CLI: configure voip > sip advanced-settings > set mn-call-duration [MinCallDuration]	Defines the minimum call duration (in seconds) for the Tel side. If an established call is terminated by the IP side before this duration expires, the device terminates the call with the IP side, but delays the termination toward the Tel side until this timeout expires. The valid value range is 0 to 10 seconds, where 0 (default) disables this feature. For example: assume the minimum call duration is set to 10 seconds and an IP phone hangs up a call established with a BRI phone after 2 seconds. As the call duration is less than the minimum call duration, the device does not disconnect the call on the Tel side. However, it sends a SIP 200 OK immediately upon receipt of the BYE to disconnect from the IP phone. The call is disconnected from the Tel side only when the call duration is greater than or equal to the minimum call duration. **Notes:** ▪ This parameter is applicable to IP-to-Tel and Tel-to-IP calls. ▪ This parameter is applicable only to ISDN and CAS protocols.
Web/EMS: Disconnect on Dial Tone CLI: disc-on-dial-tone [DisconnectOnDialTone]	Determines whether the device disconnects a call when a dial tone is detected from the PBX. ▪ [0] Disable = (Default) Call is not released. ▪ [1] Enable = Call is released if a dial tone is detected on the device's FXO port. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ This option is in addition to the mechanism that disconnects a call when either busy or reorder tones are detected.
Web: Send Digit Pattern on Connect EMS: Connect Code CLI: digit-pttrn-on-conn **[TelConnectCode]**	Defines a digit pattern to send to the Tel side after a SIP 200 OK is received from the IP side. The digit pattern is a user-defined DTMF sequence that is used to indicate an answer signal (e.g., for billing). The valid range is 1 to 8 characters. **Note:** This parameter is applicable only to FXO/CAS.
Web: Disconnect on Broken Connection EMS: Disconnect Calls on Broken Connection CLI: disc-broken-conn **[DisconnectOnBrokenConnection]**	Global parameter that enables the device to release calls if RTP packets are not received within a user-defined timeout (configured by the BrokenConnectionEventTimeout parameter). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_DisconnectOnBrokenConnection). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP

Parameter	Description
	Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Broken Connection Timeout EMS: Broken Connection Event Timeout CLI: broken-connection-event-timeout **[BrokenConnectionEventTimeout]**	Defines the time period (in 100-msec units) after which a call is disconnected if an RTP packet is not received. The valid range is from 3 (i.e., 300 msec) to an unlimited value (e.g., 20 hours). The default is 100 (i.e., 10000 msec or 10 seconds). **Notes:** ▪ This parameter is applicable only if the parameter DisconnectOnBrokenConnection is set to 1. ▪ Currently, this feature functions only if Silence Suppression is disabled.
Web: Disconnect Call on Silence Detection EMS: Disconnect On Detection Of Silence CLI: disc-on-silence-det **[EnableSilenceDisconnect]**	Determines whether calls are disconnected after detection of silence. ▪ **[1]** Yes = The device disconnects calls in which silence occurs (in both call directions) for more than a user-defined time. ▪ **[0]** No = (Default) Call is not disconnected when silence is detected. The silence duration can be configured by the FarEndDisconnectSilencePeriod parameter (default 120). **Note:** To activate this feature, set the parameters EnableSilenceCompression and FarEndDisconnectSilenceMethod to 1.
Web: Silence Detection Period [sec] EMS: Silence Detection Time Out **[FarEndDisconnectSilencePeriod]**	Defines the duration of the silence period (in seconds) after which the call is disconnected. The range is 10 to 28,800 (i.e., 8 hours). The default is 120 seconds. **Note:** For this parameter to take effect, a device reset is required.
Web: Silence Detection Method **[FarEndDisconnectSilenceMethod]**	Determines the silence detection method. ▪ **[0]** None = Silence detection option is disabled. ▪ **[1]** Packets Count = According to packet count. **Note:** For this parameter to take effect, a device reset is required.
[FarEndDisconnectSilenceThreshold]	Defines the threshold of the packet count (in percentages) below which is considered silence by the device. The valid range is 1 to 100%. The default is 8%. **Notes:** ▪ This parameter is applicable only if silence is detected according to packet count (FarEndDisconnectSilenceMethod is set to 1). ▪ For this parameter to take effect, a device reset is required.
[BrokenConnectionDuringSilence]	Enables the generation of the BrokenConnection event during a silence period if the channel's NoOp feature is enabled (using the parameter NoOpEnable) and if the channel stops receiving NoOp RTP packets. ▪ **[0]** Disable (default)

Parameter	Description
	- **[1]** Enable
Web: Trunk Alarm Call Disconnect Timeout CLI: trk-alrm-call-disc-to [TrunkAlarmCallDisconnectTimeout]	Defines the duration (in seconds) to wait after an E1/T1 trunk "Red" alarm (LOS / LOF) is raised, before the device disconnects the SIP call. If this timeout expires and the alarm is still raised, the device sends a SIP BYE message to terminate the call. If the alarm is cleared before this timeout expires, the call is not terminated, but continues as normal. The range is 1 to 3600. The default is 0 (20 for E1 and 40 for T1).
Web: Disconnect Call on Busy Tone Detection (ISDN) EMS: Isdn Disconnect On Busy Tone CLI: disc-on-bsy-tone-i [ISDNDisconnectOnBusyTone]	Determines whether a call is disconnected upon detection of a busy tone (for ISDN). ▪ [0] Disable = (Default) Do not disconnect call upon detection of busy tone. ▪ [1] Enable = Disconnect call upon detection of busy tone. **Notes:** ▪ This parameter is applicable only to ISDN protocols. ▪ IP-to-ISDN calls are disconnected on detection of SIT tones only in call alert state. If the call is in connected state, the SIT does not disconnect the calls. Detection of busy or reorder tones disconnects the IP-to-ISDN calls also in call connected state. ▪ For IP-to-CAS calls, detection of busy, reorder, or SIT tones disconnect the calls in any call state.
Web: Disconnect Call on Busy Tone Detection EMS: Disconnect On Detection End Tones CLI: disc-on-bsy-tone-c **[DisconnectOnBusyTone]**	Determines whether a call is disconnected upon detection of a busy tone. ▪ **[0]** Disable = Call is not disconnected upon detection of a busy tone. ▪ **[1]** Enable = (Default) Call is released upon detection of busy or reorder (fast busy) tone. **Notes:** ▪ This parameter is applicable only to FXO and CAS. ▪ This parameter is applicable to the IP-to-IP application. ▪ This parameter can also be configured in a Tel Profile.
Polarity (Current) Reversal for Call Release (Analog Interfaces) Parameters	
[SetDefaultLinePolarityState] CLI: fxs-fxo > default-linepolarity-state	Defines the FXO line polarity, required for DID signaling. ▪ [0] = Positive line polarity ▪ [1] = Negative line polarity ▪ [2] = (Default) Auto - The device detects the polarity upon power-up or upon insertion of the RJ-11 cable, and uses it as a reference polarity. Typically, if the RJ-11 cabling is connected correctly (without crossing, Tip to Tip, Ring to Ring), the Tip line is positive compared to the Ring line. In this case, set this parameter to 0. With this configuration, the device assumes that the idle line polarity is Tip line positive. When the device receives a SIP INVITE, it checks the FXO line polarity. If the polarity is "Reversed", it skips this FXO line and goes to the next line.

Parameter	Description
	Notes: ▪ For this parameter to take effect, a device reset is required. ▪ To take advantage of this new feature, configure all FXO lines as a single Trunk Group with ascending or descending channel select mode, and configure routing rules to route incoming INVITE messages to this Trunk Group. ▪ This parameter is applicable only to FXO interfaces.
Web: Enable Polarity Reversal EMS: Enable Reversal Polarity CLI: polarity-rvrsl [EnableReversalPolarity]	Enables the polarity reversal feature for call release. ▪ [0] Disable = (Default) Disable the polarity reversal service. ▪ [1] Enable = Enable the polarity reversal service. If the polarity reversal service is enabled, the FXS interface changes the line polarity on call answer and then changes it back on call release. The FXO interface sends a 200 OK response when polarity reversal signal is detected (applicable only to one-stage dialing) and releases a call when a second polarity reversal signal is detected. **Note:** This parameter can also be configured in a Tel Profile.
Web/EMS: Enable Current Disconnect CLI: current-disc [EnableCurrentDisconnect]	Enables call release upon detection of a Current Disconnect signal. ▪ [0] Disable = (Default) Disable the current disconnect service. ▪ [1] Enable = Enable the current disconnect service. If the current disconnect service is enabled: ▪ The FXO releases a call when a current disconnect signal is detected on its port. ▪ The FXS interface generates a 'Current Disconnect Pulse' after a call is released from IP. The current disconnect duration is configured by the CurrentDisconnectDuration parameter. The current disconnect threshold (FXO only) is configured by the CurrentDisconnectDefaultThreshold parameter. The frequency at which the analog line voltage is sampled is configured by the TimeToSampleAnalogLineVoltage parameter. **Note:** This parameter can also be configured in a Tel Profile.
EMS: Polarity Reversal Type CLI: polarity-reversal-type [PolarityReversalType]	Defines the voltage change slope during polarity reversal or wink. ▪ [0] = (Default) Soft reverse polarity. ▪ [1] = Hard reverse polarity. **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ Some Caller ID signals use reversal polarity and/or Wink signals. In these cases, it is recommended to set the parameter PolarityReversalType to 1 (Hard).

Parameter	Description
	▪ For this parameter to take effect, a device reset is required.
EMS: Current Disconnect Duration CLI: current-disconnect-duration [CurrentDisconnectDuration]	Defines the duration (in msec) of the current disconnect pulse. The range is 200 to 1500. The default is 900. **Notes:** ▪ This parameter is applicable for FXS and FXO interfaces. ▪ The FXO interface detection window is 100 msec below the parameter's value and 350 msec above the parameter's value. For example, if this parameter is set to 400 msec, then the detection window is 300 to 750 msec. ▪ For this parameter to take effect, a device reset is required.
[CurrentDisconnectDefaultThreshold]	Defines the line voltage threshold at which a current disconnect detection is considered. The valid range is 0 to 20 Volts. The default is 4 Volts. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.
CLI: time-to-sample-analog-line-voltage [TimeToSampleAnalogLineVoltage]	Defines the frequency at which the analog line voltage is sampled (after offhook), for detection of the current disconnect threshold. The valid range is 100 to 2500 msec. The default is 1000 msec. **Notes:** ▪ This parameter is applicable only to FXO interfaces. ▪ For this parameter to take effect, a device reset is required.

67.10.9 Tone Parameters

This subsection describes the device's tone parameters.

67.10.9.1 Telephony Tone Parameters

The telephony tone parameters are described in the table below.

Table 67-64: Tone Parameters

Parameter	Description
CLI: help-tone-4-ip2ip [PlayHeldToneForIP2IP]	Enables playing a held tone to an IP-to-IP leg instead of placing it on hold. ▪ [0] = (Default) Disabled. The device interworks the re-INVITE with 'a=inactive' from one SIP leg to another SIP leg. ▪ [1] = Enabled. The device plays a held tone to the IP if it receives a re-INVITE with 'a=inactive' in the SDP from the party initiating the call hold. The held tone must be configured in the CPT or PRT file. **Note:** This parameter is applicable only to the IP-to-IP application.

Parameter	Description
Web: SIP Hold Behavior CLI: sip-hold-behavior [SIPHoldBehavior]	Enables the device to handle incoming re-INVITE messages with the "a=sendonly" attribute in the SDP, in the same way as if an "a=inactive" is received in the SDP. When enabled, the device plays a held tone to the Tel phone and responds with a SIP 200 OK containing the "a=recvonly" attribute in the SDP. ▪ [0] Disable (default) ▪ [1] Enable
Web/EMS: Dial Tone Duration [sec] CLI: dt-duration **[TimeForDialTone]**	Defines the duration (in seconds) that the dial tone is played (for digital interfaces, to an ISDN terminal). For digital interfaces: This parameter is applicable for overlap dialing when ISDNInCallsBehavior is set to 65536. The dial tone is played if the ISDN Setup message doesn't include the called number. The valid range is 0 to 60. The default is 5. For analog interfaces: FXS interfaces play the dial tone after the phone is picked up (off-hook). FXO interfaces play the dial tone after the port is seized in response to ringing (from PBX/PSTN). The valid range is 0 to 60. The default time is 16. Notes for analog interfaces: ▪ During play of dial tone, the device waits for DTMF digits. ▪ This parameter is not applicable when Automatic Dialing is enabled.
Web/EMS: Stutter Tone Duration CLI: sttr-tone-duration [StutterToneDuration]	Defines the duration (in msec) of the confirmation tone. A stutter tone is played (instead of a regular dial tone) when a Message Waiting Indication (MWI) is received. The stutter tone is composed of a confirmation tone (Tone Type #8), which is played for the defined duration (StutterToneDuration) followed by a stutter dial tone (Tone Type #15). Both these tones are defined in the CPT file. The range is 1,000 to 60,000. The default is 2,000 (i.e., 2 seconds). **Notes:** ▪ This parameter is applicable only to FXS interfaces. ▪ If you want to configure the duration of the confirmation tone to longer than 16 seconds, you must increase the value of the parameter TimeForDialTone accordingly. ▪ The MWI tone takes precedence over the call forwarding reminder tone. For more information on MWI, see Message Waiting Indication on page 436.
Web: FXO AutoDial Play BusyTone EMS: Auto Dial Play Busy Tone CLI: fxo-autodial-play-bsytn [FXOAutoDialPlayBusyTone]	Determines whether the device plays a busy / reorder tone to the PSTN side if a Tel-to-IP call is rejected by a SIP error response (4xx, 5xx or 6xx). If a SIP error response is received, the device seizes the line (off-hook), and then plays a busy / reorder tone to the PSTN side (for the duration defined by the parameter TimeForReorderTone). After playing the tone, the line is released (on-hook). ▪ [0] = Disable (default) ▪ [1] = Enable **Note:** This parameter is applicable only to FXO interfaces.
Web: Hotline Dial Tone Duration EMS: Hot Line Tone Duration CLI: hotline-dt-dur	Defines the duration (in seconds) of the hotline dial tone. If no digits are received during this duration, the device initiates a call to a user-defined number (configured in the Automatic Dialing table - TargetOfChannel - see Configuring Automatic Dialing on page 478). The valid range is 0 to 60. The default is 16.

Parameter	Description
[HotLineToneDuration]	**Notes:** - This parameter is applicable to FXS and FXO interfaces. - You can define the Hotline duration per FXS/FXO port using the Automatic Dialing table.
Web/EMS: Reorder Tone Duration [sec] CLI: reorder-tone-duration **[TimeForReorderTone]**	For analog interfaces: Defines the duration (in seconds) that the device plays a busy or reorder tone before releasing the line. Typically, after playing the busy or reorder tone for this duration, the device starts playing an offhook warning tone. For digital interfaces: Defines the duration (in seconds) that the CAS device plays a busy or reorder tone before releasing the line. The valid range is 0 to 254. The default is 0 seconds for analog interfaces and 10 seconds for digital interfaces. Note that the Web interface denotes the default value (for analog and digital interfaces) as a string value of "255". **Notes:** - The selected busy or reorder tone is according to the SIP release cause code received from IP. - This parameter is also applicable for ISDN when the PlayBusyTone2ISDN parameter is set to 2. - This parameter can also be configured for a Tel Profile (in the Tel Profile table).
Web: Time Before Reorder Tone [sec] EMS: Time For Reorder Tone CLI: time-b4-reordr-tn [TimeBeforeReorderTone]	Defines the delay interval (in seconds) from when the device receives a SIP BYE message (i.e., remote party terminates call) until the device starts playing a reorder tone to the FXS phone. The valid range is 0 to 60. The default is 0. **Note:** This parameter is applicable only to FXS interfaces.
Web: Cut Through Reorder Tone Duration [sec] CLI: cut-thru-reord-dur **[CutThroughTimeForReOrderTone]**	Defines the duration (in seconds) of the reorder tone played to the Tel side after the IP call party releases the call, for the Cut-Through feature. After the tone stops playing, an incoming call is immediately answered if the FXS is off-hooked (for analog interfaces) or the PSTN is connected (for digital interfaces). The valid values are 0 to 30. The default is 0 (i.e., no reorder tone is played). **Note:** To enable the Cut-Through feature, use the DigitalCutThrough (for CAS channels) or CutThrough (for FXS channels) parameters.
Web/EMS: Enable Comfort Tone CLI: comfort-tone [EnableComfortTone]	Determines whether the device plays a comfort tone (Tone Type #18) to the FXS/FXO endpoint after a SIP INVITE is sent and before a SIP 18x response is received. - [0] Disable (default) - [1] Enable **Note:** This parameter is applicable to FXS and FXO interfaces.
[WarningToneDuration]	Defines the duration (in seconds) for which the offhook warning tone is played to the user. The valid range is -1 to 2,147,483,647. The default is 600. **Notes:** - A negative value indicates that the tone is played infinitely. - This parameter is applicable only to analog interfaces.
Web: Play Busy Tone to Tel CLI: play-bsy-tone-2tel	Enables the device to play a busy or reorder tone to the PSTN after a Tel-to-IP call is released.

Parameter	Description
[PlayBusyTone2ISDN]	▪ [0] Don't Play = (Default) Immediately sends an ISDN Disconnect message. ▪ [1] Play when Disconnecting = Sends an ISDN Disconnect message with PI = 8 and plays a busy or reorder tone to the PSTN (depending on the release cause). ▪ [2] Play before Disconnect = Delays the sending of an ISDN Disconnect message for a user-defined time (configured by the TimeForReorderTone parameter) and plays a busy or reorder tone to the PSTN. This is applicable only if the call is released from the IP [Busy Here (486) or Not Found (404)] before it reaches the Connect state; otherwise, the Disconnect message is sent immediately and no tones are played. **Note:** This parameter is applicable only to digital PSTN interfaces.
CLI: configure voip > gw digitalgw digital-gw-parameters > q850-reason-code-2play-user-tone **[Q850ReasonCode2PlayUserTone]**	Defines an ISDN Q.8931 release cause code(s), which if mapped to the SIP release reason received from the IP side, causes the device to play a user-defined tone from the installed PRT file to the Tel side. For example, if the the received SIP release cause is 480 Temporarily Unavailable and you configure this parameter with Q.931 release code 18 (No User Responding), the device plays the user-defined tone to the Tel side. The user-defined tone is configured when creating the PRT file, using AudioCodes DConvert utility. The tone must be assigned to the "acSpecialConditionTone" (Tone Type 21) option in DConvert. The parameter can be configured with up to 10 release codes. When configuring multiple codes, separate the codes by commas (without spaces). For example: `Q850ReasonCode2PlayUserTone = 1,18,24` If the SIP release reason received from the IP side is mapped to the Q.931 release code specified by the parameter, the device plays the user-defined tone. Otherwise, if not specified and the release code is 17 (User Busy), the device plays the busy tone and for all other release codes, the device plays the reorder tone. **Notes:** ▪ The parameter is applicable only to digital PSTN interfaces. ▪ To enable the feature, the 'Play Busy Tone to Tel' (PlayBusyTone2ISDN) parameter must be enabled (set to 1 or 2).
Web: Play Ringback Tone to Tel EMS: Play Ring Back Tone To Tel CLI: play-rbt2tel **[PlayRBTone2Tel]**	Determines the playing method of the ringback tone to the Tel (for analog interfaces) or Trunk (for digital interfaces) side. For digital interfaces: This parameter applies to all trunks that are not configured by the PlayRBTone2Trunk parameter (which defines ringback tone per Trunk). ▪ **[0]** Don't Play = ✓ Analog Interfaces: Ringback tone is not played. ✓ Digital Interfaces: The device configured for ISDN / CAS doesn't play a ringback tone. No PI is sent to the ISDN unless the ProgressIndicator2ISDN_x parameter is configured differently. ▪ **[1]** Play on Local = ✓ Analog Interfaces: Plays a ringback tone to the Tel side of the call when a SIP 180/183 response is received. ✓ Digital Interfaces: The device configured for CAS plays a local ringback tone to the PSTN upon receipt of a SIP 180 Ringing

Parameter	Description
	response (with or without SDP). Note that the receipt of a 183 response does not cause the device configured for CAS to play a ringback tone (unless the SIP183Behaviour parameter is set to 1). The device configured for ISDN operates according to the LocalISDNRBSource parameter: 1) If the device receives a 180 Ringing response (with or without SDP) and the LocalISDNRBSource parameter is set to 1, it plays a ringback tone and sends an ISDN Alert with PI = 8 (unless the ProgressIndicator2ISDN_x parameter is configured differently). 2) If the LocalISDNRBSource parameter is set to 0, the device doesn't play a ringback tone and an Alert message without PI is sent to the ISDN. In this case, the PBX / PSTN plays the ringback tone to the originating terminal. Note that the receipt of a 183 response does not cause the device configured for ISDN to play a ringback tone; the device issues a Progress message (unless SIP183Behaviour is set to 1). If the SIP183Behaviour parameter is set to 1, the 183 response is handled the same way as a 180 Ringing response. ▪ **[2]** Prefer IP = (Default): ✓ Analog Interfaces: Plays a ringback tone to the Tel side only if a 180/183 response without SDP is received. If 180/183 with SDP message is received, the device cuts through the voice channel and doesn't play the ringback tone. ✓ Digital Interfaces: Plays according to 'Early Media'. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device configured for ISDN / CAS doesn't play the ringback tone; PI = 8 is sent in an ISDN Alert message (unless the ProgressIndicator2ISDN_x parameter is configured differently). If a 180 response is received, but the 'early media' voice channel is not opened, the device configured for CAS plays a ringback tone to the PSTN. The device configured for ISDN operates according to the LocalISDNRBSource parameter: 1) If LocalISDNRBSource is set to 1, the device plays a ringback tone and sends an ISDN Alert with PI = 8 to the ISDN (unless the ProgressIndicator2ISDN_x parameter is configured differently). 2) If LocalISDNRBSource is set to 0, the device doesn't play a ringback tone. No PI is sent in the ISDN Alert message (unless the ProgressIndicator2ISDN_x parameter is configured differently). In this case, the PBX / PSTN plays a ringback tone to the originating terminal. Note that the receipt of a 183 response results in an ISDN Progress message (unless SIP183Behaviour is set to 1). If SIP183Behaviour is set to 1 (183 is handled the same way as a 180 + SDP), the device sends an Alert message with PI = 8, without playing a ringback tone. ▪ **[3]** Play Local Until Remote Media Arrive = Plays a ringback tone according to received media. The behaviour is similar to [2]. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device plays a local ringback tone if there are no prior received RTP packets. The device stops playing the local ringback tone as soon as it starts receiving RTP packets. At this stage, if the device receives additional 18x

Parameter	Description
	responses, it does not resume playing the local ringback tone. Note that for ISDN trunks, this option is applicable only if the LocalISDNRBSource parameter is set to 1. **Note:** This parameter is applicable to the Gateway and IP-to-IP applications.
Web: Play Ringback Tone to Trunk CLI: play-rbt-to-trk [PlayRBTone2Trunk_x]	Determines the playing method of the ringback tone to the trunk side, per trunk. ▪ [-1] Not configured = (Default) The settings of the PlayRBTone2Tel parameter is used. ▪ [0] Don't Play = When the device is configured for ISDN / CAS, it doesn't play a ringback tone. No Progress Indicator (PI) is sent to the ISDN unless the ProgressIndicator2ISDN_x parameter is configured differently. ▪ [1] Play on Local = When the device is configured for CAS, it plays a local ringback tone to the PSTN upon receipt of a SIP 180 Ringing response (with or without SDP). Note that the receipt of a SIP 183 response does not cause the device configured for CAS to play a ringback tone (unless the SIP183Behaviour parameter is set to 1). When the device is configured for ISDN, it operates according to the LocalISDNRBSource parameter, as follows: ✓ If the device receives a SIP 180 Ringing response (with or without SDP) and the LocalISDNRBSource parameter is set to 1, it plays a ringback tone and sends an ISDN Alert with PI = 8 (unless the ProgressIndicator2ISDN_x parameter is configured differently). ✓ If the LocalISDNRBSource parameter is set to 0, the device doesn't play a ringback tone and an Alert message without PI is sent to the ISDN. In this case, the PBX / PSTN plays the ringback tone to the originating terminal. Note that the receipt of a 183 response does not cause the device to play a ringback tone; the device sends a Progress message (unless SIP183Behaviour is set to 1). If the SIP183Behaviour parameter is set to 1, the 183 response is handled the same way as a 180 Ringing response. ▪ [2] Prefer IP = Plays according to 'Early Media'. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device configured for ISDN / CAS doesn't play the ringback tone; PI = 8 is sent in an ISDN Alert message (unless the ProgressIndicator2ISDN_x parameter is configured differently). If a 180 response is received, but the 'early media' voice channel is not opened, the device configured for CAS plays a ringback tone to the PSTN. The device configured for ISDN operates according to the LocalISDNRBSource parameter: ✓ If LocalISDNRBSource is set to 1, the device plays a ringback tone and sends an ISDN Alert with PI = 8 to the ISDN (unless the ProgressIndicator2ISDN_x parameter is configured differently). ✓ If LocalISDNRBSource is set to 0, the device doesn't play a ringback tone. No PI is sent in the ISDN Alert message (unless the ProgressIndicator2ISDN_x parameter is configured differently). In this case, the PBX / PSTN plays a ringback tone

Parameter	Description
	to the originating terminal. Note that the receipt of a 183 response results in an ISDN Progress message (unless SIP183Behaviour is set to 1). If SIP183Behaviour is set to 1 (183 is handled the same way as a 180 with SDP), the device sends an Alert message with PI = 8 without playing a ringback tone. ▪ [3] Play Local Until Remote Media Arrive = Plays tone according to received media. The behaviour is similar to option [2]. If a SIP 180 response is received and the voice channel is already open (due to a previous 183 early media response or due to an SDP in the current 180 response), the device plays a local ringback tone if there are no prior received RTP packets. The device stops playing the local ringback tone as soon as it starts receiving RTP packets. At this stage, if the device receives additional 18x responses, it does not resume playing the local ringback tone. Note that for ISDN trunks, this option is applicable only if LocalISDNRBSource is set to 1. **Notes:** ▪ This parameter is applicable only to the Gateway (GW) application (i.e., not the IP-to-IP application). ▪ The x in the ini file parameter name denotes the trunk number, where 0 is Trunk 1. ▪ This parameter is applicable only to digital PSTN interfaces.
Web: Play Ringback Tone to IP EMS: Play Ring Back Tone To IP CLI: play-rbt-2ip **[PlayRBTone2IP]**	Global parameter that enables the device to play a ringback tone to the IP side for IP-to-Tel calls. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_PlayRBTone2IP). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Play Local RBT on ISDN Transfer EMS: Play RBT On ISDN Transfer CLI: play-l-rbt-isdn-trsfr [PlayRBTOnISDNTransfer]	Determines whether the device plays a local ringback tone for ISDN's Two B Channel Transfer (TBCT), Release Line Trunk (RLT), or Explicit Call Transfer (ECT) call transfers to the originator when the second leg receives an ISDN Alerting or Progress message. ▪ [0] Don't Play (default) ▪ [1] Play **Notes:** ▪ For Blind transfer, the local ringback tone is played to first call PSTN party when the second leg receives the ISDN Alerting or Progress message. ▪ For Consulted transfer, the local ringback tone is played when the second leg receives ISDN Alerting or Progress message if the Progress message is received after a SIP REFER. ▪ This parameter is applicable only if the parameter SendISDNTransferOnConnect is set to 1.
Web: MFC R2 Category EMS: R2 Category CLI: mfcr2-category [R2Category]	Defines the tone for MFC R2 calling party category (CPC). The parameter provides information on the calling party such as National or International call, Operator or Subscriber and Subscriber priority. The value range is 1 to 15 (defining one of the MFC R2 tones). The default is 1.

Parameter	Description
Tone Index Table	
Web: Tone Index Table EMS: Analog Gateway Provisioning > Tone Index CLI: configure voip > gw analoggw tone-index [ToneIndex]	This table parameter configures the Tone Index table, which allows you to define distinctive ringing and call waiting tones per FXS endpoint (or for a range of FXS endpoints). The format of the ini file table parameter is as follows: [ToneIndex] FORMAT ToneIndex_Index = ToneIndex_FXSPort_First, ToneIndex_FXSPort_Last, ToneIndex_SourcePrefix, ToneIndex_DestinationPrefix, ToneIndex_PriorityIndex; [\ToneIndex] For example, the configuration below plays the tone Index #3 to FXS ports 1 and 2 if the source number prefix of the received call is 20. ToneIndex 1 = 1, 2, 20*, , 3; For a detailed description of this table, see Configuring FXS Distinctive Ringing and Call Waiting Tones per Source/Destination Number. **Note:** This parameter is applicable only to FXS interfaces.

67.10.9.2 Tone Detection Parameters

The signal tone detection parameters are described in the table below.

Table 67-65: Tone Detection Parameters

Parameter	Description
EMS: DTMF Enable CLI: DTMF-detector-enable **[DTMFDetectorEnable]**	Enables the detection of DTMF signaling. ▪ **[0]** = Disable ▪ **[1]** = Enable (default)
EMS: MF R1 Enable CLI: MFR1-detector-enable **[MFR1DetectorEnable]**	Enables the detection of MF-R1 signaling. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: R1.5 Detection Standard [R1DetectionStandard]	Determines the MF-R1 protocol used for detection. ▪ [0] = ITU (default) ▪ [1] = R1.5 **Note:** For this parameter to take effect, a device reset is required.
EMS: User Defined Tone Enable CLI: user-defined-tones-detector-enable **[UserDefinedToneDetectorEnable]**	Enables the detection of User Defined Tones signaling, applicable for Special Information Tone (SIT) detection. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
EMS: SIT Enable CLI: SIT-detector-enable **[SITDetectorEnable]**	Enables SIT detection according to the ITU-T recommendation E.180/Q.35. ▪ **[0]** = Disable (default) ▪ **[1]** = Enable To disconnect IP-to-ISDN calls when a SIT tone is detected, the following parameters must be configured: ▪ SITDetectorEnable = 1 ▪ UserDefinedToneDetectorEnable = 1

Parameter	Description
	- ISDNDisconnectOnBusyTone = 1 (applicable for Busy, Reorder and SIT tones) Another parameter for handling the SIT tone is SITQ850Cause, which determines the Q.850 cause value specified in the SIP Reason header that is included in a 4xx response when a SIT tone is detected on an IP-to-Tel call. To disconnect IP-to-CAS calls when a SIT tone is detected, the following parameters must be configured (applicable to FXO interfaces): - SITDetectorEnable = 1 - UserDefinedToneDetectorEnable = 1 - DisconnectOnBusyTone = 1 (applicable for busy, reorder, and SIT tones) **Notes:** - For this parameter to take effect, a device reset is required. - The IP-to-ISDN call is disconnected on detection of a SIT tone only in call alert state. If the call is in connected state, the SIT does not disconnect the call. Detection of busy or reorder tones disconnect these calls also in call connected state. - For IP-to-CAS calls, detection of busy, reorder, or SIT tones disconnect the call in any call state.
EMS: UDT Detector Frequency Deviation CLI: UDT-detector-frequency-deviation **[UDTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each signal frequency. The valid range is 1 to 50. The default is 50. **Note:** For this parameter to take effect, a device reset is required.
EMS: CPT Detector Frequency Deviation CLI: CPT-detector-frequency-deviation **[CPTDetectorFrequencyDeviation]**	Defines the deviation (in Hz) allowed for the detection of each CPT signal frequency. The valid range is 1 to 30. The default is 10. **Note:** For this parameter to take effect, a device reset is required.

67.10.9.3 Metering Tone Parameters

The metering tone parameters are described in the table below.

Table 67-66: Metering Tone Parameters

Parameter	Description
Web: Generate Metering Tones EMS: Metering Mode CLI: gen-mtr-tones **[PayPhoneMeteringMode]**	Defines the method for configuring metering tones that are generated to the Tel side. - **[0]** Disable = (Default) Metering tones are not generated. - **[1]** Internal Table = Metering tones are generated according to user-defined pulses and intervals in the Charge Code table (see Configuring Charge Codes on page 470). - **[2]** SIP Interval Provided = (Proprietary method of TELES Communications Corporation) Advice-of-Charge service toward the PSTN. Periodic generation of AOC-D and AOC-E toward the PSTN. Calculation is based on seconds. The time interval is calculated according to the scale and tariff provided in the proprietary formatted

Parameter	Description
	file included in SIP INFO messages, which is always sent before 200 0K. The device ignores tariffs sent after the call is established. - **[3]** SIP RAW Data Provided = (Proprietary method of Cirpack) Advice-of-Charge service toward the PSTN. The received AOC-D messages contain a subtotal. When receiving AOC-D in raw format, provided in the header of SIP INFO messages, the device parses AOC-D raw data to obtain the number of units. This number is sent in the Facility message with AOC-D. In addition, the device stores the latest number of units in order to send them in AOC-E IE when the call is disconnected). - **[4]** SIP RAW Data Incremental Provided = (Proprietary method of Cirpack) Advice-of-Charge service toward the PSTN. The AOC-D message in the payload is an increment. When receiving AOC-D in raw format, provided in the header of SIP INFO messages, the device parses AOC-D raw data to obtain the number of units. This number is sent in the Facility message with AOC-D. The device generates the AOC-E. Parsing every AOC-D received and summing the values is required to obtain the total sum (that is placed in the AOC-E). **Note:** This parameter is applicable only to FXS interfaces and ISDN Euro trunks for sending AOC Facility messages (see Advice of Charge Services for Euro ISDN on page 468).
Web: Analog Metering Type EMS: Metering Type CLI: metering-type [MeteringType]	Determines the metering method for generating pulses (sinusoidal metering burst frequency) by the FXS port. - [0] 12 KHz = (Default) 12 kHz sinusoidal bursts. - [1] 16 KHz = 16 kHz sinusoidal bursts. - [2] = Polarity Reversal pulses. **Notes:** - For this parameter to take effect, a device reset is required. - This parameter is applicable only to FXS interfaces.
Web: Analog TTX Voltage Level EMS: TTX Voltage Level [AnalogTTXVoltageLevel]	Determines the metering signal/pulse voltage level (TTX). - [0] 0V = 0 Vrms sinusoidal bursts. - [1] 0.5V = (Default) 0.5 Vrms sinusoidal bursts. - [2] 1V = 1 Vrms sinusoidal bursts **Notes:** - For this parameter to take effect, a device reset is required. - This parameter is applicable only to FXS interfaces.
Charge Codes Table	
Web: Charge Codes Table EMS: Charge Codes CLI: configure voip > gw analoggw ChargeCode **[ChargeCode]**	This table parameter configures metering tones and their time intervals that the FXS interface generates to the Tel side or the E1 trunk (Euro ISDN) sends in AOC Facility messages to the PSTN (i.e., PBX). The format of the ini file table parameter is as follows: [ChargeCode] FORMAT ChargeCode_Index = ChargeCode_EndTime1, ChargeCode_PulseInterval1, ChargeCode_PulsesOnAnswer1, ChargeCode_EndTime2, ChargeCode_PulseInterval2, ChargeCode_PulsesOnAnswer2, ChargeCode_EndTime3, ChargeCode_PulseInterval3, ChargeCode_PulsesOnAnswer3, ChargeCode_EndTime4, ChargeCode_PulseInterval4, ChargeCode_PulsesOnAnswer4;

Parameter	Description
	[\ChargeCode] Where, ▪ EndTime = Period (1 - 4) end time. ▪ PulseInterval = Period (1 - 4) pulse interval. ▪ PulsesOnAnswer = Period (1 - 4) pulses on answer. For example: ChargeCode 1 = 7,30,1,14,20,2,20,15,1,0,60,1; ChargeCode 2 = 5,60,1,14,20,1,0,60,1; ChargeCode 3 = 0,60,1; ChargeCode 0 = 6, 3, 1, 12, 2, 1, 18, 5, 2, 0, 2, 1; **Notes:** ▪ To associate a configured Charge Code to an outgoing Tel-to-IP call, use the Outbound IP Routing table. ▪ To configure the Charge Codes table using the Web interface, see Configuring Charge Codes Table on page 470.

67.10.10 Telephone Keypad Sequence Parameters

The telephony keypad sequence parameters are described in the table below.

Table 67-67: Telephone Keypad Sequence Parameters

Parameter	Description
Web/EMS: Call Pickup Key CLI: sip-definition advanced-settings > call-pickup-key **[KeyCallPickup]**	Defines the keying sequence for performing a call pick-up. Call pick-up allows the FXS endpoint to answer another telephone's incoming call by pressing this user-defined sequence of digits. When the user dials these digits (e.g., #77), the incoming call from another phone is forwarded to the user's phone. The valid value is a string of up to 15 characters (0-9, #, and *). The default is undefined. **Notes:** ▪ Call pick-up is configured only for FXS endpoints pertaining to the same Trunk Group. ▪ This parameter is applicable only to FXS interfaces.
Prefix for External Line	
[Prefix2ExtLine]	Defines a string prefix (e.g., '9' dialed for an external line) that when dialed, the device plays a secondary dial tone (i.e., stutter tone) to the FXS line and then starts collecting the subsequently dialed digits from the FXS line. The valid range is a one-character string. By default, no value is defined. **Notes:** ▪ You can enable the device to add this string as the prefix to the collected (and sent) digits, using the parameter AddPrefix2ExtLine. ▪ This parameter is applicable only to FXS interfaces.
CLI: prefix-2-ext-line **[AddPrefix2ExtLine]**	Determines whether the prefix string for accessing an external line (defined by the parameter Prefix2ExtLine) is added to the dialed number as the prefix and together sent to the IP destination (Tel-to-IP calls).

Parameter	Description
	▪ **[0]** = Disable (default) ▪ **[1]** = Enable For example, if this parameter is enabled and the prefix string for the external line is defined as "9" (using the parameter Prefix2ExtLine) and the FXS user wants to make a call to destination "123", the device collects and sends all the dialed digits, including the prefix string, as "9123" to the IP destination number. **Note:** This parameter is applicable only to FXS interfaces.
Hook Flash Parameters	
Web: Flash Keys Sequence Style CLI: flash-key-seq-style **[FlashKeysSequenceStyle]**	Determines the hook-flash key sequence for FXS interfaces. ▪ **[0]** Flash hook = (Default) Only the phone's flash button is used, according to the following scenarios: ✓ During an existing call, if the user presses the flash button, the call is put on hold; a dial tone is heard and the user is able to initiate a second call. Once the second call is established, on-hooking transfers the first (held) call to the second call. ✓ During an existing call, if a call comes in (call waiting), pressing the flash button places the active call on hold and answers the waiting call; pressing flash again toggles between these two calls. ▪ **[1]** Sequence 1 = Sequence of flash button with digit: ✓ Flash + 1: holds a call or toggles between two existing calls ✓ Flash + 2: makes a call transfer. ✓ Flash + 3: makes a three-way conference call (if the Three-Way Conference feature is enabled, i.e., the parameter Enable3WayConference is set to 1 and the parameter 3WayConferenceMode is set to 2). ▪ **[2]** Sequence 2 = Sequence of flash button with digit: ✓ Flash only: Places a call on hold. ✓ Flash + 1: 1) When the device handles two calls (an active and a held call) and this key sequence is dialed, it sends a SIP BYE message to the active call and the previously held call becomes the active call. 2) When there is an active call and an incoming waiting call, if this key sequence is dialed, the device disconnects the active call and the waiting call becomes an active call. ✓ Flash + 2: Places a call on hold and answers a call-waiting call, or toggles between active and on-hold calls. ✓ Flash + 3: Makes a three-way conference call. This is applicable only if the Enable3WayConference parameter is set to 1 and the 3WayConferenceMode parameter is set to 2. Note that the settings of the ConferenceCode parameter is ignored. ✓ Flash + 4: Makes a call transfer. **Note:** This parameter is applicable only to FXS interfaces.
Web: Flash Keys Sequence Timeout	Defines the Flash keys sequence timeout - the time (in msec) that the device waits for digits after the user presses the flash

Parameter	Description
CLI: flash-key-seq-tmout **[FlashKeysSequenceTimeout]**	button (Flash Hook + Digit mode - when the parameter FlashKeysSequenceStyle is set to 1 or 2). The valid range is 100 to 5,000. The default is 2,000.
Keypad Feature - Call Forward Parameters	
Web: Forward Unconditional EMS: Call Forward Unconditional CLI: fwd-unconditional **[KeyCFUnCond]**	Defines the keypad sequence to activate the immediate call forward option.
Web: Forward No Answer EMS: Call Forward No Answer CLI: fwd-no-answer **[KeyCFNoAnswer]**	Defines the keypad sequence to activate the forward on no answer option.
Web: Forward On Busy EMS: Call Forward Busy CLI: fwd-on-busy **[KeyCFBusy]**	Defines the keypad sequence to activate the forward on busy option.
Web: Forward On Busy or No Answer EMS: CF Busy Or No Answer CLI: fwd-busy-or-no-ans **[KeyCFBusyOrNoAnswer]**	Defines the keypad sequence to activate the forward on 'busy or no answer' option.
Web: Do Not Disturb EMS: CF Do Not Disturb CLI: fwd-dnd **[KeyCFDoNotDisturb]**	Defines the keypad sequence to activate the Do Not Disturb option (immediately reject incoming calls).
To activate the required forward method from the telephone: 1 Dial the user-defined sequence number on the keypad; a dial tone is heard. 2 Dial the telephone number to which the call is forwarded (terminate the number with #); a confirmation tone is heard.	
Web: Forward Deactivate EMS: Call Forward Deactivation CLI: fwd-deactivate **[KeyCFDeact]**	Defines the keypad sequence to deactivate any of the call forward options. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Caller ID Restriction Parameters	
Web: Restricted Caller ID Activate EMS: CLIR CLI: id-restriction-act **[KeyCLIR]**	Defines the keypad sequence to activate the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Web: Restricted Caller ID Deactivate EMS: CLIR Deactivation CLI: id-restriction-deact **[KeyCLIRDeact]**	Defines the keypad sequence to deactivate the restricted Caller ID option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Hotline Parameters	
Web: Hot-line Activate EMS: Hot Line CLI: hotline-act **[KeyHotLine]**	Defines the keypad sequence to activate the delayed hotline option. To activate the delayed hotline option from the telephone, perform the following: 1 Dial the user-defined sequence number on the keypad; a

Parameter	Description
	dial tone is heard. 2 Dial the telephone number to which the phone automatically dials after a configurable delay (terminate the number with #); a confirmation tone is heard.
Web: Hot-line Deactivate EMS: Hot Line Deactivation CLI: hotline-deact **[KeyHotLineDeact]**	Defines the keypad sequence to deactivate the delayed hotline option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Transfer Parameters **Note:** See the description of the KeyBlindTransfer parameter for this feature.	
Keypad Feature - Call Waiting Parameters	
Web: Call Waiting Activate EMS: Keypad Features CW CLI: cw-act **[KeyCallWaiting]**	Defines the keypad sequence to activate the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Web: Call Waiting Deactivate EMS: Keypad Features CW Deact CLI: cw-deact **[KeyCallWaitingDeact]**	Defines the keypad sequence to deactivate the Call Waiting option. After the sequence is pressed, a confirmation tone is heard.
Keypad Feature - Reject Anonymous Call Parameters	
Web: Reject Anonymous Call Activate EMS: Reject Anonymous Call **[KeyRejectAnonymousCall]**	Defines the keypad sequence to activate the reject anonymous call option, whereby the device rejects incoming anonymous calls. After the sequence is pressed, a confirmation tone is heard.
Web: Reject Anonymous Call Deactivate EMS: Reject Anonymous Call Deact **[KeyRejectAnonymousCallDeact]**	Defines the keypad sequence that de-activates the reject anonymous call option. After the sequence is pressed, a confirmation tone is heard.

67.10.11 FXO and FXS Parameters

The general FXO and FXS parameters are described in the table below.

Table 67-68: General FXO and FXS Parameters

Parameter	Description
Web: Update Port Info **[AnalogPortInfo_x]**	Defines an arbitrary name for an analog (FXS or FXO) port. This can be used to easily identify the port. The valid value is a string of up to 40 characters. By default, the value is undefined. **Notes:** ▪ For the ini file parameter, the *x* denotes the port number. ▪ For defining a port name in the Web interface, see "Assigning a Port Name" on page 63.
FXS Parameters	
Web: FXS Coefficient Type	Determines the FXS line characteristics (AC and DC) according to

Parameter	Description
EMS: Country Coefficients CLI: fxs-country-coefficients **[FXSCountryCoefficients]**	USA or Europe (TBR21) standards. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = (Default) United States **Note:** For this parameter to take effect, a device reset is required.
FXO Parameters	
Web: FXO Coefficient Type EMS: Country Coefficients CLI: fxo-country-coefficients **[CountryCoefficients]**	Determines the FXO line characteristics (AC and DC) according to USA or TBR21 standard. ▪ **[66]** Europe = TBR21 ▪ **[70]** USA = (Default) United States **Note:** For this parameter to take effect, a device reset is required.
CLI: fxo-dc-termination **[FXODCTermination]**	Defines the FXO line DC termination (i.e., resistance). ▪ **[0]** = (Default) DC termination is set to 50 Ohms. ▪ **[1]** = DC termination set to 800 Ohms. The termination changes from 50 to 800 Ohms only when moving from onhook to offhook. **Note:** For this parameter to take effect, a device reset is required.
CLI: enable-fxo-current-limit **[EnableFXOCurrentLimit]**	Enables limiting the FXO loop current to a maximum of 60 mA (according to the TBR21 standard). ▪ **[0]** = (Default) FXO line current limit is disabled. ▪ **[1]** = FXO loop current is limited to a maximum of 60 mA. **Note:** For this parameter to take effect, a device reset is required.
[FXONumberOfRings]	Defines the number of rings before the device's FXO interface answers a call by seizing the line. The valid range is 0 to 10. The default is 0. When set to 0, the FXO seizes the line after one ring. When set to 1, the FXO seizes the line after two rings. **Notes:** ▪ This parameter is applicable only if automatic dialing is not used. ▪ If caller ID is enabled and if the number of rings defined by the parameter RingsBeforeCallerID is greater than the number of rings defined by this parameter, the greater value is used.
Web/EMS: Dialing Mode CLI: dialing-mode **[IsTwoStageDial]**	Determines the dialing mode for IP-to-Tel (FXO) calls. ▪ **[0]** One Stage = One-stage dialing. In this mode, the device seizes one of the available lines (according to the ChannelSelectMode parameter), and then dials the destination phone number received in the INVITE message. To specify whether the dialing must start after detection of the dial tone or immediately after seizing the line, use the IsWaitForDialTone parameter. ▪ **[1]** Two Stages = (Default) Two-stage dialing. In this mode, the device seizes one of the PSTN/PBX lines without performing any dialing, connects the remote IP user to the PSTN/PBX, and all further signaling (dialing and Call Progress Tones) is performed directly with the PBX without the device's intervention. **Note:** This parameter can be configured for a Tel Profile.
Web/EMS: Waiting For Dial Tone CLI: waiting-4-dial-tone	Determines whether or not the device waits for a dial tone before dialing the phone number for IP-to-Tel (FXO) calls.

Parameter	Description
[IsWaitForDialTone]	▪ **[0]** No ▪ **[1]** Yes (default) When one-stage dialing and this parameter are enabled, the device dials the phone number (to the PSTN/PBX line) only after it detects a dial tone. If this parameter is disabled, the device immediately dials the phone number after seizing the PSTN/PBX line without 'listening' for a dial tone. **Notes:** ▪ The correct dial tone parameters must be configured in the CPT file. ▪ The device may take 1 to 3 seconds to detect a dial tone (according to the dial tone configuration in the CPT file). If the dial tone is not detected within 6 seconds, the device releases the call and sends a SIP 500 "Server Internal Error" response.
Web: Time to Wait before Dialing [msec] EMS: Time Before Dial CLI: time-wait-b4-dialing **[WaitForDialTime]**	For digital interfaces: Defines the delay after hook-flash is generated and until dialing begins. Applies to call transfer (i.e., the parameter TrunkTransferMode is set to 3) on CAS protocols. For analog interfaces: Defines the delay before the device starts dialing on the FXO line in the following scenarios: ▪ The delay between the time the line is seized and dialing begins during the establishment of an IP-to-Tel call. **Note:** Applicable only for one-stage dialing when the parameter IsWaitForDialTone is disabled. ▪ The delay between detection of a Wink and the start of dialing during the establishment of an IP-to-Tel call (for DID lines, see the EnableDIDWink parameter). ▪ For call transfer - the delay after hook-flash is generated and dialing begins. The valid range (in milliseconds) is 0 to 20,000 (i.e., 20 seconds). The default is 1,000 (i.e., 1 second).
Web: Ring Detection Timeout [sec] EMS: Timeout Between Rings CLI: ring-detection-tout **[FXOBetweenRingTime]**	Defines the timeout (in seconds) for detecting the second ring after the first detected ring. If automatic dialing is not used and Caller ID is enabled, the device seizes the line after detection of the second ring signal (allowing detection of caller ID sent between the first and the second rings). If the second ring signal is not received within this timeout, the device doesn't initiate a call to IP. If automatic dialing is used, the deviceinitiates a call to IP when the ringing signal is detected. The FXO line is seized only if the remote IP party answers the call. If the remote party doesn't answer the call and the second ring signal is not received within this timeout, the device releases the IP call. This parameter is typically set to between 5 and 8. The default is 8. **Notes:** ▪ This parameter is applicable only for Tel-to-IP calls. ▪ This timeout is calculated from the end of the ring until the start of the next ring. For example, if the ring cycle is two seconds on and four seconds off, the timeout value should be configured to five seconds (i.e., greater than the off time, e.g., four).
Web: Rings before Detecting	Determines the number of rings before the device starts detecting

Parameter	Description
Caller ID EMS: Rings Before Caller ID CLI: rings-b4-det-callerid **[RingsBeforeCallerID]**	Caller ID. ▪ **[0]** 0 = Before first ring. ▪ **[1]** 1 = (Default) After first ring. ▪ **[2]** 2 = After second ring.
Web/EMS: Guard Time Between Calls CLI: guard-time-btwn-calls **[GuardTimeBetweenCalls]**	Defines the time interval (in seconds) after a call has ended and a new call can be accepted for IP-to-Tel (FXO) calls. The valid range is 0 to 10. The default is 1. **Note:** Occasionally, after a call ends and on-hook is applied, a delay is required before placing a new call (and performing off-hook). This is necessary to prevent incorrect hook-flash detection or other glare phenomena.
Web: FXO Double Answer CLI: fxo-dbl-ans **[EnableFXODoubleAnswer]**	Enables the FXO Double Answer feature, which rejects (disconnects) incoming Tel (FXO)-to-IP collect calls and signals (informs) this call denial to the PSTN. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** This parameter can be configured for a Tel Profile.
Web: FXO Ring Timeout CLI: fxo-ring-timeout **[FXORingTimeout]**	Defines the delay (in msec) before the device generates a SIP INVITE (call) to the IP side upon detection of a RING_START event from the Tel (FXO) side. This occurs instead of waiting for a RING_END event. This feature is useful for telephony services that employ constant ringing (i.e., no RING_END is sent). For example, Ringdown circuit is a service that sends a constant ringing current over the line, instead of cadence-based 2 second on, 4 second off. For example, when a telephone goes off-hook, a phone at the other end instantly rings. If a RING_END event is received before the timeout expires, the device does not initiate a call and ignores the detected ring. The device ignores RING_END events detected after the timeout expires. The valid value range is 0 to 50 (msec), in steps of 100-msec. For example, a value of 50 represents 5 sec. The default value is 0 (i.e., standard ring operation - the FXO interface sends an INVITE upon receipt of the RING_END event). **Note:** This parameter can be configured for a Tel Profile.
[EnablePulseDialGeneration]	Enables pulse dialing generation to the analog side when dialing is received from the FXO side. ▪ **[0]** Disable = (Default) Device generates DTMF signals. ▪ **[1]** Enable = Generates pulse dialing. **Note:** For this parameter to take effect, a device reset is required.
[PulseDialGenerationBreakTime]	Defines the duration of the Break connection (off-hook) for FXO pulse dial generation. The valid value range is 20 to 120 (in msec). The default is 60. **Note:** For this parameter to take effect, a device reset is required.
[PulseDialGenerationMakeTime]	Defines the duration of the Make connection (on-hook) for FXO pulse dial generation. The valid value range is 20 to 120 (in msec). The default is 40. **Note:** For this parameter to take effect, a device reset is required.

Parameter	Description
[PulseDialGenerationInterDigitTime]	Defines the inter-digit duration (time between consecutively dialed digits) for FXO pulse dial generation. The valid value range is 300 to 1500 (in msec). The default is 700. **Note:** For this parameter to take effect, a device reset is required.

67.10.12 Trunk Groups and Routing Parameters

The routing parameters are described in the table below.

Table 67-69: Routing Parameters

Parameter	Description
Trunk Group Table	
Web: Trunk Group Table EMS: SIP Endpoints > Phones CLI: configure voip > gw hunt-or-trunk-group TrunkGroup **[TrunkGroup]**	This table parameter configures and activates the device's endpoints/Trunk channels. This is done by defining telephone numbers and assigning them to Trunk Groups. The format of the ini file table parameter is as follows: [TrunkGroup] FORMAT TrunkGroup_Index = TrunkGroup_TrunkGroupNum, TrunkGroup_FirstTrunkId, TrunkGroup_FirstBChannel, TrunkGroup_LastBChannel, TrunkGroup_FirstPhoneNumber, TrunkGroup_ProfileId, TrunkGroup_LastTrunkId, TrunkGroup_Module; [\TrunkGroup] For a description of this table, seeConfiguring Trunk Group Table on page 365. **Note:** Trunk Group ID 1 is denoted as 0 in the table.
Hunt Group Settings	
Web: Hunt Group Settings EMS: SIP Routing > Hunt Group CLI: configure voip > gw hunt-or-trunk-group trunk-group-setting **[TrunkGroupSettings]**	This table parameter configures the rules for channel allocation per Trunk Group. The format of the ini file table parameter is as follows: [TrunkGroupSettings] FORMAT TrunkGroupSettings_Index = TrunkGroupSettings_TrunkGroupId, TrunkGroupSettings_ChannelSelectMode, TrunkGroupSettings_RegistrationMode, TrunkGroupSettings_GatewayName, TrunkGroupSettings_ContactUser, TrunkGroupSettings_ServingIPGroup, TrunkGroupSettings_MWIInterrogationType, TrunkGroupSettings_TrunkGroupName; [\TrunkGroupSettings] For a description of this table, see "Configuring Hunt Group Settings" on page 367.
Web: Channel Select Mode EMS: Channel Selection Mode CLI: ch-select-mode **[ChannelSelectMode]**	Defines the method for allocating incoming IP-to-Tel calls to a channel. This parameter applies to the following: ▪ All Trunk Groups configured without a channel select mode in the Trunk Group Settings table (see "Configuring Hunt Group Settings" on page 367).

Parameter	Description
	▪ All channels and trunks configured without a Trunk Group ID. for all Trunk Groups channels that are configured without a Trunk Group ID,. ▪ **[0]** By Dest Phone Number ▪ **[1]** Cyclic Ascending (default) ▪ **[2]** Ascending ▪ **[3]** Cyclic Descending ▪ **[4]** Descending ▪ **[5]** Dest Number + Cyclic Ascending. ▪ **[6]** By Source Phone Number ▪ [7] Trunk Cyclic Ascending ▪ [8] Trunk & Channel Cyclic Ascending ▪ [9] Ring to Hunt Group ▪ [10] Select Trunk By Supplementary Service Table ▪ **[11]** Dest Number + Ascending For a detailed description of the parameter's options, see "Configuring Hunt Group Settings" on page 367.
Web: Default Destination Number CLI: dflt-dest-nb **[DefaultNumber]**	Defines the default destination phone number, which is used if the received message doesn't contain a called party number and no phone number is configured in the Trunk Group table (see Configuring the Trunk Group Table on page 365). This parameter is used as a starting number for the list of channels comprising all the device's Trunk Groups. The default is 1000.
Web: Source IP Address Input CLI: src-ip-addr-input **[SourceIPAddressInput]**	Determines which IP address the device uses to determine the source of incoming INVITE messages for IP-to-Tel routing. ▪ **[-1]** = (Default) Auto Decision - if the IP-to-IP feature is enabled, this parameter is automatically set to Layer 3 Source IP. If the IP-to-IP feature is disabled, this parameter is automatically set to SIP Contact Header (1). ▪ **[0]** SIP Contact Header = The IP address in the Contact header of the incoming INVITE message is used. ▪ **[1]** Layer 3 Source IP = The actual IP address (Layer 3) from where the SIP packet was received is used.
Web: Use Source Number As Display Name EMS: Display Name CLI: src-nb-as-disp-name **[UseSourceNumberAsDisplayName]**	Determines the use of Tel Source Number and Display Name for Tel-to-IP calls. ▪ **[0]** No = (Default) If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the IP Display Name remains empty. ▪ **[1]** Yes = If a Tel Display Name is received, the Tel Source Number is used as the IP Source Number and the Tel Display Name is used as the IP Display Name. If no Display Name is received from the Tel side, the Tel Source Number is used as the IP Source Number and also as the IP Display Name. ▪ **[2]** Overwrite = The Tel Source Number is used as the IP Source Number and also as the IP Display Name (even if

Parameter	Description
	the received Tel Display Name is not empty). - **[3]** Original = Similar to option [2], except that the operation is done before regular calling number manipulation.
Web/EMS: Use Display Name as Source Number CLI: disp-name-as-src-nb **[UseDisplayNameAsSourceNumber]**	Determines the use of Source Number and Display Name for IP-to-Tel calls. - **[0]** No = (Default) If IP Display Name is received, the IP Source Number is used as the Tel Source Number and the IP Display Name is used as the Tel Display Name. If no Display Name is received from IP, the Tel Display Name remains empty. - **[1]** Yes = If an IP Display Name is received, it is used as the Tel Source Number and also as the Tel Display Name, and Presentation is set to Allowed (0). If no Display Name is received from IP, the IP Source Number is used as the Tel Source Number and Presentation is set to Restricted (1). For example: When 'From: 100 <sip:200@201.202.203.204>' is received, the outgoing Source Number and Display Name are set to '100' and the Presentation is set to Allowed (0). When 'From: <sip:100@101.102.103.104>' is received, the outgoing Source Number is set to '100' and the Presentation is set to Restricted (1).
Web: ENUM Resolution CLI: enum-service-domain **[EnumService]**	Defines the ENUM service for translating telephone numbers to IP addresses or domain names (FQDN), for example, e164.arpa, e164.customer.net, or NRENum.net. The valid value is a string of up to 50 characters. The default is "e164.arpa". **Note:** For the Gateway / IP-to-IP application, ENUM-based routing is configured in the Outbound IP Routing table using the "ENUM" string value as the destination address to denote this parameter's value.
Web: Use Routing Table for Host Names and Profiles EMS: Use Routing Table For Host Names CLI: rte-tbl-4-host-names **[AlwaysUseRouteTable]**	Determines whether to use the device's routing table to obtain the URI host name and optionally, an IP profile (per call) even if a Proxy server is used. - **[0]** Disable = (Default) Don't use internal routing table. - **[1]** Enable = Use the Outbound IP Routing table. **Notes:** - This parameter appears only if the 'Use Default Proxy' parameter is enabled. - The domain name is used instead of a Proxy name or IP address in the INVITE SIP URI.
Web/EMS: Tel to IP Routing Mode CLI: tel2ip-rte-mode **[RouteModeTel2IP]**	For a description of this parameter, see "Configuring Outbound IP Routing Table" on page 397.
Outbound IP Routing Table	
Web: Outbound IP Routing Table EMS: SIP Routing > Tel to IP CLI: configure voip > gw routing tel2ip-	This table parameter configures the Outbound IP Routing Table for routing Tel-to-IP and IP-to-IP calls. The format of this parameter is as follows:

Parameter	Description
routing **[Prefix]**	[PREFIX] FORMAT PREFIX_Index = PREFIX_RouteName, PREFIX_DestinationPrefix, PREFIX_DestAddress, PREFIX_SourcePrefix, PREFIX_ProfileId, PREFIX_MeteringCode, PREFIX_DestPort, PREFIX_SrcIPGroupID, PREFIX_DestHostPrefix, PREFIX_DestIPGroupID, PREFIX_SrcHostPrefix, PREFIX_TransportType, PREFIX_SrcTrunkGroupID, PREFIX_DestSRD, PREFIX_CostGroup, PREFIX_ForkingGroup, PREFIX_CallSetupRulesSetId; [\PREFIX] For a detailed description of this table, see "Configuring Outbound IP Routing Table" on page 397.
Inbound IP Routing Table	
Web: Inbound IP Routing Table EMS: SIP Routing > IP to Hunt CLI: configure voip > gw routing ip2tel-routing **[PSTNPrefix]**	This table parameter configures the routing of IP-to-Trunk Groups (or inbound IP Groups). The format of the ini file table parameter is as follows: [PSTNPrefix] FORMAT PstnPrefix_Index = PstnPrefix_RouteName, PstnPrefix_DestPrefix, PstnPrefix_TrunkGroupId, PstnPrefix_SourcePrefix, PstnPrefix_SourceAddress, PstnPrefix_ProfileId, PstnPrefix_SrcIPGroupID, PstnPrefix_DestHostPrefix, PstnPrefix_SrcHostPrefix, PstnPrefix_SrcSRDID, PstnPrefix_TrunkId, PstnPrefix_CallSetupRulesSetId; [\PSTNPrefix] For a detailed description of this table, see "Configuring Inbound IP Routing Table" on page 406.
Web/EMS: IP to Tel Routing Mode CLI: ip2tel-rte-mode **[RouteModeIP2Tel]**	Determines whether to route IP calls to the Trunk Group (or IP Group) before or after manipulation of the destination number (configured in "Configuring Source/Destination Number Manipulation Rules" on page 373). ▪ **[0]** Route calls before manipulation = (Default) Calls are routed before the number manipulation rules are applied. ▪ **[1]** Route calls after manipulation = Calls are routed after the number manipulation rules are applied.
Web: IP Security EMS: Secure Call From IP CLI: ip-security **[SecureCallsFromIP]**	Determines the device's policy on accepting or blocking SIP calls (IP-to-Tel calls). This is useful in preventing unwanted SIP calls, SIP messages, and/or VoIP spam. ▪ **[0]** Disable = (Default) The device accepts all SIP calls. ▪ **[1]** Secure Incoming calls = The device accepts SIP calls (i.e., calls from the IP side) only from IP addresses that are defined in the Outbound IP Routing table or Proxy Set table, or IP addresses resolved from DNS servers from FQDN values defined in the Proxy Set table. All other incoming calls are rejected. ▪ **[2]** Secure All calls = The device accepts SIP calls only from IP addresses (in dotted-decimal notation format) that are defined in the Outbound IP Routing table or Proxy Set table, and rejects all other incoming calls. In addition, if an FQDN is defined in the routing table or Proxy Set table, the call is allowed to be sent only if the resolved DNS IP address appears in one of these tables;

Parameter	Description
	otherwise, the call is rejected. Therefore, the difference between this option and option [1] is that this option is concerned only about numerical IP addresses that are defined in the tables. **Note:** If this parameter is set to [1] or [2], when using Proxies or Proxy Sets, it is unnecessary to configure the Proxy IP addresses in the routing table. The device allows SIP calls received from the Proxy IP addresses even if these addresses are not configured in the routing table.
Web/EMS: Filter Calls to IP CLI: filter-calls-to-ip **[FilterCalls2IP]**	Enables filtering of Tel-to-IP calls when a Proxy Set is used. ▪ **[0]** Don't Filter = (Default) The device doesn't filter calls when using a proxy. ▪ **[1]** Filter = Filtering is enabled. When this parameter is enabled and a proxy is used, the device first checks the Outbound IP Routing table before making a call through the proxy. If the number is not allowed (i.e., number isn't listed in the table or a call restriction routing rule of IP address 0.0.0.0 is applied), the call is released. **Note:** When no proxy is used, this parameter must be disabled and filtering is according to the Outbound IP Routing table.
Web: IP-to-Tel Tagging Destination Dial Plan Index CLI: ip2tel-tagging-dst [IP2TelTaggingDestDialPlanIndex]	Defines the Dial Plan index in the Dial Plan file for called prefix tags for representing called number prefixes in Inbound Routing rules. The valid values are 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1 (i.e., no dial plan file used). For more information on this feature, see Dial Plan Prefix Tags for IP-to-Tel Routing on page 608. **Note:** This parameter is applicable only to digital interfaces.
Web: IP to Tel Tagging Source Dial Plan Index CLI: configure voip/gw routing general-setting/ip-to-tel-tagging-src [IP2TelTaggingSourceDialPlanIndex]	Defines the Dial Plan index in the Dial Plan file for calling prefix tags for representing calling number prefixes in Inbound Routing rules. The valid values are 0 to 7, where 0 denotes PLAN1, 1 denotes PLAN2, and so on. The default is -1 (i.e., no dial plan file used). For more information on this feature, see Dial Plan Prefix Tags for IP-to-Tel Routing on page 608. **Note:** This parameter is applicable only to digital interfaces.
CLI: etsi-diversion [EnableETSIDiversion]	Determines the method in which the Redirect Number is sent to the Tel side. ▪ [0] = (Default) Q.931 Redirecting Number Information Element (IE). ▪ [1] = ETSI DivertingLegInformation2 in a Facility IE.
Web: Add CIC CLI: add-cic **[AddCicAsPrefix]**	Determines whether to add the Carrier Identification Code (CIC) as a prefix to the destination phone number for IP-to-Tel calls. When this parameter is enabled, the 'cic' parameter in the incoming SIP INVITE can be used for IP-to-Tel routing decisions. It routes the call to the appropriate

Parameter	Description
	Trunk Group based on this parameter's value. - **[0]** No (default) - **[1]** Yes For digital interfaces: The SIP 'cic' parameter enables the transmission of the 'cic' parameter from the SIP network to the ISDN. The 'cic' parameter is a three- or four-digit code used in routing tables to identify the network that serves the remote user when a call is routed over many different networks. The 'cic' parameter is carried in the SIP INVITE and maps to the ISDN Transit Network Selection Information Element (TNS IE) in the outgoing ISDN Setup message (if the EnableCIC parameter is set to 1). The TNS IE identifies the requested transportation networks and allows different providers equal access support, based on customer choice. For example, as a result of receiving the below INVITE, the destination number after number manipulation is cic+167895550001: INVITE sip:5550001;cic=+16789@172.18.202.60:5060;user=phone SIP/2.0 **Note:** After the cic prefix is added, the Inbound IP Routing table can be used to route this call to a specific Trunk Group. The Destination Number IP to Tel Manipulation table must be used to remove this prefix before placing the call to the ISDN.
[FaxReroutingMode]	Enables re-routing of Tel-to-IP calls that are identified as fax calls. If a CNG tone is detected on the Tel side of a Tel-to-IP call, the prefix string "FAX" is appended to the destination number before routing and manipulation. If you enter the string value, "FAX" as the destination number in the Outbound IP Routing table, the routing rule is used to route the call and the destination number manipulation mechanism is used to remove the "FAX" prefix, if required. Note that the "FAX" prefix string in routing and manipulation tables is case-sensitive. If the initial INVITE used to establish the voice call (not fax) was already sent, a CANCEL (if not connected yet) or a BYE (if already connected) is sent to release the voice call. - [0] Disable (default) - [1] Rerouting without Delay - [2] Progress and Delay = Incoming ISDN calls are delayed until a CNG tone detection or timeout, set by the FaxReroutingDelay parameter. If the EnableComfortTone parameter is set to 1, a Q.931 Progress message with Protocol Discriminator set to 1 is sent to the PSTN and a comfort tone is played accordingly to the PSTN. When the timeout expires, the device sends an INVITE to a specific IP Group or to a fax server, according to the Outbound IP Routing table rules. This option is applicable only to ISDN. - [3] Connect and Delay = Incoming ISDN calls are delayed until a CNG tone detection or timeout, set by the FaxReroutingDelay parameter. A Q.931 Connect

Parameter	Description
	message is sent to the PSTN. If the EnableComfortTone parameter is set to 1, a comfort tone is played to the PSTN. When the timeout expires, the device sends an INVITE to a specific IP Group or to a fax server according to the Outbound IP Routing table rules. This option is applicable only to ISDN. **Note:** This parameter has replaced the EnableFaxRerouting parameter. For backward compatibility, the EnableFaxRerouting parameter set to 1 is equivalent to the FaxReroutingMode parameter set to 1.
[FaxReroutingDelay]	Defines the maximum time interval (in seconds) that the device waits for CNG detection before re-routing calls identified as fax calls to fax destinations (terminating fax machine). The valid value range is 1-10. The default is 5.
Call Forking Parameters	
Web/EMS: Forking Handling Mode CLI: forking-handling **[ForkingHandlingMode]**	Determines how the device handles the receipt of multiple SIP 18x forking responses for Tel-to-IP calls. The forking 18x response is the response with a different SIP to-tag than the previous 18x response. These responses are typically generated (initiated) by Proxy / Application servers that perform call forking, sending the device's originating INVITE (received from SIP clients) to several destinations, using the same Call ID. ▪ **[0]** Parallel handling = (Default) If SIP 18x with SDP is received, the device opens a voice stream according to the received SDP and disregards any subsequently received 18x forking responses (with or without SDP). If the first response is 180 without SDP, the device responds according to the PlayRBTone2TEL parameter and disregards the subsequent forking 18x responses. ▪ **[1]** Sequential handling = If 18x with SDP is received, the device opens a voice stream according to the received SDP. The device re-opens the stream according to subsequently received 18x responses with SDP, or plays a ringback tone if 180 response without SDP is received. If the first received response is 180 without SDP, the device responds according to the PlayRBTone2TEL parameter and processes the subsequent 18x forking responses. **Note:** Regardless of this parameter setting, once a SIP 200 OK response is received, the device uses the RTP information and re-opens the voice stream, if necessary.
Web: Forking Timeout CLI: forking-timeout **[ForkingTimeOut]**	Defines the timeout (in seconds) that is started after the first SIP 2xx response has been received for a User Agent when a Proxy server performs call forking (Proxy server forwards the INVITE to multiple SIP User Agents). The device sends a SIP ACK and BYE in response to any additional SIP 2xx received from the Proxy within this timeout. Once this timeout elapses, the device ignores any subsequent SIP 2xx. The number of supported forking calls per channel is 20. In

Parameter	Description
	other words, for an INVITE message, the device can receive up to 20 forking responses from the Proxy server. The valid range is 0 to 30. The default is 30.
Web: Tel2IP Call Forking Mode CLI: tel2ip-call-forking-mode **[Tel2IPCallForkingMode]**	Enables Tel-to-IP call forking, whereby a Tel call can be routed to multiple IP destinations. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** Once enabled, routing rules must be assigned Forking Groups in the Outbound IP Routing table.
CLI: configure voip/sip-definition advanced-settings/forking-delay-time-invite **[ForkingDelayTimeForInvite]**	Defines the interval (in seconds) to wait before sending INVITE messages to the other members of the forking group. The INVITE is immediately sent to the first member. The valid value range is 0 to 40. The default is 0 (i.e., sends immediately).

67.10.13 IP Connectivity Parameters

The IP connectivity parameters are described in the table below.

Table 67-70: IP Connectivity Parameters

Parameter	Description
Web: Enable Alt Routing Tel to IP EMS: Enable Alternative Routing CLI: alt-routing-tel2ip **[AltRoutingTel2IPEnable]**	Enables the Alternative Routing feature for Tel-to-IP calls. ▪ **[0]** Disable = (Default) Disables the Alternative Routing feature. ▪ **[1]** Enable = Enables the Alternative Routing feature. ▪ **[2]** Status Only = The Alternative Routing feature is disabled, but read-only information on the QoS of the destination IP addresses is provided.
Web: Alt Routing Tel to IP Mode EMS: Alternative Routing Mode CLI: alt-rte-tel2ip-mode **[AltRoutingTel2IPMode]**	Determines the IP Connectivity event(s) reason for triggering Alternative Routing. ▪ **[0]** None = Alternative routing is not used. ▪ **[1]** Connectivity = Alternative routing is performed if SIP OPTIONS message to the initial destination fails (determined according to the AltRoutingTel2IPConnMethod parameter). ▪ **[2]** QoS = Alternative routing is performed if poor QoS is detected. ▪ **[3]** Both = (Default) Alternative routing is performed if either SIP OPTIONS to initial destination fails, poor QoS is detected, or the DNS host name is not resolved. **Notes:** ▪ QoS is quantified according to delay and packet loss calculated according to previous calls. QoS statistics are reset if no new data is received within two minutes. ▪ To receive quality information (displayed in the 'Quality Status' and 'Quality Info.' fields in "Viewing IP Connectivity" on page 687) per destination, this parameter must be set to 2 or 3.
Web: Alt Routing Tel to IP	Determines the method used by the device for periodically

Parameter	Description
Connectivity Method EMS: Alternative Routing Telephone to IP Connection Method CLI: alt-rte-tel2ip-method **[AltRoutingTel2IPConnMethod]**	querying the connectivity status of a destination IP address. ▪ **[0]** ICMP Ping = (Default) Internet Control Message Protocol (ICMP) ping messages. ▪ **[1]** SIP OPTIONS = The remote destination is considered offline if the latest OPTIONS transaction timed out. Any response to an OPTIONS request, even if indicating an error, brings the connectivity status to online. **Note:** ICMP Ping is currently not supported for the IP Connectivity feature.
Web: Alt Routing Tel to IP Keep Alive Time EMS: Alternative Routing Keep Alive Time CLI: alt-rte-tel2ip-keep-alive **[AltRoutingTel2IPKeepAliveTime]**	Defines the time interval (in seconds) between SIP OPTIONS Keep-Alive messages used for the IP Connectivity application. The valid range is 5 to 2,000,000. The default is 60.
Web: Max Allowed Packet Loss for Alt Routing [%] CLI: mx-pkt-loss-4-alt-rte **[IPConnQoSMaxAllowedPL]**	Defines the packet loss (in percentage) at which the IP connection is considered a failure and Alternative Routing mechanism is activated. The default is 20%.
Web: Max Allowed Delay for Alt Routing [msec] CLI: mx-all-dly-4-alt-rte **[IPConnQoSMaxAllowedDelay]**	Defines the transmission delay (in msec) at which the IP connection is considered a failure and the Alternative Routing mechanism is activated. The range is 100 to 10,000. The default is 250.

67.10.14 Alternative Routing Parameters

The alternative routing parameters are described in the table below.

Table 67-71: Alternative Routing Parameters

Parameter	Description
Web: 3xx Use Alt Route Reasons CLI: configure voip/sip-definition advanced-settings/3xx-use-alt-route **[UseAltRouteReasonsFor3xx]**	Defines the handling of received SIP 3xx responses regarding call redirection to listed contacts in the Contact header. ▪ **[0]** No = (Default) Upon receipt of a 3xx response, the device tries each contact, one by one, listed in the Contact headers, until a successful destination is found. However, if a contact responds with a 486 or 600, the device does not try to redirect the call to next contact, and drops the call. ▪ **[1]** No if 6xx = Upon receipt of a 3xx response, the device tries each contact, one by one, listed in the Contact headers. However, if a 6xx Global Failure response is received during this process (e.g., 600 Busy Everywhere) the device does not try to redirect the call to the next contact, and drops the call. ▪ **[2]** Yes = Upon receipt of a 3xx response, the device redirects the call to the first contact listed in the Contact header. If the contact responds with a SIP response that is defined in the Reasons for Tel-to-IP Alternative Routing table, the device tries to redirect the call to the next contact, and so on. If a contact responds with a response that is not configured in the table, the device does not try to redirect the call to the next contact, and

Parameter	Description
	drops the call.
Web/EMS: Redundant Routing Mode CLI: redundant-routing-m **[RedundantRoutingMode]**	Determines the type of redundant routing mechanism when a call can't be completed using the main route. ▪ **[0]** Disable = No redundant routing is used. If the call can't be completed using the main route (using the active Proxy or the first matching rule in the Routing table), the call is disconnected. ▪ **[1]** Routing Table = (Default) Internal routing table is used to locate a redundant route. ▪ **[2]** Proxy = Proxy list is used to locate a redundant route. **Note:** To implement the Redundant Routing Mode mechanism, you first need to configure the parameter AltRouteCauseTEL2IP (Reasons for Alternative Routing table).
[DisconnectCallwithPIifAlt]	Defines when the device sends the IP-to-Tel call to an alternative route (if configured) when it receives an ISDN Q.931 Disconnect message from the Tel side. ▪ [0] (Default) = The device forwards early media to the IP side if Disconnect includes PI, and disconnects the call when a Release message is received. Only after the call is disconnected does the device send the call to an alternative route. ▪ [1] = The device immediately sends the call to the alternative route. For more information, see Alternative Routing upon ISDN Disconnect on page 420.
[EnableAltMapTel2IP]	Enables different Tel-to-IP destination number manipulation rules per routing rule when several (up to three) Tel-to-IP routing rules are defined and if alternative routing using release causes is used. For example, if an INVITE message for a Tel-to-IP call is returned with a SIP 404 Not Found response, the call can be re-sent to a different destination number (as defined using the parameter NumberMapTel2IP). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Web/EMS: Alternative Routing Tone Duration [ms] CLI: alt-rte-tone-duration [AltRoutingToneDuration]	Defines the duration (in milliseconds) for which the device plays a tone to the endpoint on each attempt for Tel-to-IP alternative routing. When the device finishes playing the tone, a new SIP INVITE message is sent to the new IP destination. The tone played is the call forward tone (Tone Type #25 in the CPT file). The valid range is 0 to 20,000. The default is 0 (i.e., no tone is played). **Note:** This parameter is applicable only to Tel-to-IP alternative routing.
Reasons for Alternative Tel-to-IP Routing Table	
Web: Reasons for Alternative Routing EMS: Alt Route Cause Tel to IP CLI: configure voip > gw manipulations general-setting alt-route-cause-tel2ip **[AltRouteCauseTel2IP]**	This table parameter configures SIP call failure reason values received from the IP side. If an IP call is released as a result of one of these reasons, the device attempts to locate an alternative IP route for the call in the Outbound IP Routing table (if a Proxy is not used) or used as a redundant Proxy (you need to set the parameter RedundantRoutingMode to 2). The release reason for Tel-to-IP calls is provided in SIP 4xx, 5xx, and 6xx response codes. The format of the ini file table parameter is as follows: [AltRouteCauseTel2IP]

Parameter	Description
	FORMAT AltRouteCauseTel2IP_Index = AltRouteCauseTel2IP_ReleaseCause; [\AltRouteCauseTel2IP] For example: AltRouteCauseTel2IP 0 = 486; (Busy Here) AltRouteCauseTel2IP 1 = 480; (Temporarily Unavailable) AltRouteCauseTel2IP 2 = 408; (No Response) For a detailed description of this table, see "Alternative Routing Based on SIP Responses" on page 413.
Reasons for Alternative IP-to-Tel Routing Table	
Web: Reasons for Alternative IP-to-Tel Routing EMS: Alt Route Cause IP to Tel CLI: configure voip > gw manipulations general-setting alt-route-cause-ip2tel **[AltRouteCauseIP2Tel]**	This table parameter configures call failure reason values received from the Tel side (in Q.931 presentation). If a call is released as a result of one of these reasons, the device attempts to locate an alternative Trunk Group for the call in the Inbound IP Routing table. The format of the ini file table parameter is as follows: [AltRouteCauseIP2Tel] FORMAT AltRouteCauseIP2Tel_Index = AltRouteCauseIP2Tel_ReleaseCause; [\AltRouteCauseIP2Tel] For example: AltRouteCauseIP2Tel 0 = 3 (No Route to Destination) AltRouteCauseIP2Tel 1 = 1 (Unallocated Number) AltRouteCauseIP2Tel 2 = 17 (Busy Here) AltRouteCauseIP2Tel 2 = 27 (Destination Out of Order) For a detailed description of this table, see "Alternative Routing to Trunk upon Q.931 Call Release Cause Code" on page 417.
Forward On Busy Trunk Destination Table	
Web/EMS: Forward On Busy Trunk Destination CLI: configure voip > gw routing fwd-on-bsy-trk-dest **[ForwardOnBusyTrunkDest]**	This table parameter configures the Forward On Busy Trunk Destination table. This table allows you to define an alternative IP destination if a trunk is busy for IP-to-Tel calls. The format of the ini file table parameter is as follows: [ForwardOnBusyTrunkDest] FORMAT ForwardOnBusyTrunkDest_Index = ForwardOnBusyTrunkDest_TrunkGroupId, ForwardOnBusyTrunkDest_ForwardDestination; [\ForwardOnBusyTrunkDest] For example, the below configuration forwards IP-to-Tel calls to destination user “112” at host IP address 10.13.4.12, port 5060, using transport protocol TCP, if Trunk Group ID 2 is unavailable: ForwardOnBusyTrunkDest 1 = 2, 112@10.13.4.12:5060;transport=tcp; For a detailed description of this table, see "Alternative Routing to IP Destination upon Busy Trunk" on page 418.

67.10.15 Number Manipulation Parameters

The number manipulation parameters are described in the table below.

Table 67-72: Number Manipulation Parameters

Parameter	Description
[ManipulateIP2PSTNReferTo]	Enables the manipulation of the called party (destination) number according to the SIP Refer-To header received by the device for TDM (PSTN) blind transfer. The number in the SIP Refer-To header is manipulated for all types of blind transfers to the PSTN (TBCT, ECT, RLT, QSIG, FXO, and CAS). ▪ [0] Disable (default) ▪ [1] Enable During the blind transfer, the device initiates a new call to the PSTN and the destination number of this call can be manipulated if this parameter is enabled. When enabled, the manipulation is done as follows: 1 If you configure a value for the xferPrefix parameter, then this value (string) is added as a prefix to the number in the Refer-To header. 2 This called party number is then manipulated using the IP-to-Tel Destination Phone Number Manipulation table. The source number of the transferred call is taken from the original call, according to its initial direction: ✓ Source number of the original call if it is a Tel-to-IP call ✓ Destination number of the original call if it is an IP-to-Tel call This source number can also be used as the value for the 'Source Prefix' field in the IP-to-Tel Destination Phone Number Manipulation table. The local IP address is used as the value for the 'Source IP Address' field. **Note:** This manipulation does not affect IP-to-Trunk Group routing rules.
Web: Use EndPoint Number As Calling Number Tel2IP EMS: Use EP Number As Calling Number Tel to IP CLI: epn-as-cpn-tel2ip [UseEPNumAsCallingNumTel2IP]	Enables the use of the B-channel number as the calling number (sent in the From field of the INVITE) instead of the number received in the Q.931 Setup message, for Tel-to-IP calls. ▪ [0] Disable (default) ▪ [1] Enable For example, if the incoming calling party number in the Q.931 Setup message is "12345" and the B-channel number is 17, then the outgoing INVITE From header is set to "17" instead of "12345". **Note:** When enabled, this feature is applied before routing and manipulation on the source number.
Web: Use EndPoint Number As Calling Number IP2Tel EMS: Use EP Number As Calling Number IP to Tel CLI: epn-as-cpn-ip2tel [UseEPNumAsCallingNumIP2Tel]	Enables the use of the B-channel number as the calling party number (sent in the Q.931 Setup message) instead of the number received in the From header of the INVITE, for IP-to-Tel calls. ▪ [0] Disable (default) ▪ [1] Enable For example, if the incoming INVITE From header contains "12345" and the destined B-channel number is 17, then the outgoing calling party number in the Q.931 Setup message is set to "17" instead of "12345". **Note:** When enabled, this feature is applied after routing and

Parameter	Description
	manipulation on the source number (i.e., just before sending to the Tel side).
Web: Tel2IP Default Redirect Reason CLI: tel-to-ip-dflt-redir-rsn [Tel2IPDefaultRedirectReason]	Determines the default redirect reason for Tel-to-IP calls when no redirect reason (or “unknown”) exists in the received Q931 ISDN Setup message. The device includes this default redirect reason in the SIP History-Info header of the outgoing INVITE. If a redirect reason exists in the received Setup message, this parameter is ignored and the device sends the INVITE message with the reason according to the received Setup message. If this parameter is not configured (-1), the outgoing INVITE is sent with the redirect reason as received in the Setup message (if none or “unknown” reason, then without a reason). ▪ [-1] Not Configured = (Default) Received redirect reason is not changed ▪ [1] Busy = Call forwarding busy ▪ [2] No Reply = Call forwarding no reply ▪ [9] DTE Out of Order = Call forwarding DTE out of order ▪ [10] Deflection = Call deflection ▪ [15] Systematic/Unconditional = Call forward unconditional
Web: Redirect Number SIP to TEL EMS: Set IP To Tel Redirect Screening Indicator CLI: redir-nb-si-2tel [SetIp2TelRedirectScreeningInd]	Determines the value of the Redirect Number screening indicator in ISDN Setup messages. ▪ [-1] Not Configured (default) ▪ [0] User Provided ▪ [1] User Passed ▪ [2] User Failed ▪ [3] Network Provided **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Set IP-to-TEL Redirect Reason CLI: ip2tel-redir-reason [SetIp2TelRedirectReason]	Defines the redirect reason for IP-to-Tel calls. If redirect (diversion) information is received from the IP, the redirect reason is set to the value of this parameter before the device sends it on to the Tel. ▪ [-1] Not Configured (default) ▪ [0] Unkown ▪ [1] Busy ▪ [2] No Reply ▪ [3] Network Busy ▪ [4] Deflection ▪ [9] DTE out of Order ▪ [10] Forwarding DTE ▪ [13] Transfer ▪ [14] PickUp ▪ [15] Systematic/Unconditional **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Set TEL-to-IP Redirect Reason CLI: tel2ip-redir-reason [SetTel2IpRedirectReason]	Defines the redirect reason for Tel-to-IP calls. If redirect (diversion) information is received from the Tel, the redirect reason is set to the value of this parameter before the device sends it on to the IP. ▪ [-1] Not Configured (default) ▪ [0] Unkown

Parameter	Description
	▪ [1] Busy ▪ [2] No Reply ▪ [3] Network Busy ▪ [4] Deflection ▪ [9] DTE out of Order ▪ [10] Forwarding DTE ▪ [13] Transfer ▪ [14] PickUp ▪ [15] Systematic/Unconditional **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Send Screening Indicator to IP EMS: Screening Indicator To IP [ScreeningInd2IP]	Overrides the calling party's number (CPN) screening indication in the received ISDN SETUP message for Tel-to-IP calls. ▪ [-1] Not Configured = (Default) Not configured (interworking from ISDN to IP) or set to 0 for CAS. ▪ [0] User Provided = CPN set by user, but not screened (verified). ▪ [1] User Passed = CPN set by user, verified and passed. ▪ [2] User Failed = CPN set by user, and verification failed. ▪ [3] Network Provided = CPN set by network. **Note:** This parameter is applicable only if the Remote Party ID (RPID) header is enabled.
Web: Send Screening Indicator to ISDN EMS: Screening Indicator To ISDN [ScreeningInd2ISDN]	Overrides the screening indicator of the calling party's number for IP-to-Tel ISDN calls. ▪ [-1] Not Configured = (Default) Not configured (interworking from IP to ISDN). ▪ [0] User Provided = user provided, not screened. ▪ [1] User Passed = user provided, verified and passed. ▪ [2] User Failed = user provided, verified and failed. ▪ [3] Network Provided = network provided **Note:** This parameter is applicable only to digital PSTN interfaces (ISDN).
Web: Copy Destination Number to Redirect Number EMS: Copy Dest to Redirect Number CLI: cp-dst-nb-2-redir-nb **[CopyDest2RedirectNumber]**	Enables the device to copy the received ISDN (digital interfaces) called number to the outgoing SIP Diversion header for Tel-to-IP calls (even if a Redirecting Number IE is not received in the ISDN Setup message, for digital interfaces). Therefore, the called number is used as a redirect number. Call redirection information is typically used for Unified Messaging and voice mail services to identify the recipient of a message. ▪ **[0]** Don't copy = (Default) Disable. ▪ **[1]** Copy after phone number manipulation = Copies the called number after manipulation. The device first performs Tel-to-IP destination phone number manipulation (i.e., on the SIP To header), and only then copies the manipulated called number to the SIP Diversion header for the Tel-to-IP call. Therefore, with this option, the called and redirect numbers are identical. ▪ **[2]** Copy before phone number manipulation = Copies the called number before manipulation. The device first copies the original called number to the SIP Diversion header, and then performs Tel-to-IP destination phone number manipulation. Therefore, this allows you to have different numbers for the called (i.e., SIP To header) and redirect (i.e., SIP Diversion header) numbers.

Parameter	Description
	Notes: ▪ For digital interfaces: If the incoming ISDN-to-IP call includes a Redirect Number, this number is overridden by the new called number if this parameter is set to [1] or [2]. ▪ You can also use this feature for IP-to-Tel calls, by configuring this parameter per IP Profile (IpProfile_CopyDest2RedirectNum). For more information, see Configuring IP Profiles on page 314.
CLI: rep-calling-w-redir disc-on-bsy-tone-i [ReplaceCallingWithRedirectNumber]	Enables the replacement of the calling number with the redirect number for ISDN-to-IP calls. ▪ [0] = Disable (default) ▪ [1] = The calling name is removed and left blank. The outgoing INVITE message excludes the redirect number that was used to replace the calling number. The replacement is done only if a redirect number is present in the incoming Tel call. ▪ [2] = Manipulation is done on the new calling party number (after manipulation of the original calling party number, using the Tel2IPSourceNumberMappingDialPlanIndex parameter), but before the regular calling or redirect number manipulation: ✓ If a redirect number exists, it replaces the calling party number. If there is no redirect number, the calling number is left unchanged. ✓ If there is a calling “display” name, it remains unchanged. ✓ The redirect number remains unchanged and is included in the SIP Diversion header.
Web/EMS: Add Trunk Group ID as Prefix CLI: trkgrpid-prefix **[AddTrunkGroupAsPrefix]**	Determines whether the Trunk Group ID is added as a prefix to the destination phone number (i.e., called number) for Tel-to-IP calls. ▪ **[0]** No = (Default) Don't add Trunk Group ID as prefix. ▪ **[1]** Yes = Add Trunk Group ID as prefix to called number. **Notes:** ▪ This option can be used to define various routing rules. ▪ To use this feature, you must configure the Trunk Group IDs (see Configuring Trunk Group Table on page 365).
Web: Add Trunk ID as Prefix EMS: Add Port ID As Prefix CLI: trk-id-as-prefix **[AddPortAsPrefix]**	Determines whether or not the slot number/port number/Trunk ID is added as a prefix to the called (destination) number for Tel-to-IP calls. ▪ **[0]** No (Default) ▪ **[1]** Yes If enabled, the device adds the following prefix to the called phone number: slot number (a single digit in the range of 1 to 6) and port number/Trunk ID (single digit in the range 1 to 8). For example, for the first trunk/channel located in the first slot, the number "11" is added as the prefix. This option can be used to define various routing rules.
Web/EMS: Add Trunk Group ID as Prefix to Source CLI: trkgrpid-pref2source **[AddTrunkGroupAsPrefixToSource]**	Determines whether the device adds the Trunk Group ID (from where the call originated) as the prefix to the calling number (i.e. source number). ▪ **[0]** No (default) ▪ **[1]** Yes
Web: Replace Empty Destination with B-channel	Determines whether the internal channel number is used as the destination number if the called number is missing.

Parameter	Description
Phone Number EMS: Replace Empty Dst With Port Number CLI: empty-dst-w-bch-nb [ReplaceEmptyDstWithPortNumber]	▪ [0] No (default) ▪ [1] Yes **Note:** This parameter is applicable only to Tel-to-IP calls and if the called number is missing.
[CopyDestOnEmptySource]	Determines whether the destination number is copied to the source number if no source number is present, for Tel-to-IP calls. ▪ [0] = (Default) Source Number is left empty. ▪ [1] = If the Source Number of a Tel-to-IP call is empty, the Destination Number is copied to the Source Number.
Web: Add NPI and TON to Calling Number EMS: Add NPI And TON As Prefix To Calling Number CLI: npi-n-ton-to-cng-nb [AddNPIandTON2CallingNumber]	Determines whether the Numbering Plan Indicator (NPI) and Type of Numbering (TON) are added to the Calling Number for Tel-to-IP calls. ▪ [0] No = (Default) Do not change the Calling Number. ▪ [1] Yes = Add NPI and TON to the Calling Number ISDN Tel-to-IP call. For example: After receiving a Calling Number of 555, NPI of 1, and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
Web: Add NPI and TON to Called Number EMS: Add NPI And TON As Prefix To Called Number CLI: npi-n-ton-to-cld-nb [AddNPIandTON2CalledNumber]	Determines whether NPI and TON are added to the Called Number for Tel-to-IP calls. ▪ [0] No = (Default) Do not change the Called Number. ▪ [1] Yes = Add NPI and TON to the Called Number of ISDN Tel-to-IP call. For example: After receiving a Called Number of 555, NPI of 1 and TON of 3, the modified number becomes 13555. This number can later be used for manipulation and routing.
Web: Add NPI and TON to Redirect Number CLI: np-n-ton-2-redirnb [AddNPIandTON2RedirectNumber]	Determines whether the NPI and TON values are added as the prefix to the Redirect number in INVITE messages' Diversion or History-Info headers, for ISDN Tel-to-IP calls. ▪ [0] Yes (Default) ▪ [1] No
Web: IP to Tel Remove Routing Table Prefix EMS: Remove Prefix CLI: ip2tel-rmv-rte-tbl **[RemovePrefix]**	Determines whether or not the device removes the prefix (as configured in the Inbound IP Routing table - see "Configuring Inbound IP Routing Table" on page 406) from the destination number for IP-to-Tel calls, before sending it to the Tel. ▪ **[0]** No (default) ▪ **[1]** Yes For example: To route an incoming IP-to-Tel call with destination number "21100", the Inbound IP Routing table is scanned for a matching prefix. If such a prefix is found (e.g., "21"), then before the call is routed to the corresponding Trunk Group, the prefix "21" is removed from the original number, and therefore, only "100" remains. **Notes:** ▪ This parameter is applicable only if number manipulation is performed after call routing for IP-to-Tel calls (i.e., RouteModeIP2Tel parameter is set to 0). ▪ Similar operation (of removing the prefix) is also achieved by using the usual number manipulation rules.

Parameter	Description
Web/EMS: Swap Redirect and Called Numbers CLI: swap-rdr-n-called-nb [SwapRedirectNumber]	- [0] No = (Default) Don't change numbers. - [1] Yes = Incoming ISDN call that includes a redirect number (sometimes referred to as 'original called number') uses the redirect number instead of the called number.
[UseReferredByForCallingNumber]	Determines whether the device uses the number from the URI in the SIP Referred-By header as the calling number in the outgoing Q.931 Setup message, when SIP REFER messages are received. - [0] = (Default) No - [1] = Yes **Notes:** - This parameter is applicable to all ISDN (TBCT, RLT, ECT) and CAS blind call transfers (except for in-band) and when the device receives SIP REFER messages with a Referred-By header. - This manipulation is done before regular IP-to-Tel source number manipulation.
[SwapTel2IPCalled&Calling Numbers]	Determines whether the device swaps the calling and called numbers received from the Tel side (for Tel-to-IP calls). The SIP INVITE message contains the swapped numbers. - **[0]** = (Default) Disabled - **[1]** = Swap calling and called numbers **Note:** This parameter can also be configured for a Tel Profile (in the Tel Profile table).
Web/EMS: Add Prefix to Redirect Number CLI: add-pref-to-redir-nb [Prefix2RedirectNumber]	Defines a string prefix that is added to the Redirect number received from the Tel side. This prefix is added to the Redirect Number in the SIP Diversion header. The valid range is an 8-character string. By default, no value is defined.
Web: Add Number Plan and Type to RPI Header EMS: Add Ton 2 RPI CLI: np-n-type-to-rpi-hdr **[AddTON2RPI]**	Determines whether the TON/PLAN parameters are included in the Remote-Party-ID (RPID) header. - **[0]** No - **[1]** Yes (default) If the Remote-Party-ID header is enabled (EnableRPIHeader = 1) and AddTON2RPI = 1, it's possible to configure the calling and called number type and number plan using the Number Manipulation tables for Tel-to-IP calls.
Web/EMS: Source Manipulation Mode CLI: src-manipulation **[SourceManipulationMode]**	Determines the SIP headers containing the source number after manipulation: - **[0]** = (Default) The SIP From and P-Asserted-Identity headers contain the source number after manipulation. - **[1]** = Only SIP From header contains the source number after manipulation, while the P-Asserted-Identity header contains the source number before manipulation.
Calling Name Manipulations IP-to-Tel Table	
CLI: configure voip > gw manipulations calling-name-map-ip2tel **[CallingNameMapIp2Tel]**	Configures rules for manipulating the calling name (caller ID) in the received SIP message for IP-to-Tel calls. This can include modifying or removing the calling name. The format of this table ini file parameter is as follows: [CallingNameMapIp2Tel] FORMAT CallingNameMapIp2Tel_Index =

Parameter	Description
	CallingNameMapIp2Tel_ManipulationName, CallingNameMapIp2Tel_DestinationPrefix, CallingNameMapIp2Tel_SourcePrefix, CallingNameMapIp2Tel_CallingNamePrefix, CallingNameMapIp2Tel_SourceAddress, CallingNameMapIp2Tel_RemoveFromLeft, CallingNameMapIp2Tel_RemoveFromRight, CallingNameMapIp2Tel_LeaveFromRight, CallingNameMapIp2Tel_Prefix2Add, CallingNameMapIp2Tel_Suffix2Add; [\CallingNameMapIp2Tel] For a detailed description of this table, see "Configuring SIP Calling Name Manipulation" on page 380.
Calling Name Manipulations Tel-to-IP Table	
CLI: configure voip > gw manipulations calling-name-map-tel2ip **[CallingNameMapTel2Ip]**	This table parameter configures rules for manipulating the calling name (caller ID) for Tel-to-IP calls. This can include modifying or removing the calling name. [CallingNameMapTel2Ip] FORMAT CallingNameMapTel2Ip_Index = CallingNameMapTel2Ip_ManipulationName, CallingNameMapTel2Ip_DestinationPrefix, CallingNameMapTel2Ip_SourcePrefix, CallingNameMapTel2Ip_CallingNamePrefix, CallingNameMapTel2Ip_SrcTrunkGroupID, CallingNameMapTel2Ip_SrcIPGroupID, CallingNameMapTel2Ip_RemoveFromLeft, CallingNameMapTel2Ip_RemoveFromRight, CallingNameMapTel2Ip_LeaveFromRight, CallingNameMapTel2Ip_Prefix2Add, CallingNameMapTel2Ip_Suffix2Add; [\CallingNameMapTel2Ip] For a detailed description of this table, see "Configuring SIP Calling Name Manipulation" on page 380.
Destination Phone Number Manipulation for IP-to-Tel Calls Table	
Web: Destination Phone Number Manipulation Table for IP > Tel Calls EMS: SIP Manipulations > Destination IP to Telcom CLI: configure voip > gw manipulations NumberMapIp2Tel2 **[NumberMapIP2Tel]**	This table parameter manipulates the destination number of IP-to-Tel calls. The format of the ini file table parameter is as follows: [NumberMapIp2Tel] FORMAT NumberMapIp2Tel_Index = NumberMapIp2Tel_ManipulationName, NumberMapIp2Tel_DestinationPrefix, NumberMapIp2Tel_SourcePrefix, NumberMapIp2Tel_SourceAddress, NumberMapIp2Tel_NumberType, NumberMapIp2Tel_NumberPlan, NumberMapIp2Tel_RemoveFromLeft, NumberMapIp2Tel_RemoveFromRight, NumberMapIp2Tel_LeaveFromRight, NumberMapIp2Tel_Prefix2Add, NumberMapIp2Tel_Suffix2Add, NumberMapIp2Tel_IsPresentationRestricted; [\NumberMapIp2Tel] For a detailed description of this table, see "Configuring Source/Destination Number Manipulation" on page 373.
EMS: Perform Additional IP2TEL Destination	Enables additional destination number manipulation for IP-to-Tel calls. The additional manipulation is done on the initially manipulated

Parameter	Description
Manipulation CLI: prfm-ip-to-tel-dst-map **[PerformAdditionalIP2TELDestinationManipulation]**	destination number, and this additional rule is also configured in the manipulation table (NumberMapIP2Tel parameter). This enables you to configure only a few manipulation rules for complex number manipulation requirements (that generally require many rules). ▪ **[0]** = Disable (default) ▪ **[1]** = Enable
Destination Phone Number Manipulation for Tel-to-IP Calls Table	
Web: Destination Phone Number Manipulation Table for Tel > IP Calls EMS: SIP Manipulations > Destination Telcom to IPs CLI: configure voip > gw manipulations NumberMapTel2Ip **[NumberMapTel2IP]**	This table parameter manipulates the destination number of Tel-to-IP calls. The format of the ini file table parameter is as follows: [NumberMapTel2Ip] FORMAT NumberMapTel2Ip_Index = NumberMapTel2Ip_ManipulationName, NumberMapTel2Ip_DestinationPrefix, NumberMapTel2Ip_SourcePrefix, NumberMapTel2Ip_SourceAddress, NumberMapTel2Ip_NumberType, NumberMapTel2Ip_NumberPlan, NumberMapTel2Ip_RemoveFromLeft, NumberMapTel2Ip_RemoveFromRight, NumberMapTel2Ip_LeaveFromRight, NumberMapTel2Ip_Prefix2Add, NumberMapTel2Ip_Suffix2Add, NumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_ SrcIPGroupID; [\NumberMapTel2Ip] For a detailed description of this table, see "Configuring Source/Destination Number Manipulation" on page 373.
Source Phone Number Manipulation for IP-to-Tel Calls Table	
Web: Source Phone Number Manipulation Table for IP > Tel Calls EMS: SIP Manipulations > Source IP to Telcom CLI: configure voip > gw manipulations SourceNumberMapIp2Tel **[SourceNumberMapIP2Tel]**	This *parameter* table manipulates the source number for IP-to-Tel calls. The format of the ini file table parameter is as follows: [SourceNumberMapIp2Tel] FORMAT SourceNumberMapIp2Tel_Index = SourceNumberMapIp2Tel_ManipulationName, SourceNumberMapIp2Tel_DestinationPrefix, SourceNumberMapIp2Tel_SourcePrefix, SourceNumberMapIp2Tel_SourceAddress, SourceNumberMapIp2Tel_NumberType, SourceNumberMapIp2Tel_NumberPlan, SourceNumberMapIp2Tel_RemoveFromLeft, SourceNumberMapIp2Tel_RemoveFromRight, SourceNumberMapIp2Tel_LeaveFromRight, SourceNumberMapIp2Tel_Prefix2Add, SourceNumberMapIp2Tel_Suffix2Add, SourceNumberMapIp2Tel_IsPresentationRestricted; [\SourceNumberMapIp2Tel] For a detailed description of this table, see "Configuring Source/Destination Number Manipulation" on page 373.
EMS: Perform Additional IP2TEL Source Manipulation CLI: prfm-ip-to-tel-src-map **[PerformAdditionalIP2TELSourceManipulation]**	Enables additional source number manipulation for IP-to-Tel calls. The additional manipulation is done on the initially manipulated source number, and this additional rule is also configured in the manipulation table (SourceNumberMapIP2Tel parameter). This enables you to configure only a few manipulation rules for complex number manipulation requirements (that generally require many

Parameter	Description
	rules). - **[0]** = Disable (default) - **[1]** = Enable
Source Phone Number Manipulation for Tel-to-IP Calls Table	
Web: Source Phone Number Manipulation Table for Tel > IP Calls EMS: SIP Manipulations > Source Telcom to IP CLI: configure voip > gw manipulations SourceNumberMapTel2Ip **[SourceNumberMapTel2IP]**	This table parameter manipulates the source phone number for Tel-to-IP calls. The format of the ini file table parameter is as follows: [SourceNumberMapTel2Ip] FORMAT SourceNumberMapTel2Ip_Index = SourceNumberMapTel2Ip_ManipulationName, SourceNumberMapTel2Ip_DestinationPrefix, SourceNumberMapTel2Ip_SourcePrefix, SourceNumberMapTel2Ip_SourceAddress, SourceNumberMapTel2Ip_NumberType, SourceNumberMapTel2Ip_NumberPlan, SourceNumberMapTel2Ip_RemoveFromLeft, SourceNumberMapTel2Ip_RemoveFromRight, SourceNumberMapTel2Ip_LeaveFromRight, SourceNumberMapTel2Ip_Prefix2Add, SourceNumberMapTel2Ip_Suffix2Add, SourceNumberMapTel2Ip_IsPresentationRestricted, NumberMapTel2Ip_SrcTrunkGroupID, NumberMapTel2Ip_SrcIPGroupID; [\SourceNumberMapTel2Ip] For a detailed description of this table, see "Configuring Source/Destination Number Manipulation" on page 373.
Redirect Number IP -to-Tel Table	
Web: Redirect Number IP -> Tel EMS: Redirect Number Map IP to Tel CLI: configure voip > gw manipulations redirect-number-map-ip2tel [RedirectNumberMapIp2Tel]	This table parameter manipulates the redirect number for IP-to-Tel calls. The format of the ini file table parameter is as follows: [RedirectNumberMapIp2Tel] FORMAT RedirectNumberMapIp2Tel_Index = RedirectNumberMapIp2Tel_ManipulationName, RedirectNumberMapIp2Tel_DestinationPrefix, RedirectNumberMapIp2Tel_RedirectPrefix, RedirectNumberMapIp2Tel_SourceAddress, RedirectNumberMapIp2Tel_SrcHost, RedirectNumberMapIp2Tel_DestHost, RedirectNumberMapIp2Tel_NumberType, RedirectNumberMapIp2Tel_NumberPlan, RedirectNumberMapIp2Tel_RemoveFromLeft, RedirectNumberMapIp2Tel_RemoveFromRight, RedirectNumberMapIp2Tel_LeaveFromRight, RedirectNumberMapIp2Tel_Prefix2Add, RedirectNumberMapIp2Tel_Suffix2Add, RedirectNumberMapIp2Tel_IsPresentationRestricted; [\RedirectNumberMapIp2Tel] For a description of this table, see Configuring Redirect Number Manipulation on page 383.
Redirect Number Tel-to-IP Table	
Web: Redirect Number Tel -> IP EMS: Redirect Number Map	This table parameter manipulates the Redirect Number for Tel-to-IP calls. The format of the ini file table parameter is as follows: [RedirectNumberMapTel2Ip]

Parameter	Description
Tel to IP CLI: configure voip > gw manipulations redirect-number-map-tel2ip **[RedirectNumberMapTel2IP]**	FORMAT RedirectNumberMapTel2Ip_Index = RedirectNumberMapTel2Ip_ManipulationName, RedirectNumberMapTel2Ip_DestinationPrefix, RedirectNumberMapTel2Ip_RedirectPrefix, RedirectNumberMapTel2Ip_RemoveFromLeft, RedirectNumberMapTel2Ip_RemoveFromRight, RedirectNumberMapTel2Ip_LeaveFromRight, RedirectNumberMapTel2Ip_Prefix2Add, RedirectNumberMapTel2Ip_Suffix2Add, RedirectNumberMapTel2Ip_IsPresentationRestricted, RedirectNumberMapTel2Ip_SrcTrunkGroupID, RedirectNumberMapTel2Ip_SrcIPGroupID; [\RedirectNumberMapTel2Ip] For a description of this table, see "Configuring Redirect Number Manipulation" on page 383.
Phone Context Table	
Web: Phone Context Table EMS: SIP Manipulations > Phone Context CLI: configure voip > gw manipulations phone-context-table **[PhoneContext]**	This table parameter configures the Phone Context table. This parameter maps NPI and TON to the SIP 'phone-context' parameter, and vice versa. The format for this parameter is as follows: [PhoneContext] FORMAT PhoneContext_Index = PhoneContext_Npi, PhoneContext_Ton, PhoneContext_Context; [\PhoneContext] For example: PhoneContext 0 = 0,0,unknown.com PhoneContext 1 = 1,1,host.com PhoneContext 2 = 9,1,na.e164.host.com For a detailed description of this table, see "Configuring NPI/TON-SIP Phone-Context Mapping Rules" on page 388.
Web/EMS: Add Phone Context As Prefix CLI: add-ph-cntxt-as-pref **[AddPhoneContextAsPrefix]**	Determines whether the received Phone-Context parameter is added as a prefix to the outgoing ISDN Setup message with (for digital interfaces) Called and Calling numbers. ▪ **[0]** Disable (default) ▪ **[1]** Enable

67.11 SBC Parameters

The SBC and CRP parameters are described in the table below.

Table 67-73: SBC and CRP Parameters

Parameter	Description
CRP-specific Parameters	
Web: CRP Application EMS: Enable CPR Application CLI: enable-crp [EnableCRPApplication]	Enables the CRP application. ▪ [0] Disable (default) ▪ [1] Enable **Note:** For this parameter to take effect, a device reset is required.
Web: CRP Survivability Mode	Defines the CRP mode.

Parameter	Description
CLI: crp-survivability-mode [CRPSurvivabilityMode]	▪ [0] Standard Mode (default) ▪ [1] Always Emergency Mode ▪ [2] Auto-answer REGISTER
CLI: crp-gw-fallback [CRPGatewayFallback]	Enables fallback routing from the proxy server to the Gateway (PSTN). ▪ [0] = Disable (default) ▪ [1] = Enable
SBC-specific Parameters	
Web/EMS: Enable SBC CLI: enable-sbc **[EnableSBCApplication]**	Enables the Session Border Control (SBC) application. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ In addition to enabling this parameter, the number of maximum SBC/IP-to-IP sessions must be included in the Software License Key.
SBC and CRP Parameters	
WAN Interface Name [WanInterfaceName]	Defines the WAN interface for the VoIP interface. The available interface options depends on the hardware configuration (e.g., Ethernet or SHDSL) and/or whether VLANs are defined for the WAN interface. The value must be enclosed in single quotation marks ('...'), for example, WanInterfaceName = 'GigabitEthernet 0/0'. This WAN interface can be assigned to SIP signaling and/or media interfaces, in the SIP Interface table, where it is represented as "WAN" (see Configuring SIP Interfaces on page 267). If VLANs are configured, for example, for the Ethernet WAN interface, then you can select the WAN VLAN on which you want to run these SIP signaling and/or media interfaces. Therefore, for each outgoing SIP packet, the device sends it on the defined outgoing WAN interface; for each incoming SIP packet, the device identifies the packet according to the WAN interface from where it is received. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ This parameter is applicable only if the data-routing functionality is supported (i.e., relevant Software License Key is installed on the device).
Web: Unclassified Calls CLI: unclassified-calls **[AllowUnclassifiedCalls]**	Determines whether incoming calls that cannot be classified (i.e. classification process fails) to a Source IP Group are rejected or processed. ▪ **[0]** Reject = (Default) Call is rejected if classification fails. ▪ **[1]** Allow = If classification fails, the incoming packet is assigned to a source IP Group (and subsequently processed) as follows: ✓ The source SRD is determined according to the SIP Interface to where the SIP-initiating dialog request is sent. The source IP Group is set to the default IP Group associated with this SRD. ✓ If the source SRD is ID 0, then source IP Group ID 0 is

Parameter	Description
	chosen. In case of any other SRD, then the first IP Group associated with this SRD is chosen as the source IP Group or the call. If no IP Group is associated with this SRD, the call is rejected.
Web: SBC No Answer Timeout CLI: sbc-no-arelt-timeout **[SBCAlertTimeout]**	Defines the timeout (in seconds) for SBC outgoing (outbound IP routing) SIP INVITE messages. If the called IP party does not answer the call within this user-defined interval, the device disconnects the session. The device starts the timeout count upon receipt of a SIP 180 Ringing response from the called party. If no other SIP response (for example, 200 OK) is received thereafter within this timeout, the call is released. The valid range is 0 to 3600 seconds. the default is 600.
CLI: configure voip/sbc general-setting/num-of-subscribes **[NumOfSubscribes]**	Defines the maximum number of concurrent SIP SUBSCRIBE sessions permitted on the device. The valid value is any value between 0 and the maximum supported SUBSCRIBE sessions. When set to -1, the device uses the default value. For more information, contact your AudioCodes sales representative. **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ The maximum number of SUBSCRIBE sessions can be increased by reducing the maximum number of SBC channels in the Software License Key. For every reduced SBC session, the device gains two SUBSCRIBE sessions.
CLI: configure voip/sbc general-setting/sbc-dialog-subsc-route-mode **[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 ('Enable Proxy Keep Alive' parameter is set to any value other than **Disable**). ▪ Load-balancing between proxies is disabled ('Proxy Load Balancing Method' parameter is set to **Disable**).
CLI: sbc-max-fwd-limit **[SBCMaxForwardsLimit]**	Defines the Max-Forwards SIP header value. The Max-Forwards header is used to limit the number of servers (such as proxies) that can forward the SIP request. The Max-Forwards value indicates the remaining number of times this request message is allowed to be forwarded. This count is decremented by each server that forwards the request. This parameter affects the Max-Forwards header in the received message as follows: ▪ If the received header’s original value is 0, the message is not passed on and is rejected.

Parameter	Description
	▪ If the received header's original value is less than this parameter's value, the header's value is decremented before being sent on. ▪ If the received header's original value is greater than the parameter's value, the header's value is replaced by the user-defined parameter's value. The valid value range is 1-70. The default is 10.
Web: SBC Session-Expires CLI: sbc-sess-exp-time **[SBCSessionExpires]**	Defines the SBC session refresh timer (in seconds) in the Session-Expires header of outgoing INVITE messages. The valid value range is 90 (according to RFC 4028) to 86400. The default is 180.
Web: Minimum Session-Expires CLI: min-session-expires **[SBCMinSE]**	Defines the minimum amount of time (in seconds) between session refresh requests in a dialog before the session is considered timed out. This value is conveyed in the SIP Min-SE header. The valid range is 0 (default) to 1,000,000, where 0 means that the device does not limit Session-Expires.
Web: SBC Session Refreshing Policy CLI: configure voip/sbc general-setting/sbc-session-refresh-policy **[SBCSessionRefreshingPolicy]**	Defines the SIP user agent responsible for periodically sending refresh requests for established sessions (active calls). The session refresh allows SIP UAs or proxies to determine the status of the SIP session. When a session expires, the session is considered terminated by the UAs, regardless of whether a SIP BYE was sent by one of the UAs. The SIP Session-Expires header conveys the lifetime of the session, which is sent in re-INVITE or UPDATE requests (session refresh requests). The 'refresher=' parameter in the Session-Expires header (sent in the initial INVITE or subsequent 2xx response) indicates who sends the session refresh requests. If the parameter contains the value 'uac', the device performs the refreshes; if the parameter contains the value 'uas', the remote proxy performs the refreshes. An example of the Session-Expires header is shown below: `Session-Expires: 4000;refresher=uac` Thus, this parameter is useful when a UA does not support session refresh requests or does not support the indication of who performs session refresh requests. In such a scenario, the device can be configured to perform the session refresh requests. ▪ **[0]** Remote Refresher = (Default) The UA (proxy) performs the session refresh requests. The device indicates this to the UA by sending the SIP message with the 'refresher=' parameter in the Session-Expires header set to 'uas'. ▪ **[1]** SBC Refresher = The device performs the session refresh requests. The device indicates this to the UA by sending the SIP message with the 'refresher=' parameter in the Session-Expires header set to 'uac'. **Note:** The time values of the Session-Expires (session refresh interval) and Min-SE (minimum session refresh interval) headers can be configured using the SBCSessionExpires and SBCMinSE parameters, respectively.
Web: User Registration Grace Time CLI: configure voip/sbc general-setting/sbc-usr-reg-grace-time	Defines additional time (in seconds) to add to the registration expiry time of registered users in the device's Users Registration database.

Parameter	Description
[SBCUserRegistrationGraceTime]	The valid value is 0 to 300 (i.e., 5 minutes). The default is 0.
Web/EMS: Handle P-Asserted-Identity CLI: p-assert-id **[SBCAssertIdentity]**	Global parameter that defines the handling of the SIP P-Asserted-Identity header. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCAssertIdentity). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: Keep original user in Register **[SBCKeepContactUserinRegister]**	Determines whether the device replaces the Contact user with a unique Contact user in the outgoing message in response to a REGISTER request. ▪ **[0]** Disable = (Default) The device replaces the original Contact user with a unique Contact user, for example: ✓ Received Contact: <sip:123@domain.com> ✓ Outgoing (unique) Contact: <sip:FEU1_7_1@SBC> ▪ **[1]** Enable = The original Contact user is retained and used in the outgoing REGISTER request. **Note:** This parameter is applicable only to REGISTER messages received from User-type IP Groups and that are sent to Server-type IP Groups.
Web: SBC Remote Refer Behavior CLI: sbc-refer-bhvr **[SBCReferBehavior]**	Global parameter that defines the handling of SIP REFER requests. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCRemoteReferBehavior). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
CLI: sbc-xfer-prefix **[SBCXferPrefix]**	When the SBCReferBehavior is set to 1, the device, while interworking the SIP REFER message, adds the prefix "T~&R-" to the user part of the URI in the Refer-To header. After this, the device can receive an INVITE with such a prefix (the INVITE is sent by the UA that receives the REFER message or 302 response). If the device receives an INVITE with such a prefix, it replaces the prefix with the value defined for the SBCXferPrefix parameter. By default, no value is defined. **Note:** This feature is also applicable to 3xx redirect responses. The device adds the prefix "T~&R-" to the URI user part in the Contact header if the SBC3xxBehavior parameter is set to 1.
CLI: sbc-3xx-bhvt **[SBC3xxBehavior]**	Global parameter that defines the handling of SIP 3xx redirect responses. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCRemote3xxBehavior). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the

Parameter	Description
	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 this parameter is enabled, the endpoint includes the previously negotiated media type (i.e., audio) with the new negotiated media type (i.e., image) in its SDP offer:
	`v=0`
	`o=bob 2890844730 2890844731 IN IP4 host.example.com`
	`s=`
	`c=IN IP4 host.example.com`
	`t=0 0`
	`m=audio 0 RTP/AVP 0`
	`m=image 12345 udptl t38`
	If this parameter is disabled, the only ‘m=’ line included in the SDP is the newly negotiated media (i.e., image).

Web: SBC Diversion URI Type CLI: sbc-diversion-uri-type (configure voip > sbc general-setting) **[SBCDiversionUriType]**	Defines the URI type to use in the SIP Diversion header of the outgoing SIP message. ▪ **[0]** Transparent = (Default) The device does not change the URI and leaves it as is. ▪ **[1]** Sip = The "sip" URI is used. ▪ **[2]** Tel = The "tel" URI is used. **Note:** The parameter is applicable only if the Diversion header is used. The SBCDiversionMode and SBCHistoryInfoMode parameters in the IP Profile table determine the call redirection (diversion) SIP header to use - History-Info or Diversion.
Web: SBC Server Auth Mode CLI: sbc-server-auth-mode **[SBCServerAuthMode]**	Defines whether authentication of the SIP client is done locally (by the device) or by a RADIUS server. ▪ **[0]** (default) = Authentication is done by the device (locally). ▪ **[1]** = Authentication is done by the RFC 5090 compliant RADIUS server ▪ **[2]** = Authentication is done according to the Draft Sterman-aaa-sip-01 method. **Note:** Currently, option [1] is not supported.
Web: Lifetime of the nonce in seconds CLI: lifetime-of-nonce **[AuthNonceDuration]**	Defines the lifetime (in seconds) that the current nonce is valid for server-based authentication. The device challenges a message that attempts to use a server nonce beyond this period. This parameter is used to provide replay protection (i.e., ensures that old communication streams are not used in replay attacks). The valid value range is 30 to 600. The default is 300.
Web: Authentication Challenge Method CLI: auth-chlng-mthd **[AuthChallengeMethod]**	Defines the type of server-based authentication challenge. ▪ **[0]** 0 = (Default) Send SIP 401 "Unauthorized" with a WWW-Authenticate header as the authentication challenge response. ▪ **[1]** 1 = Send SIP 407 "Proxy Authentication Required" with a

Parameter	Description
	Proxy-Authenticate header as the authentication challenge response.
Web: Authentication Quality of Protection CLI: auth-qop **[AuthQOP]**	Defines the authentication and integrity level of quality of protection (QoP) for digest authentication offered to the client. When the device challenges a SIP request (e.g., INVITE), it sends a SIP 401 response with the Proxy-Authenticate header or WWW-Authenticate header containing the 'qop' parameter. The QoP offered in the 401 response can be 'auth', 'auth-int', both 'auth' and 'auth-int', or the 'qop' parameter can be omitted from the 401 response. In response to the 401, the client needs to send the device another INVITE with the MD5 hash of the INVITE message and indicate the selected auth type. ▪ **[0]** 0 = The device sends 'qop=auth' in the SIP response, requesting authentication (i.e., validates user by checking user name and password). This option does not authenticate the message body (i.e., SDP). ▪ **[1]** 1 = The device sends 'qop=auth-int' in the SIP response, indicating required authentication and authentication with integrity (e.g., checksum). This option restricts the client to authenticating the entire SIP message, including the body, if present. ▪ **[2]** 2 = (Default) The device sends 'qop=auth, auth-int' in the SIP response, indicating either authentication or integrity. This enables the client to choose 'auth' or 'auth-int'. If the client chooses 'auth-int', then the body is included in the authentication. If the client chooses 'auth', then the body is not authenticated. ▪ **[3]** 3 = No 'qop' parameter is offered in the SIP 401 challenge message.
Web: SBC User Registration Time CLI: sbc-usr-rgstr-time **[SBCUserRegistrationTime]**	Global parameter that defines the duration (in seconds) of the periodic registrations that occur between the user and the device (the device responds with this value to the user). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCUserRegistrationTime). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: SBC Proxy Registration Time CLI: sbc-prxy-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.
CLI: config-voip>sbc general-setting sbc-rand-expire **[SBCRandomizeExpires]**	Defines a value (in seconds) that is used to calculate a new value for the expiry time in the Expires header of SIP 200 OK responses for user registration and subscription requests from users. The expiry time value appears in the Expires header in REGISTER and SUBSCRIBE SIP messages. When the device receives such a request from a user, it forwards it to the proxy or registrar server. Upon a successful registration or subscription,

Parameter	Description
	the server sends a SIP 200 OK response. If the expiry time was unchanged by the server, the device applies this feature and changes the expiry time in the SIP 200 OK response before forwarding it to the user; otherwise, the device does not change the expiry time. This feature is useful in scenarios where multiple users may refresh their registration or subscription simultaneously, thereby causing the device to handle many such sessions at a given time. This may result in an overload of the device (reaching maximum session capacity), thereby preventing the establishment of new calls or preventing the handling of some user registration or subscription requests. When this feature is enabled, the device assigns a random expiry time to each user registration or subscription and thus, ensuring future user registration and subscription requests are more distributed over time (i.e., do not all occur simultaneously). The device takes any random number between 0 and the value configured by this parameter, and then subtracts this random number from the original expiry time value. For example, assume that the original expiry time is 120 and this parameter is set to 10. If the device randomly chooses the number 5 (i.e., between 0 and 10), the resultant expiry time will be 115 (120 minus 5). The valid value is 0 to 20. The default is 10. If set to 0, the device does not change the expiry time. **Notes:** ▪ The lowest expiry time that the device sends in the 200 OK, regardless of the resultant calculation, is 10 seconds. For example, if the original expiry time is 12 seconds and this parameter is set to 5, theoretically, the new expiry time can be less than 10 (e.g., 12 – 4 = 8). However, the expiry time will be set to 10. ▪ The expiry time received from the user can be changed by the device before forwarding it to the proxy. This is configured by the SBCUserRegistrationTime parameter.
Web: SBC Survivability Registration Time CLI: sbc-surv-rgstr-time **[SBCSurvivabilityRegistrationTime]**	Defines the duration of the periodic registrations between the user and the device, when the device is in survivability state (i.e., when REGISTER requests cannot be forwarded to the proxy and are terminated by the device). When set to 0, the device uses the value set by the SBCUserRegistrationTime parameter for the device's response. The valid range is 0 to 2,000,000 seconds. The default is 0.
[SBCEnableAASTRASurvivabilityNotice]	Enables the device to notify Aastra IP phones that the device is currently operating in Survivability mode. When this occurs, the Aastra IP phones display the message, "Stand Alone Mode" on their LCD screens. Survivability mode occurs when connectivity with the WAN fails and as a result, the device enables communication between IP phone users within the LAN enterprise. ▪ **[0]** = Disable ▪ **[1]** = Enable When this feature is enabled and the SBC device is in Survivability mode, it responds to SIP REGISTER messages from the IP phones with a SIP 200 OK containing the following XML

Parameter	Description

	body:
	`Content-Type: application/xml`
	`<?xml version="1.0" encoding="utf-8"?>`
	`<LMIDocument version="1.0">`
	`<LocalModeStatus>`
	`   <LocalModeActive>true</LocalModeActive>`
	`   <LocalModeDisplay>StandAlone Mode</LocalModeDisplay>`
	`</LocalModeStatus>`
	`</LMIDocument>`

Web: SBC Dialog-Info Interworking CLI: configure voip/sbc general-setting/sbc-dialog-info-interwork **[EnableSBCDialogInfoInterworking]**	Enables the interworking of dialog information (parsing of call identifiers in XML body) in SIP NOTIFY messages received from a remote application server. ▪ **[0]** Disable (default) ▪ **[1]** Enable For more information, see "Interworking Dialog Information in SIP NOTIFY Messages" on page 535.
CLI: sbc-keep-call-id **[SBCKeepOriginalCallId]**	Enables the device to use the same call identification value received in incoming messages for the call identification in outgoing messages. The call identification value is contained in the SIP Call-ID header. ▪ **[0]** = (Default) Disable - the device creates a new Call-ID value for the outgoing message. ▪ **[1]** = Enable - the device uses the received Call-ID value of the incoming message in the outgoing message. **Note:** When the device sends an INVITE as a result of a REFER/3xx termination, the device always creates a new Call-ID value and ignores this parameter's settings.
Web: SBC GRUU Mode CLI: sbc-gruu-mode **[SBCGruuMode]**	Determines the Globally Routable User Agent (UA) URI (GRUU) support, according to RFC 5627. ▪ **[0]** None = No GRUU is supplied to users. ▪ **[1]** As Proxy = (Default) The device provides same GRUU types as the proxy provided the device's GRUU clients. ▪ **[2]** Temporary only = Supply only temporary GRUU to users. (Currently not supported.) ▪ **[3]** Public only = The device provides only public GRUU to users. ▪ **[4]** Both = The device provides temporary and public GRUU to users. (Currently not supported.) This parameter allows the device to act as a GRUU server for its SIP UA clients, providing them with public GRUU's, according to RFC 5627. The public GRUU provided to the client is denoted in the SIP Contact header parameters, "pub-gruu". Public GRUU remains the same over registration expirations. On the other SBC leg communicating with the Proxy/Registrar, the device acts as a GRUU client. The device creates a GRUU value for each of its registered clients, which is mapped to the GRUU value received from the Proxy server. In other words, the created GRUU value is only used between the device and its clients (endpoints).

Parameter	Description
	`Public-GRUU: sip:userA@domain.com;gr=unique-id`
Web: Bye Authentication CLI: sbc-bye-auth **[SBCEnableByeAuthentication]**	Enables authenticating a SIP BYE request before disconnecting the call. This feature prevents, for example, a scenario in which the SBC SIP client receives a BYE request from a third-party imposer assuming the identity of a participant in the call and as a consequence, the call between the first and second parties is inappropriately disconnected. ▪ **[0]** Disable (default) ▪ **[1]** Enable = The device forwards the SIP authentication response (for the BYE request) to the request sender and waits for the user to authenticate it. The call is disconnected only if the authenticating server responds with a 200 OK.
Web: SBC Enable Subscribe Trying CLI: configure voip > sbc general-setting > set sbc-subs-try **[SBCSendTryingToSubscribe]**	Enables the device to send SIP 100 Trying responses upon receipt of SUBSCRIBE or NOTIFY messages. ▪ **[0]** Disable (Default) ▪ **[1]** Enable
[SBCExtensionsProvisioningMode]	Enables SBC user registration for interoperability with BroadSoft's BroadWorks server, to provide call survivability in case of connectivity failure with the BroadWorks server. ▪ **[0]** = (Default) Normal processing of REGISTER messages. ▪ **[1]** = Registration method for BroadWorks server. In a failure scenario with BroadWorks, the device acts as a backup SIP proxy server, maintaining call continuity between the enterprise LAN users (subscribers) and between the subscribers and the PSTN (if provided). **Note:** For a detailed description of this feature, see "Auto-Provisioning of Subscriber-Specific Information for BroadWorks Server" on page 526.
Web: SBC Direct Media CLI: sbc-direct-media **[SBCDirectMedia]**	Enables the No Media Anchoring feature (i.e., direct media) for all SBC calls, whereby SIP signaling is handled by the device without handling the RTP/SRTP (media) flow between the user agents (UA). The RTP packets do not traverse the device. Instead, the two SIP UAs establish a direct RTP/SRTP flow between one another. Signaling continues to traverse the device with minimal intermediation and involvement to enable certain SBC abilities such as routing ▪ **[0]** Disable = (Default) All calls traverse the device (i.e., no direct media). If No Media Anchoring is enabled for an SRD (in the SRD table), then calls between endpoints belonging to that SRD use No Media Anchoring. ▪ **[1]** Enable = All SBC calls use the No Media Anchoring feature (i.e., direct media). **Note:** For more information on No Media Anchoring, see "No Media Anchoring (Anti Tromboning)" on page 511.
Web: Transcoding Mode CLI: transcoding-mode [TranscodingMode]	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 of this parameter and for configuring this functionality in the IP Profile table, see Configuring IP Profiles on page 314. **Note:** If this functionality is configured for a specific IP Profile, the

Parameter	Description
	settings of this global parameter is ignored for calls associated with the IP Profile.
Web: SBC 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:** This parameter is applicable only if an Extension Coders Group is assigned to the IP Profile (IP Profile table's parameter, SBCExtensionCodersGroupID).
SBC RTCP Mode CLI: sbc-rtcp-mode **[SBCRTCPMode]**	Global parameter that defines the handling of RTCP packets. You can also configure this functionality per specific calls, using IP Profiles (IPProfile_SBCRTCPMode). For a detailed description of this parameter and for configuring this functionality in the IP Profile table, see "Configuring IP Profiles" on page 314. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.
Web: SBC Send Invite To All Contacts CLI: sbc-send-invite-to-all-contacts **[SBCSendInviteToAllContacts]**	Enables call forking of INVITE message received with a Request-URI of a specific contact registered in the device's database, to all users under the same AOR as the contact. ▪ **[0]** Disable (default) = Sends the INVITE only to the contact of the received Request-URI. ▪ **[1]** Enable To configure call forking initiated by the device, see "Initiating SIP Call Forking" on page 531.
Web: SBC Shared Line Registration Mode CLI: sbc-shared-line-reg-mode **[SBCSharedLineRegMode]**	Enables the termination on the device of SIP REGISTER messages from secondary lines 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.
Web: SBC Forking Handling Mode CLI: 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. ▪ **[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

Parameter	Description
	discarded.
Web: Gateway Direct Route Prefix CLI: configure voip/sbc general-setting/gw-direct-route-prefix [GWDirectRoutePrefix]	Defines the prefix destination Request-URI user part that is appended to the original user part for alternative IP-to-IP call routing from SBC to Gateway (Tel) interfaces. The valid value is a string of up to 16 characters. The default is "acgateway-<original prefix destination number>". For example, "acgateway-200". For more information, see Configuring SBC IP-to-IP Routing Rules on page 552.
CLI: sbc-media-sync **[EnableSBCMediaSync]**	Enables SBC media synchronization process for calls established from SIP forking that is initiated by external proxy servers. It is possible that a call is established with the media not synchronized between the SBC legs. Media synchronization resolves this issue. ▪ **[0]** Disable ▪ **[1]** Enable (default)
Web/EMS: SBC Fax Detection Timeout CLI: 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).
Admission Control Table	
Web: Admission Control EMS: Call Admission Control CLI: configure voip > sbc sbc-admission-control **[SBCAdmissionControl]**	This table parameter defines Call Admission Control (CAC) rules for limiting the number of allowed concurrent calls (SIP dialogs). The format of the ini file table parameter is as follows: [SBCAdmissionControl] FORMAT SBCAdmissionControl_Index = SBCAdmissionControl_AdmissionControlName, SBCAdmissionControl_LimitType, SBCAdmissionControl_IPGroupID, SBCAdmissionControl_SRDID, SBCAdmissionControl_RequestType, SBCAdmissionControl_RequestDirection, SBCAdmissionControl_Limit, SBCAdmissionControl_LimitPerUser, SBCAdmissionControl_Rate, SBCAdmissionControl_MaxBurst, SBCAdmissionControl_Reservation; [\SBCAdmissionControl] For a detailed description of this table, see "Configuring Admission Control" on page 539.
Allowed Audio Coders Table	
Web: Allowed Audio Coders CLI: configure voip > sbc allowed-coders-group AllowedCodersGroup0 **[AllowedCodersGroupX]**	This table parameter defines Allowed Coders Groups, which determine the audio (voice) coders that can be used for a specific SIP entity. The format of the ini file table parameter is as follows:

Parameter	Description
	[AllowedCodersGroupX] FORMAT AllowedCodersGroup_Index = AllowedCodersGroup_Name; [\AllowedCodersGroup] Where *X* represents the index number. For a detailed description of this table, see "Configuring Allowed Audio Coder Groups" on page 543.
Allowed Video Coders Table	
CLI: configure voip/sbc allowed-video-coders-group group-X **[AllowedVideoCodersGroupX]**	This table parameter defines Allowed Video Coders Groups, which determine the video coders that can be used for a specific SIP entity. The format of the ini file table parameter is as follows: [AllowedVideoCodersGroup0] FORMAT AllowedVideoCodersGroup_Index = AllowedVideoCodersGroup_Name; [\AllowedVideoCodersGroup] Where *X* represents the index number. For a detailed description of this table, see "Configuring Allowed Video Coder Groups" on page 544.
Classification Table	
Web: Classification Table EMS: SBC Classification CLI: configure voip > sbc routing classification **[Classification]**	This table parameter configures the Classification table. This table classifies incoming SIP dialogs to Source IP Groups. The format of the ini file table parameter is as follows: [Classification] FORMAT Classification_Index = Classification_ClassificationName, Classification_MessageCondition, Classification_SrcSRDID, Classification_SrcAddress, Classification_SrcPort, Classification_SrcTransportType, Classification_SrcUsernamePrefix, Classification_SrcHost, Classification_DestUsernamePrefix, Classification_DestHost, Classification_ActionType, Classification_SrcIPGroupID; [\Classification] For a detailed description of this table, see "Configuring Classification Rules" on page 545.
Condition Table	
Web: Condition Table CLI: configure voip > sbc routing condition-table **[ConditionTable]**	This table parameter configures Message Condition rules for SIP messages. [ConditionTable] FORMAT ConditionTable_Index = ConditionTable_Condition, ConditionTable_Description; [\ConditionTable] For a detailed description of this table, see "Configuring Message Condition Rules" on page 551.
SBC IP-to-IP Routing Table	
Web: IP-to-IP Routing Table EMS: IP to IP Routing CLI: configure voip > sbc routing ip2ip-routing	This table parameter configures the SBC IP-to-IP Routing table for routing incoming SIP messages such as INVITE messages to an IP destination. The format of the ini file table parameter is as follows:

Parameter	Description
[IP2IPRouting]	[IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_RouteName, IP2IPRouting_SrcIPGroupID, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_RequestType, IP2IPRouting_MessageCondition, IP2IPRouting_ReRouteIPGroupID, IP2IPRouting_Trigger, IP2IPRouting_CallSetupRulesSetId, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupID, IP2IPRouting_DestSRDID, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions, IP2IPRouting_GroupPolicy, IP2IPRouting_CostGroup; [\IP2IPRouting] For a detailed description of this table, see "Configuring SBC IP-to-IP Routing Rules" on page 552.
SBC Alternative Routing Reasons Table	
Web: SBC Alternative Routing Reasons EMS: Alternative Routing Reasons CLI: configure voip > sbc routing sbc-alternative-routing-reasons **[SBCAlternativeRoutingReasons]**	This table parameter configures the SBC Alternative Routing Reasons table. The format of the ini file table parameter is as follows: [SBCAlternativeRoutingReasons] FORMAT SBCAlternativeRoutingReasons_Index = SBCAlternativeRoutingReasons_ReleaseCause; [\SBCAlternativeRoutingReasons] For a detailed description of this table, see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 561.
IP to IP Inbound Manipulation Table	
Web: IP to IP Inbound Manipulation EMS: IP to IP Inbound Manipulation CLI: configure voip > sbc manipulations ip-inbound-manipulation **[IPInboundManipulation]**	This table parameter configures the IP to IP Inbound Manipulation table. This table allows you to manipulate the SIP URI user part (source and/or destination) of the inbound SIP dialog message. The format of the ini file table parameter is as follows: [IPInboundManipulation] FORMAT IPInboundManipulation_Index = IPInboundManipulation_ManipulationName IPInboundManipulation_IsAdditionalManipulation, IPInboundManipulation_ManipulatedURI, IPInboundManipulation_ManipulationPurpose, IPInboundManipulation_SrcIPGroupID, IPInboundManipulation_SrcUsernamePrefix, IPInboundManipulation_SrcHost, IPInboundManipulation_DestUsernamePrefix, IPInboundManipulation_DestHost, IPInboundManipulation_RequestType, IPInboundManipulation_RemoveFromLeft, IPInboundManipulation_RemoveFromRight, IPInboundManipulation_LeaveFromRight, IPInboundManipulation_Prefix2Add, IPInboundManipulation_Suffix2Add; [\IPInboundManipulation] For a detailed description of this table, see "Configuring IP-to-IP Inbound Manipulations" on page 565.
IP to IP Outbound Manipulation Table	

Parameter	Description
Web: IP to IP Outbound Manipulation EMS: IP to IP Outbound Manipulation CLI: configure voip > sbc manipulations ip-outbound-manipulation **[IPOutboundManipulation]**	This table parameter configures the IP to IP Outbound Manipulation table. This table allows you to manipulate the SIP URI user part (source and/or destination) of the outbound SIP dialog message. The format of the ini file table parameter is as follows: FORMAT IPOutboundManipulation_Index = IPOutboundManipulation_ManipulationName, IPOutboundManipulation_IsAdditionalManipulation, IPOutboundManipulation_SrcIPGroupID, IPOutboundManipulation_DestIPGroupID, IPOutboundManipulation_SrcUsernamePrefix, IPOutboundManipulation_SrcHost, IPOutboundManipulation_DestUsernamePrefix, IPOutboundManipulation_DestHost, IPOutboundManipulation_RequestType, IPOutboundManipulation_ReRouteIPGroupID, IPOutboundManipulation_Trigger, IPOutboundManipulation_ManipulatedURI, IPOutboundManipulation_RemoveFromLeft, IPOutboundManipulation_RemoveFromRight, IPOutboundManipulation_LeaveFromRight, IPOutboundManipulation_Prefix2Add, IPOutboundManipulation_Suffix2Add, IPOutboundManipulation_PrivacyRestrictionMode; For a detailed description of this table, see "Configuring IP-to-IP Outbound Manipulations" on page 568.

67.12 Standalone Survivability Parameters

The Stand-alone Survivability (SAS) parameters are described in the table below. For a detailed description of SAS, refer to the *SAS Configuration Guide*.

Table 67-74: SAS Parameters

Parameter	Description
Web: Enable SAS EMS: Enable CLI: enable-sas **[EnableSAS]**	Enables the Stand-Alone Survivability (SAS) feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable When enabled, the device receives the registration requests from different SIP entities in the local network and then forwards them to the defined proxy. If the connection to the proxy fails ('Emergency Mode'), the device serves as a proxy by allowing calls internal to the local network or outgoing to PSTN. **Note:** For this parameter to take effect, a device reset is required.
Web: SAS Local SIP UDP Port EMS: Local SIP UDP CLI: sas-local-sip-udp-port **[SASLocalSIPUDPPort]**	Defines the local UDP port for sending and receiving SIP messages for SAS. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default is 5080.
Web: SAS Default Gateway IP	Defines the Default Gateway used in SAS 'Emergency Mode'.

Parameter	Description
EMS: Default Gateway IP CLI: sas-default-gw-ip **[SASDefaultGatewayIP]**	When an incoming SIP INVITE is received and the destination Address-Of-Record is not included in the SAS database, the request is immediately sent to this default gateway. The address can be configured as an IP address (dotted-decimal notation) or as a domain name (up to 49 characters). You can also configure the IP address with a destination port, e.g., "10.1.2.3:5060". The default is a null string, i.e., the local IP address of the gateway.
Web: SAS Registration Time EMS: Registration Time CLI: sas-registration-time **[SASRegistrationTime]**	Defines the value of the SIP Expires header that is sent in a 200 OK response to an incoming REGISTER message when in SAS 'Emergency Mode'. The valid range is 0 (Analog) or 10 (Digital) to 2,000,000. The default is 20.
Web: SAS Local SIP TCP Port EMS: Local SIP TCP Port CLI: sas-local-sip-tcp-port **[SASLocalSIPTCPPort]**	Defines the local TCP port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default is 5080.
Web: SAS Local SIP TLS Port EMS: Local SIP TLS Port CLI: sas-local-sip-tls-port **[SASLocalSIPTLSPort]**	Defines the local TLS port used to send/receive SIP messages for the SAS application. The SIP entities in the local network need to send the registration requests to this port. When forwarding the requests to the proxy ('Normal Mode'), this port serves as the source port. The valid range is 1 to 65,534. The default is 5081.
Web: SAS Connection Reuse CLI: sas-connection-reuse **[SASConnectionReuse]**	Enables the re-use of the same TCP connection for sessions with the same user in the SAS application. ▪ **[0]** Disable ▪ **[1]** Enable (default) The device can use the same TCP connection for multiple SIP requests / responses for a specific SIP UA. For example, assume the following: ▪ User A sends a REGISTER message to SAS with transport=TCP. ▪ User B sends an INVITE message to A using SAS. In this scenario, the SAS application forwards the INVITE request using the TCP connection that User A initially opened with the REGISTER message.
Web/EMS: Enable Record-Route CLI: record-route **[SASEnableRecordRoute]**	Determines whether the device's SAS application adds the SIP Record-Route header to SIP requests. This ensures that SIP messages traverse the device's SAS agent by including the SAS IP address in the Record-Route header. ▪ **[0]** Disable (default) ▪ **[1]** Enable The Record-Route header is inserted in a request by a SAS proxy to force future requests in the dialog session to be routed through the SAS agent. Each traversed proxy in the path can insert this header, causing all future dialogs in the session to pass through it as well. When this feature is enabled, the SIP Record-Route header includes the URI "lr" parameter, indicating loose routing, for

Parameter	Description
	example: `Record-Route: <sip:server10.biloxi.com;lr>`
Web: SAS Proxy Set EMS: Proxy Set CLI: sas-proxy-set **[SASProxySet]**	Defines the Proxy Set (index number) used in SAS Normal mode to forward REGISTER and INVITE requests from users that are served by the SAS application. The valid range is 0 to 5. The default is 0 (i.e., default Proxy Set).
Web: Redundant SAS Proxy Set EMS: Redundant Proxy Set CLI: rdcy-sas-proxy-set **[RedundantSASProxySet]**	Defines the Proxy Set (index number) used in SAS Emergency mode for fallback when the user is not found in the Registered Users database. Each time a new SIP request arrives, the SAS application checks whether the user is listed in the registration database. If the user is located in the database, the request is sent to the user. If the user is not found, the request is forwarded to the next redundant SAS defined in the Redundant SAS Proxy Set. If that SAS Proxy IP appears in the Via header of the request, it is not forwarded (thereby, preventing loops in the request's course). If no such redundant SAS exists, the SAS sends the request to its default gateway (configured by the parameter SASDefaultGatewayIP). The valid range is -1 to 5. The default is -1 (i.e., no redundant Proxy Set).
Web/EMS: SAS Block Unregistered Users CLI: sas-block-unreg-usrs **[SASBlockUnRegUsers]**	Determines whether the device rejects SIP INVITE requests received from unregistered SAS users. This applies to SAS Normal and Emergency modes. ▪ **[0]** Un-Block = (Default) Allow INVITE from unregistered SAS users. ▪ **[1]** Block = Reject dialog-establishment requests from un-registered SAS users.
CLI: sas-contact-replace **[SASEnableContactReplace]**	Enables the device to change the SIP Contact header so that it points to the SAS host and therefore, the top-most SIP Via header and the Contact header point to the same host. ▪ **[0]** (default) = Disable - when relaying requests, the SAS agent adds a new Via header (with the SAS IP address) as the top-most Via header and retains the original Contact header. Thus, the top-most Via header and the Contact header point to different hosts. ▪ **[1]** = Enable - the device changes the Contact header so that it points to the SAS host and therefore, the top-most Via header and the Contact header point to the same host. **Note:** Operating in this mode causes all incoming dialog requests to traverse the SAS, which may cause load problems.
Web: SAS Survivability Mode EMS: Survivability Mode CLI: sas-survivability **[SASSurvivabilityMode]**	Determines the Survivability mode used by the SAS application. ▪ **[0]** Standard = (Default) Incoming INVITE and REGISTER requests are forwarded to the defined Proxy list of SASProxySet in Normal mode and handled by the SAS application in Emergency mode. ▪ **[1]** Always Emergency = The SAS application does not use Keep-Alive messages towards the SASProxySet, instead it always operates in Emergency mode (as if no Proxy in the SASProxySet is available). ▪ **[2]** Ignore Register = Use regular SAS Normal/Emergency

Parameter	Description
	logic (same as option [0]), but when in Normal mode incoming REGISTER requests are ignored. - **[3]** Auto-answer REGISTER = When in Normal mode, the device responds to received REGISTER requests by sending a SIP 200 OK (instead of relaying the registration requests to a Proxy), and enters the registrations in its SAS database. - **[4]** Use Routing Table only in Normal mode = The device uses the IP-to-IP Routing table to route IP-to-IP SAS calls only when in SAS Normal mode (and is unavailable when SAS is in Emergency mode). This allows routing of SAS IP-to-IP calls to different destinations (and not only to the SAS Proxy Set).
Web: SAS Subscribe Response CLI: sas-subscribe-resp **[SASSubscribeResponse]**	Defines the SIP response upon receipt of a SUBSCRIBE message when SAS is in Emergency mode. For example, if this parameter is set to "200", then SAS sends a SIP 200 OK in response to a SUBSCRIBE message, when in Emergency mode. The valid value is 200 to 699. The default is 489.
Web: Enable ENUM CLI: enable-enum **[SASEnableENUM]**	Enables SAS to perform ENUM (E.164 number to URI mapping) queries when receiving INVITE messages in SAS emergency mode. - **[0]** Disable (default) - **[1]** Enable
Web: SAS Binding Mode EMS: Binding Mode CLI: sasbindingmode **[SASBindingMode]**	Determines the SAS application database binding mode. - **[0]** URI = (Default) If the incoming AoR in the INVITE requests is using a 'tel:' URI or 'user=phone' is defined, the binding is performed according to the user part of the URI only. Otherwise, the binding is according to the entire URI, i.e., User@Host. - **[1]** User Part only = The binding is always performed according to the User Part only.
Web: SAS Emergency Numbers CLI: sas-emerg-nb **[SASEmergencyNumbers]**	Defines emergency numbers for the device's SAS application. When the device's SAS agent receives a SIP INVITE (from an IP phone) that includes one of the emergency numbers (in the SIP user part), it forwards the INVITE to the default gateway (configured by the parameter SASDefaultGatewayIP), i.e., the device itself, which sends the call directly to the PSTN. This is important for routing emergency numbers such as 911 (in North America) directly to the PSTN. This is applicable to SAS operating in Normal and Emergency modes. Up to four emergency numbers can be defined, where each number can be up to four digits.
CLI: sas-emerg-prefix **[SASEmergencyPrefix]**	Defines a prefix that is added to the Request-URI user part of the INVITE message that is sent by the device's SAS agent when in Emergency mode to the default gateway or to any other destination (using the IP-to-IP Routing table). This parameter is required to differentiate between normal SAS calls routed to the default gateway and emergency SAS calls. Therefore, this allows you to define different manipulation rules for normal and emergency calls. This valid value is a character string. By default, no value is defined.

Parameter	Description
Web: SAS Entering Emergency Mode CLI: sas-enter-emg-mode **[SASEnteringEmergencyMode]**	Determines for which sent SIP message types the device enters SAS Emergency mode if no response is received for them from the proxy server. ▪ **[0]** = (Default) SAS enters Emergency mode only if no response is received from sent SIP OPTIONS messages. ▪ **[1]** = SAS enters Emergency mode if no response is received from sent SIP OPTIONS, INVITE, or REGISTER messages. **Note:** If the keep-alive mechanism is disabled for the Proxy Set (in the Proxy Set table) and this parameter is set to [1], SAS enters Emergency mode only if no response is received from sent INVITE or REGISTER messages.
CLI: sas-indialog-mode **[SASInDialogRequestMode]**	Defines how the device sends incoming SIP dialog requests received from users when not in SAS Emergency mode. ▪ **[0]** = (Default) Send according to the SIP Request-URI. ▪ **[1]** = Send to Proxy server.
Web: SAS Inbound Manipulation Mode CLI: sas-inb-manipul-md **[SASInboundManipulationMode]**	Enables destination number manipulation of incoming INVITE messages when SAS is in Emergency mode. The manipulation rule is done in the IP to IP Inbound Manipulation table. ▪ **[0]** None (default) ▪ **[1]** Emergency Only **Note:** Inbound manipulation applies only to INVITE requests.
SAS Registration Manipulation Table	
Web: SAS Registration Manipulation EMS: Stand-Alone Survivability CLI: configure voip > sas sasregistrationmanipulation **[SASRegistrationManipulation]**	This table parameter configures the SAS Registration Manipulation table. This table is used by the SAS application to manipulate the SIP Request-URI user part of incoming INVITE messages and of incoming REGISTER request AoR (To header), before saving it to the registered users database. The format of this table parameter is as follows: [SASRegistrationManipulation] FORMAT SASRegistrationManipulation_Index = SASRegistrationManipulation_RemoveFromRight, SASRegistrationManipulation_LeaveFromRight; [\SASRegistrationManipulation] For example, the manipulation rule below routes an INVITE with Request-URI header "sip:7184002@10.33.4.226" to user "4002@10.33.4.226" (i.e., keep only four digits from right of user part): `SASRegistrationManipulation 0 = 0, 4;`
Web: SAS IP-to-IP Routing Table	
[IP2IPRouting]	This table parameter configures the IP-to-IP Routing table for SAS routing rules. The format of the ini file table parameter is as follows: [IP2IPRouting] FORMAT IP2IPRouting_Index = IP2IPRouting_SrcIPGroupID, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupID, IP2IPRouting_DestSRDID, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions;

Parameter	Description
	[\IP2IPRouting] For example: IP2IPRouting 1 = -1, *, *, *, *, 0, -1, -1, , 0, -1, 0;

67.13 IP Media Parameters

The IP media parameters are described in the table below.

Table 67-75: IP Media Parameters

Parameter	Description
Web: Number of Media Channels EMS: Media Channels CLI: media-channels [MediaChannels]	Defines the maximum number of DSP channels allocated for various functionalities such as transcoding, IP conferencing, The default is 0. **Notes:** ▪ For this 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. ▪ Other DSP channels can be used for PSTN interfaces.
Web: NetAnn Announcement ID [NetAnnAnncID]	Defines the NetAnn identification string (up to 16 characters) for playing an announcement using the NetAnn interface. The application server sends a regular SIP INVITE message with a SIP URI that includes this identifier string and a "play=" parameter that identifies the necessary announcement. The default is 'annc'. Example 1: INVITE sip: annc@10.2.3.4;play=http://localhost/1. Example 2: INVITE sip: annc@10.2.3.4;play=http://10.2.3.4/Annc/ello.wav.
Web: Transcoding ID [TranscodingID]	Defines the Transcoding identification string (up to 16 characters) used for identifying an incoming Transcoding call. The default is 'trans'. For more information on Transcoding, see NetAnn Interface.
Conferencing Parameters	
Web/EMS: Conference ID [ConferenceID]	Defines the Conference Identification string. The valid value is a string of up to 16 characters. The default is "conf". **Note:** To join a conference, the INVITE URI must include the Conference ID string preceded by the number of the participants in the conference and terminated by a unique number. For example: INVITE sip:4conf1234@10.1.10.10 INVITE messages with the same URI join the same conference.
Web: Beep on Conference CLI: beep-on-conf	Enables playing a beep when a participant joins or leaves a conference (in the latter case, a beep of a different pitch is

Parameter	Description
[BipOnConference]	heard). - [0] Disable - [1] Enable (default)
Web: Enable Conference DTMF Clamping CLI: conf-dtmf-clamping [EnableConferenceDTMFClamp]	Determines the device logic once a DTMF is received on any conference participant. If enabled, the DTMF is not regenerated toward the other conference participants. This logic is only relevant for simple conferencing (NetAnn). - [0] Disable - [1] Enable (default)
Web: Enable Conference DTMF Reporting CLI: conf-dtmf-reporting [EnableConferenceDTMFReporting]	Determines the device logic once a DTMF is received on any conference participant. If enabled, the device reports this DTMF in an out-of-band SIP message (according to TxDTMFOptions). This logic is only relevant for simple conferencing (NetAnn). - [0] Disable (default) - [1] Enable
Web: Active Speakers Min. Interval CLI: active-speaker-notification-minimum-interval [ActiveSpeakersNotificationMinInterval]	Defines the minimum interval (in 100 msec units) between each Active Speaker Notification (ASN) events report. These events report on the active speakers in a conference. The event is issued whenever the active speakers change. Minimum configurable interval between events is 500 msec (5 units). The range is 5 to 2147483647 units. The default is 20 (i.e., 100 msec).
Automatic Gain Control (AGC) Parameters	
Web: Enable AGC EMS: AGC Enable CLI: 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 **Notes:** - This parameter can also be configured for a Tel Profile (in the Tel Profile table). - For a description of AGC, see Automatic Gain Control (AGC) on page 195.
Web: AGC Slope EMS: Gain Slope CLI: 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

Parameter	Description
	- [13] 13 = 5.00 dB/sec - [14] 14 = 5.50 dB/sec - [15] 15 = 6.00 dB/sec - [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
Web: AGC Redirection EMS: Redirection CLI: 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.
Web: AGC Target Energy EMS: Target Energy CLI: 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.
Web: AGC Minimum Gain EMS: Minimal Gain CLI: 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 this parameter to take effect, a device reset is required.
Web: AGC Maximum Gain EMS: Maximal Gain CLI: 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 this parameter to take effect, a device reset is required.
Web/EMS: Disable Fast Adaptation CLI: AGC-disable-fast-adaptation [AGCDisableFastAdaptation]	Enables the AGC Fast Adaptation mode. - [0] = Disable (default) - [1] = Enable **Note:** For this parameter to take effect, a device reset is required.
Energy Detector Parameters	
Enable Energy Detector CLI: energy-detector-enable [EnableEnergyDetector]	Enables the Energy Detector feature. This feature generates events (notifications) when the signal received from the PSTN is higher or lower than a user-defined threshold (defined by the EnergyDetectorThreshold parameter). - [0] Disable (default)

Parameter	Description
	- [1] Enable
Energy Detector Quality Factor CLI: energy-detector-sensitivity [EnergyDetectorQualityFactor]	Defines the Energy Detector's sensitivity level. The valid range is 0 to 10, where 0 is the lowest sensitivity and 10 the highest sensitivity. The default is 4.
Energy Detector Threshold CLI: energy-detector-threshold [EnergyDetectorThreshold]	Defines the Energy Detector's threshold. A signal below or above this threshold invokes an 'Above' or 'Below' event. The threshold is calculated as follows: Actual Threshold = -44 dBm + (EnergyDetectorThreshold * 6) The valid value range is 0 to 7. The default is 3 (i.e., -26 dBm).
Pattern Detection Parameters **Note:** For an overview on the pattern detector feature for TDM tunneling, see DSP Pattern Detector on page 358.	
Web: Enable Pattern Detector [EnablePatternDetector]	Enables the Pattern Detector (PD) feature. - [0] Disable (default) - [1] Enable
[PDPattern]	Defines the patterns that can be detected by the Pattern Detector. The valid range is 0 to 0xFF. **Note:** For this parameter to take effect, a device reset is required.
[PDThreshold]	Defines the number of consecutive patterns to trigger the pattern detection event. The valid range is 0 to 31. The default is 5. **Note:** For this parameter to take effect, a device reset is required.

67.14 Services

67.14.1 SIP-based Media Recording Parameters

The SIP-based media recording parameters are described in the table below.

Table 67-76: SIP-based Media Recording Parameters

Parameter	Description
Web: SIP Recording Application CLI: configure voip/services sip-recording general-setting/enable-sip-rec **[EnableSIPRec]**	Enables the SIP-based Media Recording feature: - **[0]** Disable (default) - **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Recording Server (SRS) Destination Username CLI: configure voip/services sip-recording general-	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

Parameter	Description
setting/siprec-server-dest-username **[SIPRecServerDestUsername]**	user part is defined.
SIP Recording Routing Table	
Web: SIP Recording Routing table CLI: configure voip/services sip-recording sip-rec-routing **[SIPRecRouting]**	Defines SIP Recording Routing rules (calls to record). The format of the ini file table parameter is as follows: [SIPRecRouting] FORMAT SIPRecRouting_Index = SIPRecRouting_RecordedIPGroupID, SIPRecRouting_RecordedSourcePrefix, SIPRecRouting_RecordedDestinationPrefix, SIPRecRouting_PeerIPGroupID, SIPRecRouting_PeerTrunkGroupID, SIPRecRouting_Caller, SIPRecRouting_SRSIPGroupID; [\SIPRecRouting] For a detailed description of this table, see "Configuring SIP Recording Routing Rules" on page 203.

67.14.2 RADIUS and LDAP Parameters

67.14.2.1 General Parameters

The general RADIUS and LDAP parameters are described in the table below.

Table 67-77: General RADIUS and LDAP Parameters

Parameter	Description
Web: Use Local Users Database CLI: configure system > mgmt-auth > use-local-users-db **[MgmtLocalDatabaseBehavior]**	Defines when the device uses its local user database (Web Users table) for LDAP- or RADIUS-based management-user login authentication. ▪ **[0]** When No Auth Server Defined = (Default) When no LDAP/RADIUS server is configured (or as fallback if the server is inaccessible). ▪ **[1]** Always = Always first verify the user's credentials in the Web Users table, and if not found, then search the LDAP/RADIUS server.
Web: Behavior upon Authentication Server Timeout CLI: configure system > mgmt-auth > timeout-behavior **[MgmtBehaviorOnTimeout]**	Defines the device's response when a connection timeout occurs with the LDAP/RADIUS server. ▪ **[0]** Deny Access = User is denied access to the management platform. ▪ **[1]** Verify Access Locally = (Default) Device verifies the user's credentials in its Web Users table (local database). **Note:** The parameter is applicable to LDAP- or RADIUS-based management-user login authentication.
Web: Default Access Level CLI: default-access-level **[DefaultAccessLevel]**	Defines the default access level for the device when the LDAP/RADIUS response doesn't include an access level attribute for determining the user's management access level. The valid range is 0 to 255. The default is 200 (i.e., Security Administrator). **Note:** The parameter is applicable to LDAP- or RADIUS-based management-user login authentication and authorization.

67.14.2.2 RADIUS Parameters

The RADIUS parameters are described in the table below.

Table 67-78: RADIUS Parameters

Parameter	Description
RADIUS Accounting Parameters	
Web: Enable RADIUS Access Control CLI: enable **[EnableRADIUS]**	Enables the RADIUS application. ▪ **[0]** Disable (Default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
Web: Accounting Server IP Address	Defines the IP address of the RADIUS accounting server.

Parameter	Description
CLI: accounting-server-ip **[RADIUSAccServerIP]**	
Web: Accounting Port CLI: accounting-port **[RADIUSAccPort]**	Defines the port of the RADIUS accounting server. The default is 1646.
Web/EMS: RADIUS Accounting Type CLI: radius-accounting **[RADIUSAccountingType]**	Determines when the RADIUS accounting messages are sent to the RADIUS accounting server. ▪ **[0]** At Call Release = (Default) Sent at call release only. ▪ **[1]** At Connect & Release = Sent at call connect and release. ▪ **[2]** At Setup & Release = Sent at call setup and release.
Web: AAA Indications EMS: Indications CLI: aaa-indications **[AAAIndications]**	Determines the Authentication, Authorization and Accounting (AAA) indications. ▪ **[0]** None = (Default) No indications. ▪ **[3]** Accounting Only = Only accounting indications are used.
General RADIUS Parameters	
Web: Use RADIUS for Web/Telnet Login EMS: Web Use Radius Login CLI: enable-mgmt-login **[WebRADIUSLogin]**	Enables RADIUS queries for Web and Telnet login authentication. When enabled, logging into the device's Web and Telnet embedded servers is done through a RADIUS server. The device communicates with a user-defined RADIUS server and verifies the given username and password against a remote database, in a secure manner. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For RADIUS login authentication to function, you also need to set the following parameters: ✓ EnableRADIUS = 1 (Enable) ✓ WebAuthMode = 0 (Basic Mode) ▪ RADIUS authentication requires HTTP basic authentication, where the username and password are transmitted in clear text over the network. Therefore, it's recommended to set the HTTPSOnly parameter to 1 to force the use of HTTPS, since the transport is encrypted.
Web: RADIUS Authentication Server IP Address EMS: RADIUS Auth Server IP CLI: auth-server-ip **[RADIUSAuthServerIP]**	Defines the IP address of the RADIUS authentication server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Authentication Server Port EMS: RADIUS Auth Server Port CLI: auth-server-port **[RADIUSAuthPort]**	Defines the port of the RADIUS authentication server. **Note:** For this parameter to take effect, a device reset is required.
Web: RADIUS Shared Secret EMS: RADIUS Auth Server Secret CLI: shared-secret **[SharedSecret]**	Defines the 'secret' used to authenticate the device to the RADIUS server. This should be a cryptically strong password.

Parameter	Description
RADIUS Authentication Parameters	
Web: Password Local Cache Mode CLI: local-cache-mode **[RadiusLocalCacheMode]**	Defines the device's mode of operation regarding the timer (configured by the parameter RadiusLocalCacheTimeout) that determines the validity of the username and password (verified by the RADIUS server). ▪ **[0]** Absolute Expiry Timer = When you access a Web page, the timeout doesn't reset, instead it continues decreasing. ▪ **[1]** Reset Timer Upon Access = (Default) Upon each access to a Web page, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).
Web: Password Local Cache Timeout CLI: local-cache-timeout **[RadiusLocalCacheTimeout]**	Defines the time (in seconds) the locally stored username and password (verified by the RADIUS server) are valid. When this time expires, the username and password become invalid and a must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFF. The default is 300 (5 minutes). ▪ **[-1]** = Never expires. ▪ **[0]** = Each request requires RADIUS authentication.
Web: RADIUS VSA Vendor ID CLI: vsa-vendor-id **[RadiusVSAVendorID]**	Defines the vendor ID that the device accepts when parsing a RADIUS response packet. The valid range is 0 to 0xFFFFFFFF. The default is 5003.
Web: RADIUS VSA Access Level Attribute CLI: vsa-access-level **[RadiusVSAAccessAttribute]**	Defines the code that indicates the access level attribute in the Vendor Specific Attributes (VSA) section of the received RADIUS packet. The valid range is 0 to 255. The default is 35.
[MaxRADIUSSessions]	Defines the number of concurrent calls that can communicate with the RADIUS server (optional). The valid range is 0 to 240. The default is 240.
EMS: RADIUS Auth Number of Retries **[RADIUSRetransmission]**	Defines the number of retransmission retries. The valid range is 1 to 10. The default is 3.
[RadiusTO]	Defines the time interval (measured in seconds) that the device waits for a response before a RADIUS retransmission is issued. The valid range is 1 to 30. The default is 10.

67.14.2.3 LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below.

Table 67-79: LDAP Parameters

Parameter	Description
Web: LDAP Service CLI: configure voip/ldap/enable **[LDAPServiceEnable]**	Enables the LDAP feature. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Note:** For this parameter to take effect, a device reset is

Parameter	Description
	required.
CLI: search-dns-in-parallel **[LDAPSearchDNsinParallel**]	Defines the method of how the device queries the DN object within each LDAP server. - **[0]** Sequential = (Default) The query is done in each DN object, one by one, until a result is returned. - **[1]** Parallel = The query is done in all DN objects at the same time.
Web: LDAP Search Server Method CLI: ldap-search-server-method **[LDAPSearchServerMethod]**	Defines the method of how the device queries between two LDAP severs. - **[0]** Sequential = The device first queries one of the LDAP servers, and if the DN object is not found, it queries the second LDAP server. - **[1]** Parallel = (Default) The device queries the LDAP servers at the same time.
Web: LDAP Authentication Filter CLI: configure voip > ldap > auth-filter **[LDAPAuthFilter]**	Defines the LDAP search filter attribute for searching the login username in the directory's subtree for LDAP-based user authentication and authorization. You can use the dollar ($) sign to represent the username. For example, if this parameter is set to "(sAMAccountName=$)" and the user logs in with the username "SueM", the LDAP query is run for sAMAcountName=SueM.
Web: Use LDAP for Web/Telnet Login CLI: configure voip > ldap > enable-mgmt-login **[LoginMgmtLDAPLogin]**	Enables LDAP-based management-user login authentication and authorization. - **[0]** Disable (default) - **[1]** Enable **Note:** For this parameter to take effect, a device reset is required.
[LDAPDebugMode]	Determines whether to enable the LDAP task debug messages. This is used for providing debug information regarding LDAP tasks. The valid value range is 0 to 3. The default is 0.
Web: MS LDAP OCS Number attribute name EMS: LDAP ocs Number Attribute Name CLI: ldap-ocs-nm-attr **[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-PrimaryUserAddress".
Web: MS LDAP PBX Number attribute name CLI: ldap-pbx-nm-attr **[MSLDAPPBXNumAttributeName]**	Defines the name of the attribute that represents the user PBX number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "telephoneNumber".
Web: MS LDAP MOBILE Number attribute name CLI: ldap-mobile-nm-attr **[MSLDAPMobileNumAttributeName]**	Defines the name of the attribute that represents the user Mobile number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is "mobile".

Parameter	Description
CLI: ldap-private-nm-attr **[MSLDAPPrivateNumAttributeName]**	Defines the name of the attribute that represents the user's private number in the AD. If this value equals the value of the MSLDAPPrimaryKey or MSLDAPSecondaryKey parameter, then the device queries the AD for the destination number in this private attribute name; otherwise, this parameter is not used as a search key. The default is "msRTCSIP-PrivateLine".
Web: MS LDAP DISPLAY Name Attribute Name CLI: ldap-display-nm-attr **[MSLDAPDisplayNameAttributeName]**	Defines the attribute name that represents the Calling Name in the AD for LDAP queries based on calling number. The valid value is a string of up to 49 characters. The default is "displayName".
CLI: ldap-primary-key **[MSLDAPPrimaryKey]**	Defines the name of the attribute used as a query search key for the destination number in the AD. This is used instead of the "PBX" attribute name (configured by the MSLDAPPBXNumAttributeName parameter). The default is not configured.
CLI: ldap-secondary-key **[MSLDAPSecondaryKey]**	Defines the name of the attribute used as the second query search key for the destination number in the AD, if the primary search key or PBX search is not found.
LDAP Cache Service CLI: cache **[LDAPCacheEnable]**	Enables the LDAP cache service. ▪ **[0]** Disable (default) ▪ **[1]** Enable **Notes:** ▪ For this parameter to take effect, a device reset is required. ▪ For more information on LDAP caching, see "Configuring the Device's LDAP Cache" on page 220.
LDAP Cache Entry Timeout CLI: entry-timeout **[LDAPCacheEntryTimeout]**	Defines the duration (in minutes) that an entry in the LDAP cache is valid. If the timeout expires, the cached entry is only used if there is no connectivity with the LDAP server. The default is 1200.
LDAP Cache Entry Removal Timeout CLI: entry-removal-timemout **[LDAPCacheEntryRemovalTimeout]**	Defines the duration (in hours) after which the LDAP entry is removed from the cache. The default is 0.
LDAP Configuration Table	
Web: LDAP Configuration Table CLI: configure voip > ldap > ldap-configuration **[LdapConfiguration]**	Defines the LDAP servers. [LdapConfiguration] FORMAT LdapConfiguration_Index = LdapConfiguration_LdapConfServerIp, LdapConfiguration_LdapConfServerPort, LdapConfiguration_LdapConfServerMaxRespondTime, LdapConfiguration_LdapConfServerDomainName, LdapConfiguration_LdapConfPassword, LdapConfiguration_LdapConfBindDn, LdapConfiguration_LdapConfInterfaceType, LdapConfiguration_Type, LdapConfiguration_MngmAuthAtt, LdapConfiguration_ConnectionStatus;

Parameter	Description
	[\LdapConfiguration] For a detailed description of this table, see "Configuring LDAP Servers" on page 212.
LDAP Server Search DN Table	
Web: LDAP Search DN Table CLI: configure voip > ldap > ldap-servers-search-dns **[LdapServersSearchDNs]**	Defines the full base path (i.e., distinguished name / DN) to the objects in the AD where the query is done, per LDAP server. [LdapServersSearchDNs] FORMAT LdapServersSearchDNs_Index = LdapServersSearchDNs_Base_Path, LdapServersSearchDNs_LdapConfigurationIndex, LdapServersSearchDNs_SearchDnInternalIndex; [\LdapServersSearchDNs] For a detailed description of this table, see "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 216.
Management LDAP Groups Table	
Web: Management LDAP Groups Table CLI: configure voip > ldap > mgmt-ldap-groups **[MgmntLDAPGroups]**	Defines the users group attribute in the AD and corresponding management access level. [MgmntLDAPGroups] FORMAT MgmntLDAPGroups_Index = MgmntLDAPGroups_LdapConfigurationIndex, MgmntLDAPGroups_GroupIndex, MgmntLDAPGroups_Level, MgmntLDAPGroups_Group; [\MgmntLDAPGroups] For a detailed description of this table, see "Configuring Access Level per Management Groups Attributes" on page 217.

67.14.3 Least Cost Routing Parameters

The Least Cost Routing (LCR) parameters are described in the table below.

Table 67-80: LCR Parameters

Parameter	Description
Web: Routing Rule Groups Table CLI: configure voip > services least-cost-routing routing-rule-groups **[RoutingRuleGroups]**	This table parameter enables the LCR feature and configures the average call duration and default call cost. The default call cost determines whether routing rules that are not configured with a Cost Group are considered as a higher or lower cost route compared to other matching routing rules that are assigned Cost Groups. [RoutingRuleGroups] FORMAT RoutingRuleGroups_Index = RoutingRuleGroups_LCREnable, RoutingRuleGroups_LCRAverageCallLength, RoutingRuleGroups_LCRDefaultCost; [\RoutingRuleGroups] **Note:** For a detailed description of this table, see "Enabling LCR and Configuring Default LCR" on page 236.
Web: Cost Group Table EMS: Cost Group	This table parameter configures the Cost Groups for LCR, where each Cost Group is configured with a name, fixed call connection charge,

Parameter	Description
Provisioning > Cost Group CLI: configure voip > services least-cost-routing cost-group **[CostGroupTable]**	and a call rate (charge per minute). [CostGroupTable] FORMAT CostGroupTable_Index = CostGroupTable_CostGroupName, CostGroupTable_DefaultConnectionCost, CostGroupTable_DefaultMinuteCost; [\CostGroupTable] For example: CostGroupTable 2 = "Local Calls", 2, 1; **Note:** For a detailed description of this table, see "Configuring Cost Groups" on page 238.
Web: Cost Group > Time Band Table EMS: Time Band Provisioning > Time Band CLI: configure voip > services least-cost-routing cost-group-time-bands **[CostGroupTimebands]**	This table parameter configures time bands and associates them with Cost Groups. [CostGroupTimebands] FORMAT CostGroupTimebands_TimebandIndex = CostGroupTimebands_StartTime, CostGroupTimebands_EndTime, CostGroupTimebands_ConnectionCost, CostGroupTimebands_MinuteCost; [\CostGroupTimebands] **Note:** For a detailed description of this table, see "Configuring Time Bands for Cost Groups" on page 239.

67.14.4 Call Setup Rules Parameters

The Call Setup Rules parameters are described in the table below.

Table 67-81: Call Setup Rules Parameters

Parameter	Description
Web: Call Setup Rules CLI: configure voip/services call-setup-rules **[CallSetupRules]**	This table parameter defines Call Setup Rules that the device runs at call setup for LDAP-based routing and other advanced routing logic requirements including manipulation. [CallSetupRules] FORMAT CallSetupRules_Index = CallSetupRules_RulesSetID, CallSetupRules_AttributesToQuery, CallSetupRules_AttributesToGet, CallSetupRules_RowRole, CallSetupRules_Condition, CallSetupRules_ActionSubject, CallSetupRules_ActionType, CallSetupRules_ActionValue; [\CallSetupRules] **Note:** For a detailed description of this table, see "Configuring Call Setup Rules" on page 241.

68 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.

68.1 Signaling-Media Sessions & User Registrations

The table below lists the maximum capacity. This includes SIP signaling sessions, SBC sessions, and registered users.

Table 68-1: Maximum Call Sessions and Registered Users

Model	Signaling Sessions	Media Sessions			Registered Users
		RTP-to-RTP	SRTP-RTP **and SRTP-TDM**	Codec Transcoding	
Mediant 800 MSBR	60	60	60	See Channel Capacity and Capabilities on page 1016	200
Mediant 800B MSBR	180	100	100	See Channel Capacity and Capabilities on page 1016	500

Notes:

- The capacity figures listed in the table below are accurate at the time of publication of this document. However, these figures may change due to a later software update. For the latest figures, please contact your AudioCodes sales representative.
- The maximum number of SBC signaling and media sessions are specified in the installed Software License Key, which defines maximum figures for each one separately.
- The maximum number of voice transcoding sessions is specified in the installed Software License Key.
- *Registered Users* indicates the maximum number of users that can be registered with the device (i.e., in the device's registration database). This applies to all the supported applications.
- Regarding signaling, media, and transcoding session resources:
  - √ A signaling session is a SIP dialog session between two SIP entities, traversing the SBC and using one signaling session resource.
  - √ A media session is an audio (RTP or SRTP), fax (T.38), or video session between two SIP entities, traversing the SBC and using one media session resource.
  - √ In case of direct media (i.e., *Anti-tromboning* / *Non-Media Anchoring*), where only SIP signaling traverses the SBC and media flows directly between the SIP entities, only a signaling session resource is used. Thus, if a greater signaling session capacity exists than media, even when media session resources have been exhausted, additional signaling sessions can still be handled for direct-media calls.
  - √ For call sessions requiring transcoding, one transcoding session resource is also used. For example, for a non-direct media call in which one leg uses G.711 and the other leg uses G.729, one signaling resource, one media session resource, and one transcoding session resource is used.

68.2 Channel Capacity and Capabilities

The DSP channel capacity and SBC session capacity for Mediant 800 MSBR and Mediant 800B MSBR are shown in the table below.

Table 68-2: Mediant 800/B MSBR Channel Capacity per PSTN and Capabilities

Telephony Interface Assembly	DSP Channels Allocated for PSTN	SBC Transcoding Sessions								Conf. Participants	Max. SBC Sessions	
		From Profile 2 with Additional Advanced DSP Capabilities						**To Profile 1**	**To Profile 2**		**Mediant 800**	**Mediant 800B**
		IPM Detectors	**G.722**	**AMR WB**	**SILK NB / iLBC**	**SILK WB**	**V.150.1**					
2 x E1/T1	60/48	-	-	-	-	-	-	4/12	3/11	-	0/12	120/132
2 x T1	48	√	-	-	-	-	√	9	7	-	12	132
1 x E1/T1 & 8 x FXS/FXO Mix	38/32	√	-	-	-	-	-	16/21	14/18	-	22/28	142/148
	38/32	√	-	-	√	-	-	3/7	2/6	-	22/28	142/148
1 x E1/T1	30/24	√	-	-	√	-	√	9/14	7/11	-	30/36	150/156
1 x E1 & 4 x BRI	38	√	-	-	-	-	-	16	14	-	22	145
1 x E1 & 4 x FXS	34	√	-	-	-	-	-	19	16	-	26	146
2 x E1 & 4 x FXS	64	-	-	-	-	-	-	0	0	-	0	116
4 x BRI & 4 x FXS & 4 x FXO	16	√	-	-	-	-	-	3	2	-	44	164
8 x BRI & 4 x FXS	20	√	-	-	-	-	-	1	1	-	40	160
8 x BRI	16	√	-	-	-	-	-	3	2	-	44	164
12 x FXS	12	√	-	-	√	-	√	1	1	-	48	168
4 x FXS & 8 x FXO	12	√	-	-	√	-	-	1	1	-	48	168
8 x FXS & 4 x FXO	12	√	-	-	√	-	-	1	1	-	48	168

Telephony Interface Assembly	DSP Channels Allocated for PSTN	SBC Transcoding Sessions								Conf. Participants	Max. SBC Sessions	
		From Profile 2 with Additional Advanced DSP Capabilities						**To Profile 1**	**To Profile 2**			
		IPM Detectors	**G.722**	**AMR WB**	**SILK NB / iLBC**	**SILK WB**	**V.150.1**				**Mediant 800**	**Mediant 800B**
4 x BRI & 4 x FXS	12	√	-	-	√	-	-	1	1	-	48	168
4 x FXS & 4 x FXO	8	-	-	-	-	-	-	9	8	6	52	172
	8	√	-	-	√	-	-	4	3	-	52	172
4 x BRI	8	-	-	-	-	-	-	9	8	6	52	172
	8	√	-	-	√	-	-	4	3	-	52	172
1/2/3 x BRI	2/4/6	-	-	-	-	-	-	13/12/10	12/11/10	-	56/52/48	176/172/168
	2/4/6	√	-	-	√	-	-	9/7/6	7/6/5	-	56/52/48	176/172/168
4 x FXS or 4 x FXO	4	-	-	-	√	-	√	7	6	-	56	176
	4	-	√	-	-	-	-	12	10	8	56	176
	4	-	-	-	√	-	-	8	7	7	56	176
	4	√	-	√	√	-	-	7	6	4	56	176
	4	√	-	√	√	√	-	5	4	4	56	176
FXS, FXO, and/or BRI, but not in use	0	-	-	-	-	-	-	15	14	-	60	180

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, and AMR-NB, T.38 with fax detection, in-band signaling, and Silence Compression.
- All hardware assemblies also support the following DSP channel capabilities: echo cancellation (EC), CID (caller ID), RTCP XR reporting, and SRTP.
- *SBC Enhancements* includes the network Acoustic Echo Suppressor.
- *IPM Detectors* includes Automatic Gain Control (AGC) and Answer Detector (AD).
- *V.150.1* is supported only for the US Department of Defense (DoD).
- *Transcoding Sessions* represents part of the total SBC Sessions.
- *Conference Participants* represents the number of participants in one or more conference (bridge), where each conference may include three or more participants. Conferences are supported on all above configurations. For figures on the maximum number of participants per configuration, contact your AudioCodes sales representative.
- For availability of the telephony assemblies listed in the table above, contact your AudioCodes sales representative.

This page is intentionally left blank.

69 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 69-1: Technical Specifications

Function	Specification
Telephony Interfaces	
PRI Interfaces	Up to two E1/T1/J1 spans (RJ-48c ports)
BRI Interfaces	Up to 8 BRI ports (16 calls), network S/T interfaces - NT or TE termination; 2W per port (power supplied)
Analog Interfaces	▪ Up to 12 FXS ports (RJ-11) ▪ Up to 12 FXO ports (RJ-11) ▪ Analog (FXS) Lifeline in case of power failure. FXS Analog Lifeline is on FXS Port 1. For the combined FXS-FXO hardware configuration, one Lifeline is available; for the 12-FXS hardware configuration, up to three Lifelines are available. ▪ FXS Loop Impedance - Up to 1,600 Ohms **Note:** The device is an indoor unit and thus, the device and these interfaces must be installed only indoors.
Networking Interfaces	
WAN	▪ Multiple WAN interfaces: ▪ Copper 1000Base-T (Gigabit Ethernet) using RJ-45 port ▪ Dual-mode, Optical Fiber Small Form-Factor Pluggable (SFP), supporting 100/1000 Mbps Ethernet ▪ SHDSL (Annex, A, B, F, G), 2Base-2L using RJ-45 ports ▪ ADSL2+(Annex A, B, I, J, L, M) using RJ-11 ports ▪ VDSL1, VDSL2 (bandplan 997, 998, profiles 8, 12, 17, 30), 10PASS-TS ▪ 4 E1/T1 WAN data service unit/channel service unit (DSU/CSU) ▪ 3G Cellular (primary or backup) using USB modem Additional WAN support: ▪ EFM (802.3ah) for VDSL2 and SHDSL ▪ ATM for ADSL and SHDSL ▪ RFC2684, VC-MUX, LLC SNAP ▪ 8 PVCs, multiple IP interfaces ▪ IP, IPoE, PPPoE, PPPoE, IPCP, NCP, LCP, PAP, CHAP, PPP, ML - PPP ▪ ATM Service Categories (UBR, VBR-RT, VBR-nRT, CBR
LAN	▪ 4 Gigabit Ethernet (10/100/1000Base-T) using RJ-45 ports ▪ 8 Fast Ethernet (FE) 10/100Base-TX ports using RJ-45 ports ▪ Power-over-Ethernet (PoE) on all ports is optional, compliant to 802.3af-2003 and EEE 802.3at-2003 with auto-detection, supporting up to 15.4W and 30W per port respectively. PoE management

Function	Specification
Wi-Fi	▪ Wireless LAN 802.11 b/g/n access point at 2.4 and 5 GHz, integrated 3 Tx / 3 Rx, enabling data rates of up to 300 Mbps ▪ 2 Wi-Fi antennas
USB	2 USB ports for optional, 3G cellular WAN modem and/or USB storage services
Media Processing	
Voice Coders	G.711, G.726, G.723.1, G.726, G.729A, AMR, G.722, iLBC, AMR-WB, SILK-NB, SILK-WB Independent dynamic vocoder selection per channel
Echo Cancellation	G.165 and G.168-2002, with 32, 64 or 128 msec tail length
Quality Enhancement	Dynamic programmable jitter buffer, VAD, CNG
DTMF/MF Tones	Packet-side or PSTN-side detection and generation, RFC 2833 compliant DTMF relay and Call Progress tones Detection and Generation
IP Transport	VoIP (RTP/RTCP) per IETF RFC 3550 and 3551, IPv6
Fax Transport	T.38 compliant (real time fax), Automatic bypass to PCM
Signaling	
Digital PSTN Protocols	▪ CAS: MF-R1: T1 CAS (E&M, loop start, Feature Group-D, E911CAMA), E1 CAS (R2 MFC), R1.5, numerous protocol and country variants ▪ ISDN PRI: ETSI/EURO ISDN, ANSI NI2 and other variants (DMS100, 5ESS), VN3, VN4, VN6 ▪ ISDN BRI: Euro ISDN, VN4/6 or QSIG
Analog	Loop Start FXS/FXO, Caller ID, polarity reversal, distinctive ringing, visual Message Waiting Indication
OSN Server Platform (Optional)	
Single Chassis Integration	Embedded, open Network Solution Platform for third-party services
CPU Options	▪ Atom 1.5 GHz ▪ 2nd Generation Intel Core Celeron 1.6 GHz ▪ 3rd Generation Intel Core i7 Quad Core ▪ Intel Atom N2800 1.86 GHz Dual Core
Memory	1G RAM, 2G RAM, 4G RAM depends on model
Control and Management	
Control Protocols	▪ Static and dynamic routing (RIP1, RIP2, OSPF, BGP), multi-VRF, L2 bridging ▪ SIP-TCP, SIP-UDP, SIP-TLS, IPv6 ▪ Cloud Resilience Package and Standalone Survivability for service continuity, Microsoft RBA
Operations & Management	▪ AudioCodes' Element Management System (EMS) ▪ Embedded HTTP Web Server, SNMP V2/V3 ▪ Telnet, SSH, TR-069 ▪ Remote configuration and software download via HTTP/S, RADIUS, Syslog (for events and alarms)
Security	

Function	Specification
Session Border Controller (SBC)	▪ SIP Header conversion ▪ SIP Normalization ▪ Survivability ▪ IP-to-IP routing translations of various SIP transport types; UDP, TCP, TLS ▪ Translation of RTP, SRTP ▪ SIP trunk with multi-ITSP (Registrations to ITSPs is invoked independently) ▪ Topology hiding ▪ Call Admission Control ▪ Call Black/White list
Data Security (Optional)	▪ IPsec ▪ ESP – Tunnel mode ▪ IKE mode – IPsec VPN ▪ IDS/IPS: ✓ Fragmented traffic ✓ Malformed Request ✓ Ping of Death ✓ Properly formed request from unauthenticated source ✓ DDoS attack ✓ SYN flood ▪ Stateful packet inspection firewall ▪ DMZ Host ▪ Port Triggering ▪ Packet Filtering ▪ Application Layer Gateway
Data-Routing Features	
LAN Switching	▪ 802.1Q for VLAN tagging ▪ Rapid Spanning Tree Protocol / RSTP (802.1D-2004 and 802.1w) ▪ LAN switch ports with non-blocking switching performance ▪ Up to four RJ-45 10/100/1000Base-T (Gigabit Ethernet) LAN ports ▪ Eight RJ-45 10/100Base-TX (Fast Ethernet) LAN ports ▪ Power-over-Ethernet (PoE) supported on all LAN ports, complying with IEEE 802.3af-2003 (15.4W maximum wattage per port; 120 total wattage over all ports) ▪ LAN ports supporting half- and full-duplex modes, auto-negotiation, and straight or crossover cable detection.
Routing	▪ Dynamic Host Configuration Protocol (DHCP): ✓ DHCP server, DHCP relay and DHCP client ✓ DHCP server supports fixed binding of IP-to-MAC address ✓ DHCP server supports user-defined DNS server allocation ▪ Multiple IP interfaces for LAN routing: IP interfaces assignment to different VLANs ▪ Assignment of different VLAN IDs to VoIP and Data traffic ▪ Symmetric High-Speed Digital Subscriber Line (SHDSL): ✓ ATM: RFC 2684 over AAL5 with LLC-SNAP and VC-MUX over AAL5; ATM UNI 4.1; UBR, CBR, VBR-RT and VBR-NRT; IP QoS; Multiple IP interfaces; RFC 2684 Routed Mode; RFC 2364 PPPoA; RFC 2516 PPPoE over ATM; Up to 8 PVCs ✓ EFM: ITU G.991.2 Annex E for Ethernet, also known as EFM or

Function	Specification
	2Base-TL, as defined in IEEE 802.3ah; Up to 8 IP interfaces for EFM/2Base-TL; IP QoS classification, Marking, scheduling and shaping; 802.1Q VLANs ▪ Routing protocols: ✓ Static routing ✓ RIPv1 - RFC 1058 ✓ RIPv2 - RFC 2453 ✓ OSPFv2 - RFC 2328 ✓ BGPv4 - RFC 1771 and RFC 2858 ✓ BGP Extended Community Attribute for BGP/MPLS VPNs ✓ Policy-based routing (e.g. DSCP-based and BGP policy routing) ▪ Network Address Translation (NAT/NAPT): ✓ ACL-like classification with ALG support ✓ Source and destination-based IP addresses ACLs ✓ Multiple NAT and NAPT WAN addresses ▪ WAN access via PPPoE, L2TP, DHCP ▪ Quality of service (QoS): ✓ Traffic Classification and Marking: Connection-based (with SPI) or packet-based classification; VoIP classification for SIP signaling and media traffic (tracking UDP ports selected by SDP offer-answer negotiations); DSCP and 802.1p marking; Explicit classification criteria (source/destination MAC and IP addresses; Protocol ALG-based; L4 port numbers DSCP/802.1p; Length of packets or their data portion only) ✓ Traffic scheduling and shaping: Ingress traffic policing; Traffic reservation - when not utilized, other classes are served with extra bandwidth; Maximum egress traffic shaping; Scheduling (Strict Priority, Fair, Weighted Round-Robin Queuing, and Class-Based WRR Queuing); Queue management (RED; TCP Serialization Reduction to minimize jitter in VoIP environments)
Data Security	▪ Access Control for pinpoint security policy ▪ Extensive list of ALG-modules combined with SPI for error-free configuration and maximum security ▪ Port-forwarding and DMZ support for local applications and hosts ▪ Website Restriction allows static URL-based blocking of public/extranet websites ▪ Advanced Filtering allows full control on Inbound/Outbound Rules per interface/device ▪ Site-to-Site VPN: ✓ Supports two IPSec use-cases: Site-to-Site Gateway-to-Gateway) VPN; Teleworker (User-to-Gateway) VPN ✓ Fully compliant with IPSec RFCs: RFC 2401 - Security Architecture for IP; RFC 2402 - IP Authentication Header; RFC 2406 – ESP; RFC 2403 and RFC 2404 for Authentication ▪ L2TP Client-Server VPN: ✓ Supports two VPN use-cases: Server support for remote Teleworker VPN access; Client-to-Gateway support with L2TP ✓ Layer Two Tunneling Protocol - RFC 2661 (with L2TP/IPSec) ✓ Support all WiN OS versions as well as Linux ▪ DoS and IDS/IPS: ✓ Denial of Service (DoS) protection: TCP RST, Ping Flood, ICMP Echo storm, UDP snork attack, ICMP Smurf, UDP fraggle and more ✓ IP spoofing attacks: FTP bounce, Broadcast/multicast source IP attack ✓ Intrusion and scanning attacks: IP source route, ICMP Echo reply

Function	Specification
	without request, ICMP Ping sweep, TCP Stealth; Scan (FIN, XMAS, NULL), UDP port, FTP passive attack, loopback / Echo chargen, Block security hazard ICMP messages ✓ IP fragment overlap, Ping of Death, Fragmentation attacks and more
Supported RFCs	▪ IP/Routing: ✓ RFC 768 UDP ✓ RFC 791 IP ✓ RFC 792 ICMP ✓ RFC 793 TCP ✓ RFC 959 FTP ✓ RFC 2822 Internet Message Format ✓ RFC 1305 NTP ✓ RFC 826 ARP ✓ RFC 2131 DHCP ✓ RFC 1918 IP Addressing ✓ RFC 2516 PPPoE ✓ RFC 2663 NAT ▪ QoS: ✓ IEEE 802.1p Priority Tagging ✓ RFC 1349 ✓ RFC 2475 ✓ RFC 2597 ✓ RFC 3246 ▪ IPSec: ✓ RFC 2401 Security Architecture for IP ✓ RFC 2402 AH - IP Authentication Header ✓ RFC 2403 IPsec Authentication - MD5 ✓ RFC 2404 IPsec Authentication - SHA-1 ✓ RFC 2405 IPsec Encryption - DES ✓ RFC 2406 ESP - IPsec Encryption ✓ RFC 2407 IPsec DOI ✓ RFC 2408 ISAKMP ✓ RFC 2409 IKE ✓ RFC 2410 IPsec Encryption – NULL ✓ RFC 2411 ✓ RFC 2412 OAKLEY
Hardware Specifications	
Power Supply	Single universal AC power supply 100-240 VAC, 1.5A, 50-60 Hz
Physical Dimensions	32 x 34.5 cm (12.6 x 13.6 inches) x 1U
Weight	2.5 kg (5.5 lb)
Environmental	▪ Operational: 5 to 40°C (41 to 104°F) ▪ Storage: -25 to 85°C (-13 to 185°F) ▪ Humidity: 10 to 90% non-condensing

AudioCodes

User's Manual Ver. 6.8

www.audiocodes.com