Mediant Software SBC
Virtual Edition (VE) & Server Edition (SE)

Version 7.2
Table of Contents

1 Introduction ....................................................................................................... 27
  1.1 Product Overview .......................................................................................... 27
  1.2 Typographical Conventions ........................................................................... 27
  1.3 Getting Familiar with Configuration Concepts and Terminology ................. 28
    1.3.1 SBC Application ....................................................................................... 28

Getting Started with Initial Connectivity .............................................................. 33

2 Introduction ....................................................................................................... 35

3 Default OAMP IP Address ............................................................................... 37

4 Installing the Software ..................................................................................... 39

5 Changing Default IP Address to Suit your Network Addressing Scheme ....... 41

6 Licensing the Device ......................................................................................... 43

Management Tools ..................................................................................................45

7 Introduction ....................................................................................................... 47

8 Web-Based Management ................................................................................... 49
  8.1 Getting Acquainted with the Web Interface ................................................... 49
    8.1.1 Computer Requirements .......................................................................... 49
    8.1.2 Accessing the Web Interface ..................................................................... 50
    8.1.3 Areas of the GUI ....................................................................................... 51
    8.1.4 Accessing Configuration Pages from Navigation Tree .............................. 53
    8.1.5 Configuring Stand-alone Parameters ....................................................... 55
    8.1.6 Configuring Table Parameters ................................................................... 56
      8.1.6.1 Adding Table Rows ........................................................................... 57
      8.1.6.2 Assigning Rows from Other Tables .................................................. 58
      8.1.6.3 Modifying Table Rows ....................................................................... 59
      8.1.6.4 Deleting Table Rows .......................................................................... 59
      8.1.6.5 Invalid Value Indications .................................................................. 60
      8.1.6.6 Viewing Table Rows .......................................................................... 62
      8.1.6.7 Sorting Tables by Column .................................................................. 63
      8.1.6.8 Changing Index Position of Table Rows ........................................... 63
      8.1.6.9 Searching Table Rows ........................................................................ 64
    8.1.7 Searching for Configuration Parameters .................................................. 65
    8.1.8 Getting Help ............................................................................................... 66
    8.1.9 Logging Off the Web Interface ................................................................... 66
  8.2 Customizing the Web Interface ...................................................................... 66
    8.2.1 Replacing the Corporate Logo ................................................................. 66
      8.2.1.1 Replacing the Corporate Logo with an Image .................................... 67
      8.2.1.2 Replacing the Corporate Logo with Text ........................................... 68
    8.2.2 Customizing the Product Name .................................................................. 69
    8.2.3 Customizing the Favicon .......................................................................... 70
    8.2.4 Creating a Login Welcome Message .......................................................... 71
  8.3 Configuring Additional Management Interfaces .......................................... 71
  8.4 Configuring Management User Accounts .................................................... 73
8.5 Displaying Login Information upon Login ............................................................... 78
8.6 Viewing Logged-In User Information .................................................................... 78
8.7 Configuring Web Session and Access Settings ..................................................... 79
8.8 Changing Login Password for Administrator and Monitor Users ....................... 80
8.9 Configuring Secured (HTTPS) Web ....................................................................... 80
8.10 Web Login Authentication using Smart Cards ................................................... 81
8.11 Configuring Web and Telnet Access List .............................................................. 81

9 CLI-Based Management ........................................................................................... 83
  9.1 Enabling CLI ........................................................................................................ 83
    9.1.1 Enabling Telnet for CLI ................................................................................. 83
    9.1.2 Enabling SSH with RSA Public Key for CLI ................................................. 84
  9.2 Configuring Maximum Telnet/SSH Sessions ....................................................... 86
  9.3 Establishing a CLI Session .................................................................................. 86
  9.4 Viewing Current CLI Sessions ............................................................................ 87
  9.5 Terminating a User’s CLI Session ...................................................................... 88
  9.6 Configuring Displayed Output Lines in CLI Terminal Window ......................... 88

10 SNMP-Based Management .................................................................................... 89
  10.1 Disabling SNMP ............................................................................................... 89
  10.2 Configuring SNMP Community Strings ........................................................... 89
  10.3 Configuring SNMP Trap Destinations with IP Addresses ................................. 91
  10.4 Configuring an SNMP Trap Destination with FQDN ......................................... 93
  10.5 Configuring SNMP Trusted Managers ............................................................. 93
  10.6 Enabling SNMP Traps for Web Activity .......................................................... 94
  10.7 Configuring SNMP V3 Users .......................................................................... 94

11 INI File-Based Management ................................................................................ 97
  11.1 INI File Format ................................................................................................ 97
    11.1.1 Configuring Individual ini File Parameters .................................................. 97
    11.1.2 Configuring Table ini File Parameters ......................................................... 97
    11.1.3 General ini File Formatting Rules ................................................................. 99
  11.2 Configuring an ini File ..................................................................................... 99
  11.3 Loading an ini File to the Device .................................................................... 100
  11.4 Secured Encoded ini File ................................................................................. 100
  11.5 Configuring Password Display in ini File ....................................................... 100
  11.6 INI Viewer and Editor Utility .......................................................................... 102

General System Settings ......................................................................................... 103

12 Configuring SSL/TLS Certificates ...................................................................... 105
  12.1 Configuring TLS Certificate Contexts ............................................................. 105
  12.2 Assigning CSR-based Certificates to TLS Contexts ........................................ 109
  12.3 Viewing Certificate Information ...................................................................... 111
  12.4 Assigning Externally Created Private Keys to TLS Contexts ........................... 113
  12.5 Generating Private Keys for TLS Contexts ..................................................... 114
  12.6 Creating Self-Signed Certificates for TLS Contexts ....................................... 115
  12.7 Importing Certificates into Trusted Certificate Store ......................................... 116
### 12.8 Configuring Mutual TLS Authentication

- **12.8.1 TLS for SIP Clients** .......................................................... 118
- **12.8.2 TLS for Remote Device Management** .............................. 118
- **12.9 Configuring TLS Server Certificate Expiry Check** .............. 120

### 13 Date and Time

- **13.1 Configuring Automatic Date and Time using SNTP** ............. 121
- **13.2 Configuring Date and Time Manually** ................................. 122
- **13.3 Configuring the Time Zone** ................................................ 122
- **13.4 Configuring Daylight Saving Time** ..................................... 123

### 14 Network

- **14.1 Building and Viewing your Network Topology** .................... 127
- **14.2 Configuring Physical Ethernet Ports** ................................. 130
- **14.3 Configuring Ethernet Port Groups** ........................................ 132
- **14.4 Configuring Underlying Ethernet Devices** ........................ 134
- **14.5 Configuring IP Network Interfaces** ..................................... 137
  - **14.5.1 Assigning NTP Services to Application Types** .................. 141
  - **14.5.2 IP Interfaces Table Configuration Guidelines** ................... 141
  - **14.5.3 Networking Configuration Examples** ............................... 142
    - **14.5.3.1 One VoIP Interface for All Applications** ..................... 142
    - **14.5.3.2 VoIP Interface per Application Type** .......................... 142
    - **14.5.3.3 VoIP Interfaces for Combined Application Types** ....... 143
    - **14.5.3.4 VoIP Interfaces with Multiple Default Gateways** ...... 144
- **14.6 Configuring Static IP Routes** .............................................. 144
  - **14.6.1 Configuration Example of Static IP Routes** ...................... 146
  - **14.6.2 Troubleshooting the Static Routes Table** ....................... 147
- **14.7 Network Address Translation Support** ............................... 148
  - **14.7.1 Device Located behind NAT** ........................................... 148
    - **14.7.1.1 Configuring NAT Translation per IP Interface** .......... 149
  - **14.7.2 Remote UA behind NAT** ................................................ 151
    - **14.7.2.1 SIP Signaling Messages** ........................................... 151
    - **14.7.2.2 Media (RTP/RTCP/T.38)** ......................................... 151
- **14.8 Configuring Quality of Service** ......................................... 155
  - **14.8.1 Configuring Class-of-Service QoS** ................................. 155
  - **14.8.2 Configuring DiffServ-to-VLAN Priority Mapping** .......... 157
- **14.9 Configuring ICMP Messages** ............................................ 158
- **14.10 DNS** ............................................................................... 159
  - **14.10.1 Configuring the Internal DNS Table** .............................. 159
  - **14.10.2 Configuring the Internal SRV Table** ............................. 160
- **14.11 Robust Receipt of Media Streams by Media Latching** ......... 162
- **14.12 Multiple Routers Support** ............................................... 163

### 15 Security

- **15.1 Configuring Firewall Settings** ........................................... 165
- **15.2 Configuring TLS for SIP** ................................................... 170
- **15.3 Intrusion Detection System** ................................................. 171
16 Media ........................................................................................................... 181

16.1 Configuring Voice Settings ........................................................................... 181
   16.1.1 Configuring Voice Gain (Volume) Control .................................................. 181
   16.1.2 Configuring Echo Cancellation ................................................................... 181

16.2 Fax and Modem Capabilities ........................................................................ 182
   16.2.1 Fax/Modem Operating Modes .................................................................... 183
   16.2.2 Fax/Modem Transport Modes .................................................................... 183
      16.2.2.1 T.38 Fax Relay Mode .......................................................................... 184
      16.2.2.2 G.711 Fax / Modem Transport Mode .................................................. 186
      16.2.2.3 Fax Fallback ....................................................................................... 187
      16.2.2.4 Fax/Modem Bypass Mode ................................................................... 188
      16.2.2.5 Fax / Modem NSE Mode .................................................................... 189
      16.2.2.6 Fax / Modem Transparent with Events Mode ........................................ 190
      16.2.2.7 Fax / Modem Transparent Mode .......................................................... 190
      16.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection .............................. 191
   16.2.3 V.34 Fax Support ...................................................................................... 191
      16.2.3.1 Bypass Mechanism for V.34 Fax Transmission ....................................... 192
      16.2.3.2 Relay Mode for T.30 and V.34 Faxes ....................................................... 192
   16.2.4 V.152 Support .......................................................................................... 193

16.3 Configuring RTP/RTCP Settings .................................................................... 193
   16.3.1 Configuring the Dynamic Jitter Buffer ....................................................... 193
   16.3.2 Configuring RFC 2833 Payload ................................................................. 194
   16.3.3 Configuring RTP Base UDP Port ................................................................. 195

16.4 Event Detection and Notification using X-Detect Header ............................. 196
   16.4.1 Detecting Answering Machine Beeps .......................................................... 197
   16.4.2 SIP Call Flow Examples of Event Detection and Notification .................... 198

16.5 Answering Machine Detection (AMD) ........................................................... 200
   16.5.1 Configuring AMD ..................................................................................... 202

16.6 Automatic Gain Control (AGC) .................................................................... 203

16.7 Configuring Media (SRTP) Security .............................................................. 204
   16.7.1 SRTP using DTLS Protocol ...................................................................... 207

17 Services ......................................................................................................... 209

17.1 DHCP Server Functionality ........................................................................... 209
   17.1.1 Configuring the DHCP Server ................................................................... 209
   17.1.2 Configuring the Vendor Class Identifier .................................................... 214
   17.1.3 Configuring Additional DHCP Options .................................................... 215
   17.1.4 Configuring Static IP Addresses for DHCP Clients .................................... 217
   17.1.5 Viewing and Deleting DHCP Clients ....................................................... 218

17.2 SIP-based Media Recording .......................................................................... 219
   17.2.1 Enabling SIP-based Media Recording ....................................................... 224
   17.2.2 Configuring SIP Recording Rules ............................................................. 224
   17.2.3 Configuring SIP User Part for SRS ............................................................ 226
   17.2.4 Interworking SIP-based Media Recording with Third-Party Vendors ......... 226
      17.2.4.1 Genesys ............................................................................................. 227
      17.2.4.2 Avaya UCID ...................................................................................... 227

17.3 RADIUS-based Services ............................................................................. 227
   17.3.1 Enabling RADIUS Services ...................................................................... 227
   17.3.2 Configuring RADIUS Servers ................................................................. 228
17.4 LDAP-based Management and SIP Services ................................................. 235
17.4.1 Enabling the LDAP Service ................................................................. 237
17.4.2 Enabling LDAP-based Web/CLI User Login Authentication and Authorization ........................... 237
17.4.3 Configuring LDAP Server Groups .......................................................... 238
17.4.4 Configuring LDAP Servers .................................................................... 240
17.4.5 Configuring LDAP DNs (Base Paths) per LDAP Server ......................... 243
17.4.6 Configuring the LDAP Search Filter Attribute ..................................... 244
17.4.7 Configuring Access Level per Management Groups Attributes ............... 245
17.4.8 Configuring the Device's LDAP Cache ................................................. 247
  17.4.8.1 Refreshing the LDAP Cache .............................................................. 249
  17.4.8.2 Clearing the LDAP Cache ................................................................. 251
17.4.9 Configuring Local Database for Management User Authentication .......... 251
17.4.10 LDAP-based Login Authentication Example ......................................... 252
17.4.11 Enabling LDAP Searches for Numbers with Characters ....................... 257
17.4.12 AD-based Routing for Microsoft Skype for Business ............................ 258
  17.4.12.1 Querying the AD and Routing Priority ............................................ 258
  17.4.12.2 Configuring AD-Based Routing Rules ............................................ 261
17.5 Least Cost Routing .................................................................................... 263
17.5.1 Overview ......................................................................................... 263
17.5.2 Configuring LCR ............................................................................... 265
  17.5.2.1 Configuring Cost Groups ................................................................. 265
  17.5.2.2 Assigning Cost Groups to Routing Rules ........................................ 268
17.6 Remote Web Services ................................................................................ 268
17.6.1 Configuring Remote Web Services ....................................................... 268
  17.6.1.1 Configuring Remote HTTP Hosts ..................................................... 273
17.6.2 Enabling Topology Status Services ....................................................... 275
17.6.3 Centralized Third-Party Routing Server ............................................ 275
17.6.4 Configuring QoS-Based Routing by Routing Server ............................ 278
17.7 HTTP-based Proxy Services ..................................................................... 279
17.7.1 Enabling the HTTP Proxy Application ............................................... 280
17.7.2 Debugging Remote HTTP Services ...................................................... 281
17.7.3 Configuring HTTP Interfaces ............................................................... 281
17.7.4 Configuring HTTP Proxy Services ....................................................... 283
  17.7.4.1 Configuring HTTP Proxy Hosts .................................................... 284
17.7.5 Configuring an HTTP-based EMS Service ........................................... 286
17.8 E9-1-1 Support for Microsoft Skype for Business ...................................... 288
17.8.1 About E9-1-1 Services ........................................................................ 288
17.8.2 Microsoft Skype for Business and E9-1-1 .......................................... 289
  17.8.2.1 Gathering Location Information of Skype for Business Clients for 911 Calls .................. 289
  17.8.2.2 Adding ELINs to the Location Information Server .......................... 291
  17.8.2.3 Passing Location Information to the PSTN Emergency Provider ...... 292
17.8.3 AudioCodes ELIN Device for Skype for Business E9-1-1 Calls to PSTN .......... 293
  17.8.3.1 Detecting and Handling E9-1-1 Calls ............................................ 294
  17.8.3.2 Pre-empting Existing Calls for E9-1-1 Calls ................................ 296
SIP Message Manipulation ............................................................................. 393
SIP Definitions ................................................................................................. 375
Control Network .............................................................................................. 319
Quality of Experience ...................................................................................... 305
Reporting Voice Quality of Experience to SEM .............................................. 305
Configuring the SEM Server .......................................................................... 305
Configuring Clock Synchronization between Device and SEM ...................... 306
Enabling RTCP XR Reporting to SEM ............................................................ 306
Configuring Quality of Experience Profiles ................................................... 307
Configuring Bandwidth Profiles ..................................................................... 312
Configuring Quality of Service Rules ............................................................. 316
Building and Viewing SIP Entities in Topology View ...................................... 369
Configuring Media Realms ............................................................................ 319
Configuring Remote Media Subnets ............................................................... 322
Configuring Media Realm Extensions ............................................................. 325
Configuring SRDs ........................................................................................... 327
Filtering Tables in Web Interface by SRD ....................................................... 333
Multiple SRDs for Multi-tenant Deployments ................................................. 333
Cloning SRDs .................................................................................................. 336
Color-Coding of SRDs in Web Interface .......................................................... 337
Automatic Configuration based on SRD .......................................................... 337
Configuring SIP Interfaces ............................................................................ 337
Configuring IP Groups .................................................................................... 345
Configuring Proxy Sets ................................................................................... 359
Building and Viewing SIP Entities in Topology View ...................................... 369
Configuring Registration Accounts ............................................................... 375
Regular Registration Mode ............................................................................ 380
Single Registration for Multiple Phone Numbers using GIN ......................... 380
Registrar Stickiness ......................................................................................... 381
Configuring Proxy and Registration Parameters ........................................... 382
SIP Message Authentication Example ........................................................... 382
Configuring Call Setup Rules ........................................................................ 384
Call Setup Rule Examples ............................................................................... 390
Configuring SIP Message Manipulation ........................................................ 393
Configuring Message Condition Rules ........................................................... 398
Configuring SIP Message Policy Rules .......................................................... 400
Configuring Pre-Parsing Manipulation Rules .................................................. 403
22 Coders and Profiles ........................................................................................................... 407
  22.1 Configuring Coder Groups .......................................................................................... 407
    22.1.1 Supported Audio Coders ....................................................................................... 410
    22.1.2 Configuring Various Codec Attributes ............................................................. 411
  22.2 Configuring Allowed Audio Coder Groups ................................................................. 412
  22.3 Configuring Allowed Video Coder Groups ................................................................. 415
  22.4 Configuring IP Profiles ............................................................................................. 417

23 SBC Overview ................................................................................................................. 453
  23.1 Feature List ................................................................................................................ 453
  23.2 B2BUA and Stateful Proxy Operating Modes ........................................................... 454
  23.3 Call Processing of SIP Dialog Requests .................................................................... 457
  23.4 User Registration ....................................................................................................... 459
    23.4.1 Initial Registration Request Processing ............................................................. 459
    23.4.2 Classification and Routing of Registered Users .................................................. 460
    23.4.3 General Registration Request Processing ....................................................... 461
    23.4.4 Registration Refreshes ........................................................................................ 461
    23.4.5 Registration Restriction Control ....................................................................... 462
    23.4.6 Deleting Registered Users ............................................................................... 462
  23.5 Media Handling ......................................................................................................... 462
    23.5.1 Media Anchoring ............................................................................................... 463
    23.5.2 Direct Media ....................................................................................................... 464
    23.5.3 Restricting Audio Coders .................................................................................... 466
    23.5.4 Coder Transcoding ............................................................................................. 467
    23.5.5 Transcoding Mode ............................................................................................. 470
    23.5.6 Prioritizing Coder List in SDP Offer .................................................................. 470
    23.5.7 SRTP-RTP and SRTP-SRTP Transcoding .......................................................... 471
    23.5.8 Multiple RTP Media Streams per Call Session ................................................. 471
    23.5.9 Interworking Miscellaneous Media Handling ..................................................... 472
      23.5.9.1 Interworking DTMF Methods ....................................................................... 472
      23.5.9.2 Interworking RTP Redundancy .................................................................... 472
      23.5.9.3 Interworking RTP-RTCP Multiplexing .......................................................... 472
      23.5.9.4 Interworking RTCP Attribute in SDP ............................................................. 472
      23.5.9.5 Interworking Crypto Lifetime Field ............................................................. 473
      23.5.9.6 Interworking Media Security Protocols ....................................................... 473
      23.5.9.7 Interworking ICE Lite for NAT Traversal ..................................................... 473
  23.6 Fax Negotiation and Transcoding ............................................................................. 473
  23.7 Limiting SBC Call Duration ....................................................................................... 474
  23.8 SBC Authentication ................................................................................................... 474
    23.8.1 SIP Authentication Server Functionality ............................................................ 474
    23.8.2 User Authentication based on RADIUS ............................................................. 475
  23.9 Interworking SIP Signaling ......................................................................................... 475
    23.9.1 Interworking SIP 3xx Redirect Responses ........................................................ 476
      23.9.1.1 Resultant INVITE Traversing Device .......................................................... 476
      23.9.1.2 Local Handling of SIP 3xx ........................................................................... 477
    23.9.2 Interworking SIP Diversion and History-Info Headers ..................................... 478
    23.9.3 Interworking SIP REFER Messages ................................................................. 478
    23.9.4 Interworking SIP PRACK Messages ................................................................. 479
    23.9.5 Interworking SIP Session Timer ......................................................................... 479
23.9.6 Interworking SIP Early Media ................................................................. 479
23.9.7 Interworking SIP re-INVITE Messages .............................................. 482
23.9.8 Interworking SIP UPDATE Messages .................................................. 482
23.9.9 Interworking SIP re-INVITE to UPDATE .......................................... 483
23.9.10 Interworking Delayed Offer ............................................................... 483
23.9.11 Interworking Call Hold ....................................................................... 483
23.9.12 Interworking SIP Via Headers ............................................................ 483
23.9.13 Interworking SIP User-Agent Headers .............................................. 484
23.9.14 Interworking SIP Record-Route Headers .......................................... 484
23.9.15 Interworking SIP To-Header Tags in Multiple SDP Answers .......... 484
23.9.16 Interworking In-dialog SIP Contact and Record-Route Headers ...... 484

24 Enabling the SBC Application ................................................................. 485

25 Configuring General SBC Settings .......................................................... 487
  25.1 Interworking Dialog Information in SIP NOTIFY Messages ................. 487

26 Configuring Admission Control ............................................................... 489

27 Routing SBC .......................................................................................... 493
  27.1 Configuring Classification Rules ......................................................... 493
    27.1.1 Classification Based on URI of Selected Header Example .............. 500
  27.2 Configuring SBC IP-to-IP Routing ....................................................... 501
    27.2.1 Configuring Rerouting of Calls to Fax Destinations .................... 513
    27.2.2 Configuring Specific UDP Ports using Tag-based Routing .......... 515
  27.3 Configuring SIP Response Codes for Alternative Routing Reasons ... 519
  27.4 Configuring SBC Routing Policy Rules .............................................. 521
  27.5 Configuring IP Group Sets ................................................................. 525

28 SBC Manipulations ................................................................................. 529
  28.1 Configuring IP-to-IP Inbound Manipulations ..................................... 531
  28.2 Configuring IP-to-IP Outbound Manipulations .................................. 535

29 Configuring Dial Plans ........................................................................... 541
  29.1 Importing and Exporting Dial Plans .................................................... 546
  29.2 Creating Dial Plan Files ..................................................................... 549
  29.3 Using Dial Plan Tags for IP-to-IP Routing .......................................... 550
    29.3.1 Using Dial Plan Tags for Matching Routing Rules ....................... 550
    29.3.2 Using Dial Plan Tags for Routing Destinations ....................... 551
    29.3.3 Dial Plan Backward Compatibility ........................................... 554
  29.4 Using Dial Plan Tags for Outbound Manipulation ............................. 556
  29.5 Using Dial Plan Tags for Call Setup Rules ....................................... 557
  29.6 Using Dial Plan Tags for Message Manipulation .............................. 558

30 Configuring Malicious Signatures ........................................................... 559

31 Advanced SBC Features ........................................................................ 561
  31.1 Configuring Call Preemption for SBC Emergency Calls ................... 561
  31.2 Emergency Call Routing using LDAP to Obtain ELIN ....................... 562
  31.3 Enabling Interworking of SIP and SIP-I Endpoints .......................... 564
  31.4 WebRTC ............................................................................................ 566
    31.4.1 SIP over WebSocket ................................................................. 568
    31.4.2 Configuring WebRTC ............................................................. 570
31.5 Call Forking .......................................................................................................... 573
  31.5.1 Initiating SIP Call Forking ............................................................................. 573
  31.5.2 Configuring SIP Forking Initiated by SIP Proxy ........................................... 573
  31.5.3 Configuring Call Forking-based IP-to-IP Routing Rules ............................... 574
31.6 Call Survivability ............................................................................................... 574
  31.6.1 Enabling Auto-Provisioning of Subscriber-Specific Information of BroadWorks Server for Survivability ...................................................................................... 574
  31.6.2 Configuring BroadSoft's Shared Phone Line Call Appearance for Survivability ........................................................................................................ 575
  31.6.3 Configuring Call Survivability for Call Centers ............................................. 577
  31.6.4 Enabling Survivability Display on Aastra IP Phones ..................................... 579
31.7 Alternative Routing on Detection of Failed SIP Response ................................ 580
31.8 VoIPerfect ......................................................................................................... 580

Cloud Resilience Package ....................................................................................... 585
32 CRP Overview .................................................................................................... 587
33 CRP Configuration ............................................................................................. 589
  33.1 Enabling the CRP Application ......................................................................... 589
  33.2 Configuring Call Survivability Mode ............................................................... 589
  33.3 Pre-Configured IP Groups .................................................................................. 591
  33.4 Pre-Configured IP-to-IP Routing Rules ............................................................ 592
    33.4.1 Normal Mode .............................................................................................. 592
    33.4.2 Emergency Mode ...................................................................................... 593
    33.4.3 Auto Answer to Registrations .................................................................... 593
  33.5 Configuring PSTN Fallback ............................................................................. 594

High Availability System ....................................................................................... 595
34 HA Overview .................................................................................................... 597
  34.1 Connectivity and Synchronization between Devices ...................................... 598
  34.2 Device Switchover upon Failure ................................................................... 599
  34.3 Viewing HA Status on Monitor Web Page ..................................................... 600
35 HA Configuration ............................................................................................. 603
  35.1 Initial HA Configuration ................................................................................ 603
    35.1.1 Network Topology Types and Rx/Tx Ethernet Port Group Settings ............. 603
    35.1.2 Configuring the HA Devices ..................................................................... 605
      35.1.2.1 Step 1: Configure the First Device ....................................................... 606
      35.1.2.2 Step 2: Configure the Second Device ............................................... 607
      35.1.2.3 Step 3: Initialize HA on the Devices .................................................. 608
  35.2 Configuration while HA is Operational ......................................................... 609
  35.3 Configuring Firewall Allowed Rules ............................................................. 610
  35.4 Monitoring IP Entity and HA Switchover upon Ping Failure ......................... 611
36 HA Maintenance ............................................................................................... 613
  36.1 Maintenance of Redundant Device ............................................................... 613
  36.2 Replacing a Failed Device ............................................................................. 613
  36.3 Initiating an HA Switchover .......................................................................... 613
36.4 Reseating the Redundant Unit ........................................................................ 614
36.5 Software Upgrade ......................................................................................... 614
36.6 Rescue Options ............................................................................................. 615
  36.6.1 Taking a Snapshot .................................................................................... 615
  36.6.2 Viewing Available Snapshots ................................................................. 615
  36.6.3 Changing the Default Snapshot .............................................................. 616
  36.6.4 Deleting a Snapshot ................................................................................. 616
  36.6.5 Manual Recovery .................................................................................... 616
    36.6.5.1 Returning to the Default Snapshot ..................................................... 616
    36.6.5.2 Fixing the Current Installation .......................................................... 617
    36.6.5.3 Returning to an Arbitrary Snapshot .................................................. 617
    36.6.5.4 Returning to a Factory Snapshot ....................................................... 617
  36.6.6 Automatic Recovery ................................................................................. 618
36.7 Disconnecting and Reconnecting HA ........................................................... 618

Media Transcoding Cluster ............................................................................... 619

37 Media Transcoding Cluster Overview ......................................................... 621
  37.1 Network Topology and Cluster Interfaces .................................................. 622
  37.2 Prerequisites ............................................................................................... 624

38 Configuring Media Transcoding Cluster ...................................................... 625
  38.1 Initial Cluster Configuration ........................................................................ 625
    38.1.1 Enabling the Cluster Manager .............................................................. 625
    38.1.2 Adding Media Transcoders ................................................................. 625
    38.1.3 Configuring Cluster Redundancy Mode .............................................. 625
    38.1.4 Configuring Maximum Bandwidth for Cluster Interfaces ..................... 626
  38.2 Adding a Media Transcoder ........................................................................ 626
    38.2.1 Step 1: Configure Device with Cluster Network Interface ..................... 626
    38.2.2 Step 2: Initial Configuration of Media Transcoder ............................... 628
      38.2.2.1 Changing OAMP Address for MT .................................................. 628
      38.2.2.2 Changing OAMP Address of vMT .................................................. 629
    38.2.3 Step 3: Configure Media Transcoder with Cluster Interface ................ 630
    38.2.4 Step 4: Cable Device to Media Transcoder .......................................... 630
    38.2.5 Step 5: Configure Media Transcoder with Device's Cluster Address .... 631
    38.2.6 Step 6: Adding Media Transcoder to Cluster Manager ....................... 631

39 Media Transcoding Cluster Maintenance ..................................................... 633
  39.1 Configuring Graceful Timeout .................................................................... 633
  39.2 Actions on Media Transcoders .................................................................. 633
    39.2.1 Locking and Unlocking Media Transcoders ........................................ 633
    39.2.2 Resetting Media Transcoders ............................................................... 634
    39.2.3 Deleting Media Transcoders ................................................................ 634
    39.2.4 Changing the Name of Media Transcoders ....................................... 635
  39.3 Changing OAMP Address of Media Transcoders ....................................... 635
  39.4 Upgrading the Cluster Software Version ................................................... 635
  39.5 Loading Auxiliary Files ............................................................................. 638

40 Media Transcoding Cluster Status and Monitoring ....................................... 639
  40.1 Viewing Cluster Activity Logged Messages .............................................. 639
  40.2 Viewing Overall Status of Media Transcoding Cluster ............................. 640
  40.3 Viewing Status per Media Transcoder on Cluster Manager ........................ 641
40.4 Viewing Media Transcoder Status ................................................................. 645

Maintenance ........................................................................................................... 647

41 Basic Maintenance .......................................................................................... 649
  41.1 Resetting the Device .................................................................................... 649
  41.2 Remotely Resetting Device using SIP NOTIFY ............................................. 650
  41.3 Locking and Unlocking the Device ................................................................. 651
  41.4 Saving Configuration .................................................................................... 652

42 Channel Maintenance ..................................................................................... 653
  42.1 Disconnecting Active Calls .......................................................................... 653

43 Auxiliary Files .................................................................................................. 655
  43.1 Loading Auxiliary Files ................................................................................ 655
    43.1.1 Loading Auxiliary Files through Web Interface ........................................ 656
    43.1.2 Loading Auxiliary Files through CLI ..................................................... 657
  43.2 Deleting Auxiliary Files ................................................................................ 657
  43.3 Call Progress Tones File ............................................................................... 657
  43.4 Prerecorded Tones File ................................................................................ 660
  43.5 Dial Plan File ................................................................................................ 661
    43.5.1 Creating a Dial Plan File .......................................................................... 661
    43.5.2 Obtaining IP Destination from Dial Plan File ............................................ 662
    43.5.3 Viewing Information of Installed Dial Plan File ....................................... 662
  43.6 User Information File ................................................................................... 663
    43.6.1 Enabling the User Info Table ................................................................... 663
    43.6.2 User Information File for SBC User Database ........................................ 663
      43.6.2.1 Configuring SBC User Info Table through Web Interface ............... 664
      43.6.2.2 Configuring SBC User Info Table through CLI ................................ 665
      43.6.2.3 Configuring SBC User Info Table in Loadable Text File ................... 666
    43.6.3 Viewing the Installed User Info File Name .............................................. 667
  43.7 AMD Sensitivity File .................................................................................... 667

44 License Key ...................................................................................................... 669
  44.1 Viewing the License Key ............................................................................... 669
  44.2 Free Product Evaluation for Mediant VE .................................................... 670
  44.3 Obtaining License Key for Initial Activation ............................................... 671
  44.4 Installing a New License Key ........................................................................ 672
    44.4.1 Installing License Key through Web Interface ......................................... 673
      44.4.1.1 Installing a License Key String ........................................................... 673
      44.4.1.2 Installing a License Key File ............................................................... 675
    44.4.2 Installing License Key through CLI ....................................................... 679
    44.4.3 Verifying Installed License Key ............................................................... 679
  44.5 Upgrading SBC Capacity Licenses by License Pool Manager Server .......... 679
  44.6 Backing up the License Key .......................................................................... 681
  44.7 Viewing the Device's Product Key ................................................................. 681

45 Software Upgrade Wizard .............................................................................. 683
  45.1 Post-Upgrade from Version 7.0 Procedure .................................................. 688
46 Configuration File ........................................................................................................ 691
   46.1 Saving Configuration to a File ............................................................................. 691
   46.2 Loading a Configuration File .............................................................................. 691
47 Automatic Provisioning ............................................................................................. 693
   47.1 Automatic Configuration Methods ........................................................................ 693
      47.1.1 DHCP-based Provisioning ................................................................. 693
      47.1.2 HTTP-based Provisioning ................................................................. 694
      47.1.3 FTP-based Provisioning .................................................................... 695
      47.1.4 Provisioning using AudioCodes EMS ................................................... 695
   47.2 HTTP/S-Based Provisioning using the Automatic Update Feature ..................... 695
      47.2.1 Files Provisioned by Automatic Update .................................................. 696
      47.2.2 File Location for Automatic Update ...................................................... 697
      47.2.3 MAC Address Placeholder in Configuration File Name ......................... 697
      47.2.4 File Template for Automatic Provisioning .............................................. 698
      47.2.5 Triggers for Automatic Update ............................................................... 699
      47.2.6 Access Authentication with HTTP Server .............................................. 700
      47.2.7 Querying Provisioning Server for Updated Files ................................... 701
      47.2.8 File Download Sequence ...................................................................... 703
      47.2.9 Cyclic Redundancy Check on Downloaded Configuration Files ............. 704
      47.2.10 Automatic Update Configuration Examples ......................................... 705
         47.2.10.1 Automatic Update for Single Device ........................................... 705
         47.2.10.2 Automatic Update from Remote Servers .................................... 706
         47.2.10.3 Automatic Update for Mass Deployment .................................... 707
48 SBC Configuration Wizard ...................................................................................... 711
   48.1 Starting the SBC Configuration Wizard ........................................................... 712
   48.2 General Setup Page .......................................................................................... 713
   48.3 System Page ...................................................................................................... 715
   48.4 Interfaces Page .................................................................................................. 716
   48.5 IP-PBX Page ...................................................................................................... 717
   48.6 SIP Trunk Page .................................................................................................. 719
   48.7 Number Manipulation Page .............................................................................. 721
   48.8 Remote Users Page .......................................................................................... 722
   48.9 Summary Page .................................................................................................. 723
   48.10 Congratulations Page ..................................................................................... 724
49 Restoring Factory Defaults ...................................................................................... 725
   49.1 Restoring Factory Defaults through CLI ........................................................... 725
   49.2 Restoring Factory Defaults through Web Interface .......................................... 725
   49.3 Restoring Defaults through ini File ................................................................... 726

Status, Performance Monitoring and Reporting ........................................................ 727

50 System Status .......................................................................................................... 729
   50.1 Viewing Device Information ............................................................................ 729
   50.2 Viewing Device Status on Monitor Page ......................................................... 730
51 Reporting DSP Utilization through SNMP MIB .............................................. 733
52 Viewing Carrier-Grade Alarms ....................................................................... 735
  52.1 Viewing Active Alarms .............................................................................. 735
  52.2 Viewing History Alarms ........................................................................... 736
53 Viewing Management User Activity Logs ..................................................... 739
54 Viewing Performance Monitoring .................................................................. 741
  54.1 Viewing Call Success and Failure Ratio ...................................................... 741
  54.2 Viewing Average Call Duration ................................................................ 742
  54.3 Configuring Performance Profiles .............................................................. 743
55 Viewing VoIP Status ..................................................................................... 749
  55.1 Viewing SBC Registered Users ................................................................. 749
  55.2 Viewing Proxy Set Status ......................................................................... 750
  55.3 Viewing Registration Status ....................................................................... 751
  55.4 Viewing CDR Test Calls ........................................................................... 752
  55.5 Viewing SBC CDR History ....................................................................... 753
56 Viewing Network Status ................................................................................ 755
  56.1 Viewing Active IP Interfaces ..................................................................... 755
  56.2 Viewing Ethernet Device Status ................................................................. 755
  56.3 Viewing Ethernet Port Information ............................................................. 755
  56.4 Viewing Static Routes Status ...................................................................... 756
57 Reporting Information to External Party ....................................................... 757
  57.1 Configuring RTCP XR ................................................................................ 757
  57.2 Generating Call Detail Records .................................................................. 762
    57.2.1 CDR Field Description ......................................................................... 762
    57.2.1.1 CDR Fields for SBC Signaling ......................................................... 762
    57.2.1.2 CDR Fields for SBC Media .............................................................. 767
    57.2.1.3 CDR Fields for SBC Local Storage .................................................. 769
    57.2.2 Customizing CDRs for SBC Calls ....................................................... 770
    57.2.3 Configuring CDR Reporting ................................................................. 774
    57.2.4 Storing CDRs on the Device ................................................................. 775
  57.3 Configuring RADIUS Accounting ............................................................. 777

Diagnostics ............................................................................................................785
58 Syslog and Debug Recording ........................................................................ 787
  58.1 Configuring Log Filter Rules ...................................................................... 787
    58.1.1 Filtering IP Network Traces ................................................................. 791
  58.2 Configuring Syslog ..................................................................................... 792
    58.2.1 Syslog Message Format ........................................................................ 792
      58.2.1.1 Event Representation in Syslog Messages ...................................... 792
      58.2.1.2 Identifying AudioCodes Syslog Messages using Facility Levels ...... 796
      58.2.1.3 Syslog Fields for Answering Machine Detection (AMD) ............... 796
      58.2.1.4 SNMP Alarms in Syslog Messages ............................................... 797
    58.2.2 Enabling Syslog .................................................................................... 797
Appendix ................................................................................................................825

65 Configuration Parameters Reference .................................................................831

65.1 Management Parameters ...................................................................................831
  65.1.1 General Parameters .....................................................................................831
  65.1.2 Web Parameters ..........................................................................................832
  65.1.3 Telnet Parameters .......................................................................................837
  65.1.4 ini File Parameters .......................................................................................838
  65.1.5 SNMP Parameters .......................................................................................838
  65.1.6 Serial Parameters .......................................................................................842
  65.1.7 Auxiliary and Configuration File Name Parameters ..................................843
  65.1.8 Automatic Update Parameters ..................................................................844

65.2 Networking Parameters .....................................................................................848
  65.2.1 Ethernet Parameters ...................................................................................848
  65.2.2 Multiple VoIP Network Interfaces and VLAN Parameters .......................849
  65.2.3 Routing Parameters ...................................................................................850
  65.2.4 Quality of Service Parameters ...................................................................850
  65.2.5 NAT Parameters .......................................................................................851
  65.2.6 DNS Parameters .......................................................................................853
  65.2.7 DHCP Parameters .....................................................................................853
  65.2.8 NTP and Daylight Saving Time Parameters .............................................855

65.3 Debugging and Diagnostics Parameters ...........................................................857
  65.3.1 General Parameters ...................................................................................857
  65.3.2 SIP Test Call Parameters ...........................................................................857
  65.3.3 Syslog, CDR and Debug Parameters ..........................................................858
  65.3.4 Resource Allocation Indication Parameters .............................................864

65.4 HA Parameters ..................................................................................................864
65.5 Security Parameters........................................................................................................ 866
    65.5.1 General Security Parameters.............................................................................. 866
    65.5.2 HTTPS Parameters............................................................................................ 868
    65.5.3 SRTP Parameters................................................................................................ 869
    65.5.4 TLS Parameters............................................................................................... 871
    65.5.5 SSH Parameters............................................................................................... 873
    65.5.6 IDS Parameters............................................................................................... 874
    65.5.7 OCSP Parameters............................................................................................ 875
65.6 Media Transcoding Cluster Parameter ........................................................................... 876
65.7 Quality of Experience Parameters ................................................................................ 876
65.8 Control Network Parameters.......................................................................................... 879
    65.8.1 IP Group, Proxy, Registration and Authentication Parameters........................... 879
    65.8.2 Network Application Parameters....................................................................... 887
65.9 General SIP Parameters................................................................................................. 889
65.10 Coders and Profile Parameters..................................................................................... 906
65.11 Channel Parameters.................................................................................................... 908
    65.11.1 Voice Parameters............................................................................................ 908
    65.11.2 Coder Parameters.......................................................................................... 910
    65.11.3 DTMF Parameters.......................................................................................... 911
    65.11.4 RTP, RTCP and T.38 Parameters...................................................................... 912
65.12 SBC Parameters........................................................................................................... 916
    65.12.1 Supplementary Services................................................................................ 934
65.13 IP Media Parameters.................................................................................................... 935
65.14 Services....................................................................................................................... 939
    65.14.1 SIP-based Media Recording Parameters.......................................................... 939
    65.14.2 RADIUS and LDAP Parameters...................................................................... 940
        65.14.2.1 General Parameters............................................................................. 940
        65.14.2.2 RADIUS Parameters......................................................................... 940
        65.14.2.3 LDAP Parameters............................................................................. 942
    65.14.3 Least Cost Routing Parameters....................................................................... 945
    65.14.4 Call Setup Rules Parameters.......................................................................... 946
    65.14.5 HTTP-based Services..................................................................................... 946
    65.14.6 HTTP Proxy Parameters................................................................................ 947
66 Channel Capacity.............................................................................................................. 951
66.1 Mediant VE SBC............................................................................................................. 954
    66.1.1 Mediant VE SBC for OpenStack and VMware Hypervisors............................... 954
        66.1.1.1 2-vCPU Mediant VE SBC..................................................................... 954
        66.1.1.2 4-vCPU Mediant VE SBC..................................................................... 955
        66.1.1.3 8-vCPU Mediant VE SBC..................................................................... 957
    66.1.2 Amazon AWS EC2.............................................................................................. 958
    66.1.3 Mediant VE SBC for Hyper-V Hypervisor.......................................................... 959
        66.1.3.1 2-vCPU Mediant VE SBC..................................................................... 959
        66.1.3.2 4-vCPU Mediant VE SBC..................................................................... 961
    66.1.4 Mediant VE SBC with Media Transcoders........................................................ 962
66.2 Mediant SE SBC............................................................................................................ 964
67 Technical Specifications.................................................................................................... 965
Notice

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

This document is subject to change without notice.

Date Published: August-14-2017

WEEE EU Directive

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

Customer Support

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

Abbreviations and Terminology

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

Related Documentation

<table>
<thead>
<tr>
<th>Manual Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP Release Notes</td>
</tr>
<tr>
<td><strong>Complementary Guides</strong></td>
</tr>
<tr>
<td>CLI Reference Guide</td>
</tr>
<tr>
<td>SNMP User’s Guide</td>
</tr>
<tr>
<td>SBC Design Guide</td>
</tr>
<tr>
<td>Recommended Security Guidelines Configuration Note</td>
</tr>
</tbody>
</table>
Notes and Warnings

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

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

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

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

Note:
- This device includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit ([http://www.openssl.org/](http://www.openssl.org/)).
- This device includes cryptographic software written by Eric Young ([eay@cryptsoft.com](mailto:eay@cryptsoft.com)).

Note: Some of the features listed in this document are available only if the relevant License Key has been purchased from AudioCodes and installed on the device. For a list of 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.

Document Revision Record

<table>
<thead>
<tr>
<th>LRTT</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>41863</td>
<td>Initial document release for Version 7.2.</td>
</tr>
<tr>
<td>41864</td>
<td>Updated for patch version 7.20A.001.</td>
</tr>
<tr>
<td></td>
<td>Updated sections: Licensing the Device; First Incoming Packet Mechanism; Configuring the Device's LDAP Cache; Centralized Third-Party Routing Server; Configuring the SEM Server (port removed); Configuring Call Setup Rules (Dial Plan queries); Call Setup Rule Examples; Registration Refreshes; Using Dial Plan Tags for IP-to-IP Routing; Enabling Interworking of SIP and SIP-I Endpoints (SPIROU and SIP header X-AC-Action); Computer Requirements (supported browsers); Configuring BroadSoft's Shared Phone Line Call Appearance for Survivability (example); Configuring RTCP XR (IP Group); Enabling Same Call Session ID over Multiple Devices (removed).</td>
</tr>
<tr>
<td></td>
<td>New sections: Using Dial Plan Tags for Message Manipulation; Using Dial Plan Tags for Message Manipulation; VoIPerfect; Configuring Test Call Endpoints.</td>
</tr>
<tr>
<td></td>
<td>New parameters: WebLoginBlockAutoComplete; EnforcePasswordComplexity; IPGroup_SBCKeepOriginalCallID; IPGroup_SBCDialPlanName; IPGroup_CallSetupRulesSetId; CallSetupRules_QueryType; CallSetupRules_QueryTarget; IpProfile_SBCVoiceQualityEnhancement; IpProfile_SBCMaxOpusBW; IpProfile_SBCISUPVariant; PublicationIPGroupID; GeneratedRegistersInterval.</td>
</tr>
<tr>
<td></td>
<td>Updated parameters: InterfaceTable_InterfaceName; Web password; IPGroup_MediaRealm (Web name); CallSetupRules_AttributesToQuery (Web name and description); CallSetupRules_ActionValue; CallSetupRules_ActionSubject; CallSetupRules_Condition; CallSetupRules_AttributesToQuery; IpProfile_SBCRTCPFeedback (values); IpProfile_MediaIPVersionPreference; ConditionTable_Name; Test_Call_RouteBy (default); Test_Call_SIPInterfaceName; CLIPvPrivPass; NATMode (values); SendAcSessionIDHeader (removed); InboundMediaLatchMode (Note); QOEPort (removed); MaxGeneratedRegistersRate; RTPOnlyMode (removed); SBCUserRegistrationGraceTime; SBCKeepOriginalCallId.</td>
</tr>
<tr>
<td></td>
<td>Channel Capacity: Hyper-V 1/2/4 vCPU 4 GB RAM for Mediant VE low-capacity; AWS EC2 for Mediant VE low-capacity; VMware for Mediant VE high-capacity; KVM 8 vCPU 16 GB RAM SR-IOV Intel NICs for Mediant VE high-capacity.</td>
</tr>
<tr>
<td></td>
<td>Miscellaneous: Allowed access to CLI re user levels; configuring media latching.</td>
</tr>
<tr>
<td>41866</td>
<td>Updated sections: Changing Index Position of Table Rows; Searching for</td>
</tr>
<tr>
<td>LTRT</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>Configuration Parameters; Configuring TLS Certificate Contexts (IPSec removed); Enabling the HTTP Proxy Application (license); Direct Media; Configuring SBC IP-to-IP Routing (IP Group load balancing); MAC Address Placeholder in Configuration File Name; VoIPerfect; Channel Capacity.</td>
</tr>
<tr>
<td></td>
<td>New sections: Configuring IP Group Sets.</td>
</tr>
<tr>
<td></td>
<td>Updated parameters: SIPInterface_SBCDirectMedia; IPProfile_SBCDirectMediaTag; IpProfile_DisconnectOnBrokenConnection; IP2IPRouting_DestType; IPOutboundManipulation_PrivacyRestrictionMode; BrokenConnectionEventTimeout.</td>
</tr>
<tr>
<td></td>
<td>New parameters: IP2IPRouting_IPGroupSetName; EnableNonCallCdr; PGroupSet; IPGroupSetMember; NoRTPDetectionTimeout; DisconnectOnBrokenConnection; BrokenConnectionEventTimeout.</td>
</tr>
<tr>
<td></td>
<td>Patch version 7.20A.100.</td>
</tr>
<tr>
<td>41869</td>
<td>Updated sections: Areas of the GUI (SBC Wizard); Assigning Rows from Other Tables (search, add new, and view); Invalid Value Indications; Creating a Login Welcome Message; Configuring Management User Accounts (CLI); Enabling SSH with RSA Public Key for CLI (public key); Configuring TLS Certificate Contexts (DTLS); Assigning CSR-based Certificates to TLS Contexts; Generating Private Keys for TLS Contexts; Configuring Underlying Ethernet Devices (max.); Configuring IP Network Interfaces (max.); Configuring Media Realms (max.); SRTP using DTLS Protocol; Building and Viewing your Network Topology; SIP-based Media Recording (multiple SRSs); Enabling SIP-based Media Recording; Configuring SIP Recording Rules; Configuring Proxy Sets (keep-alive); WebRTC (RFCs); Configuring WebRTC; Pre-Configured IP Groups; Normal Mode (CRP); Emergency Mode (CRP); Auto Answer to Registrations (CRP); Network Topology Types and Rx/Tx Ethernet Port Group Settings; Media Transcoding Cluster; License Key; Viewing the License Key; Obtaining License Key for Feature Upgrade (removed); Installing the a New License Key; Installing License Key through Web Interface; Upgrading SBC Capacity Licenses by License Pool Manager Server; Viewing Device Information; Viewing Call Routing Status (removed); Configuring RTCP XR (IP Group); Configuring RADIUS Accounting (typo for Accounting-Request); Automatic Provisioning (Startup CLI Script File); Channel Capacity.</td>
</tr>
<tr>
<td></td>
<td>New sections: Customizing the Web Interface; Replacing the Corporate Logo; Replacing the Corporate Logo with an Image; Replacing the Corporate Logo with Text; Customizing the Product Name; Customizing the Favicon; SRTP using DTLS Protocol; SBC Wizard; Media Transcoding Cluster; Viewing the Device's Product Key; Saving Configuration to a File; Loading a Configuration File; Viewing Proxy Set Status; Configuring Maximum Bandwidth for Cluster Interfaces.</td>
</tr>
<tr>
<td></td>
<td>Updated parameters: TLSContexts_ServerCipherString; TLSContexts_ClientCipherString; NATTranslation_SourceStartPort; NATTranslation_SourceEndPort; NATTranslation_TargetStartPort; NATTranslation_TargetEndPort; SNMPSysOid; SNMPTrapEnterpriseOid; EnableCoreDump (typo); HTTPSClientCipherString (removed); SSHAdminKey; SessionExpiresDisconnectTime; BrokenConnectionEventTimeout; RADIUSRetransmission (default); RadiusTO (default); SIPRecRouting_RecordedIPGroupName; SIPRecRouting_SRSIPGroupName.</td>
</tr>
<tr>
<td></td>
<td>New parameters: WebUsers_SSHPublicKey; TLSContexts_DTLSVersion; TLSContexts_DHKeySize; SIPRecRouting_SRSRedundantIPGroupName; ProxySet_SuccessDetectionRetries; ProxySet_SuccessDetectionInterval; ProxySet_FailureDetectionRetransmissions; ProxySet_MinActiveServersLB; WebUsers; WebFaviconFileUrl; AUDPStartupScriptURL;</td>
</tr>
<tr>
<td>LTRT</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>MtcClusterNetworkMaxBandwidth.</td>
<td></td>
</tr>
<tr>
<td>41871</td>
<td>Updated sections: Changing Default IP Address to Suit your Network Addressing Scheme (CLI); Configuring Management User Accounts (typo); Enabling SNMP; Configuring IP Network Interfaces; SIP-based Media Recording (multiple SRS); Configuring LDAP Servers (max. and cache); Configuring Call Setup Rules; Configuring SBC IP-to-IP Routing (note); Configuring SIP Response Codes for Alternative Routing Reasons; SBC Wizard (screens); Auxiliary Files (SBC Wizard); Viewing IP Connectivity (typo); Creating Core Dump and Debug Files upon Device Crash (reset); Configuring IP Group Sets (max); Media Transcoding Cluster Status and Monitoring; Changing OAMP Address of vMT; Channel Capacity (Mediant SE – SRTP-RTP)</td>
</tr>
<tr>
<td></td>
<td>New sections: Debugging Remote HTTP Services</td>
</tr>
<tr>
<td></td>
<td>Updated parameters: IpProfile_SBCUseSilenceSupp (removed); SIPRecRouting_SRSIPGroupName; SIPInterface_InterfaceName (max. char); ProxySet_ProxyName (max. char); MessageManipulations_ManipulationName (max. char); MessagePolicy_Name (max. char); AllowedAudioCodersGroups_Name (max. char); AllowedVideoCodersGroups_Name (max. char); _ManipulationName (max. char); SBCAdmissionControl_AdmissionControlName (max. char); Classification_ClassificationName (max. char); IP2IPRouting_RouteName (max. char); SBCRoutingPolicy_Name (max. char); IPGroupSet_Name (max. char); IPInboundManipulation_ManipulationName (max. char); IPOutboundManipulation_ManipulationName (max. char); SBCAdmissionControl_Rate; EnableWebAccessFromAllInterfaces; ResetWebPassword; DisableSNMP; EnableCoreDump; SSHMaxLoginAttempts; IgnoreAlertAfterEarlyMedia; ECNLPModes; PremiumServiceClassMediaDiffServ; PremiumServiceClassControlDiffServ; IsFaxUsed; EnableAGC</td>
</tr>
<tr>
<td></td>
<td>New parameters: HTTPProxySyslogDebugLevel</td>
</tr>
<tr>
<td>41874</td>
<td>Updated with patch version 7.2.150.</td>
</tr>
<tr>
<td></td>
<td>Updated sections: Areas of the GUI (Configuration Wizard button); Enabling Disabling SNMP; Viewing Certificate Information (screen); Assigning Externally Created Private Keys to TLS Contexts (pass-phrase); Generating Private Keys for TLS Contexts (pass-phrase); Importing Certificates into Trusted Certificate Store (bulk import); Configuring Underlying Ethernet Devices (MTU); Configuring Firewall Settings (note); SIP-based Media Recording (max.); Configuring Remote Web Services (QoS routing); Returning to an Arbitrary Snapshot; Centralized Third-Party Routing Server (QoS); Configuring Proxy Sets; Configuring SBC IP-to-IP Routing; Configuring IP Group Sets (dialed plan tags); Configuring Dial Plans; Software Upgrade; Installing License Key through Web Interface; Upgrading SBC Capacity Licenses by License Pool Manager Server; Viewing the Device's Product Key; SBC Configuration Wizard; Configuring RADIUS Accounting (typo); Configuring Reporting of Management User Activities (persistent log); Configuring DTMF Tones for Test Calls; Configuring Basic Test Calls; Configuring SBC Test Call with External Proxy (removed); Channel Capacity.</td>
</tr>
<tr>
<td></td>
<td>New sections: Configuring QoS-Based Routing by Routing Server; Microsoft Skype for Business Presence of Third-Party Endpoints; Registrar Stickiness; Configuring Pre-Parsing Manipulation Rules; Configuring Rerouting of Calls to Fax Destinations; Using Dial Plan Tags for Routing Destinations; Disconnecting and Reconnecting HA; Viewing the License Key; Installing a License Key String; Debugging Web Services.</td>
</tr>
<tr>
<td>LTRT</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>- Updated parameters: AccessList_Source_IP; AccessList_Source_Port; AccessList_Start_Port; AccessList_End_Port; HTTPRemoteServices_HTTPType (option 5); IPGroup_SBCDialPlanName (note); ProxySet_IsProxyHotSwap; ProxyIP_IpAddress; IpProfile_SBCRemoteReferBehavior (4); IpProfile_SBCPlayHeldTone; IP2IPRouting_Trigger (6); IP2IPRouting_DestType (12/13); DialPlanRule_Tag; SBCCDRFormat_FieldType (818); Test_Call_RouteBy; Test_Call_Play (tone); KeepAliveTrapPort (default); SBCtestID (removed); ProxyIPListRefreshTime; RegistrationRetryTime (note); EnablePChargingVector (removed); EnableSBCApplication (default).</td>
</tr>
<tr>
<td></td>
<td>- New parameters: DeviceTable_MTU; SRD_SBCDialPlanName; SIPInterface_PreParsingManSetName; IPGroup_Tags; Account_RegistrarStickiness; Account_RegistrarSearchMode; Account_RegEventPackageSubscription; IpProfile_SBCFaxReroutingMode; IP2IPRouting_RoutingTagName; IP2IPRouting_InternalAction; IPGroupSet_Tags; CustomerSN; MaxRegistrationBackoffTime; MaxSDPSessionVersionId; UseRandomUser; UnregisterOnStartup; PresencePublishIPGroupId; EnableMSPresence; PreParsingManipulationSets; PreParsingManipulationRules; MWINotificationTimeout; RoutingServerQualityStatus; RoutingServerQualityStatusRate; SystemConsoleMode.</td>
</tr>
<tr>
<td>41876</td>
<td>- Updated with patch version 7.20A.152.</td>
</tr>
<tr>
<td></td>
<td>- Updated sections: SIP-based Media Recording (max. SIPRec); Configuring the LDAP Search Filter Attribute (Web path); Enabling LDAP Searches for Numbers with Characters; Microsoft Skype for Business Presence of Third-Party Endpoints; Configuring the Device for Skype for Business Presence (example); Configuring Media Realm Extensions; Configuring Firewall Allowed Rules; Returning to an Arbitrary Snapshot; Prerecorded Tones File; Installing on HA Devices (note); Loading a Configuration File (note); Channel Capacity (8 vCPU 16 GB RAM SR-IOV Intel NICs; 8 vCPU 32 GB RAM SR-IOV Intel NICs; 2-vCPU Mediant VE SBC; 4-vCPU Mediant VE SBC; 8-vCPU Mediant VE SBC</td>
</tr>
<tr>
<td></td>
<td>- Updated parameters: SNMPReadOnlyCommunityString (max. char); SNMPReadWriteCommunityString (max. char); SNMPTrapCommunityString (max. char); DeviceTable_VlanID; MediaRealmExtension_IPv4IF; MediaRealmExtension_IPv6IF; ProxySet_EnableProxyKeepAlive; SIPSDPSessionOwner; IP2IPRouting_InternalAction</td>
</tr>
<tr>
<td></td>
<td>- New parameters: IPGroup_SBCUserStickiness; IPProfile_LocalRingbackTone; IPProfile_LocalHeldTone</td>
</tr>
<tr>
<td>41877</td>
<td>- Updated with patch Version 7.20A.154.007</td>
</tr>
<tr>
<td></td>
<td>- Updated sections: Silence Suppression (removed); Fax / Modem Transparent Mode (silence suppression removed); Configuring SIP Recording Rules (view sessions in CLI); Configuring RTP Base UDP Port (note removed re SIP Interface); Configuring IP Groups (max.); Configuring Proxy Sets (max.); Customizing CDRs for SBC Calls (max. RADIUS); Centralized Third-Party Routing Server (call preemption added); Locking and Unlocking the Device (typos); Viewing Active Alarms (max display); Capacity (8-vCPU Mediant VE SBC)</td>
</tr>
<tr>
<td></td>
<td>- New sections: Configuring Additional Management Interfaces; Configuring Specific UDP Ports using Tag-based Routing</td>
</tr>
<tr>
<td></td>
<td>- Updated parameters: WebUsers_Password (note); InterfaceTable_ApplicationTypes; CpMediaRealm_PortRangeStart (note removed); SIPInterface_UDPPort (note removed); ProxySet_SuccessDetectionRetries (max);</td>
</tr>
<tr>
<td>LTRT</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ProxySet_SuccessDetectionInterval (max); Account_RegistrarSearchMode (phys link); AudioCoders_Sce (global parameter removed); IpProfile_SCE (removed); IpProfile_SBCRemoteReferBehavior (new option 5); IPProfile_SBCRemoteHoldFormat (new option 6); SBCCDRFormat_Title (max.); IP2IPRouting_InternalAction; SBCCDRFormat_Title (max. char.); WebUsers (CLI name); EnableWebAccessFromAllInterfaces; FaxBypassPayloadType; ModemBypassPayloadType; EnableSilenceCompression (removed)</td>
<td>▪ New parameters: SIPInterface_AdditionalUDPPorts; IPProfile_SBCSupportMultipleDTMFMethods; AdditionalManagementInterfaces; DefaultTerminalWindowHeight; ActiveAlarmTableMaxSize; SBCRemoveSIPSFromNonSecuredTransport</td>
</tr>
</tbody>
</table>
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 Introduction

This User's Manual describes how to configure and manage your AudioCodes product (hereafter, referred to as device). This document is intended for the professional person responsible for installing, configuring and managing the device.

1.1 Product Overview

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

- **Mediant Server Edition SBC**: x86 server-based platform, which must be installed on a server that complies to the specified hardware requirements (see "Technical Specifications" on page 965 or refer to the Mediant Server Edition SBC Installation Manual)

- **Mediant Virtual Edition SBC**: Installed and hosted in a virtual machine environment that complies to specified requirements (see "Technical Specifications" on page 965 or refer to the Mediant Virtual Edition SBC Installation Manual)

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

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

**Note**: For maximum call capacity figures, see "Channel Capacity" on page 951.

1.2 Typographical Conventions

This document uses the following typographical conventions to convey information:

<table>
<thead>
<tr>
<th>Table 1-1: Typographical Conventions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Convention</strong></td>
</tr>
<tr>
<td>Boldface font</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Text enclosed by double apostrophe &quot;...&quot;</td>
</tr>
<tr>
<td>Courier font</td>
</tr>
</tbody>
</table>
### 1.3 Getting Familiar with Configuration Concepts and Terminology

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

#### 1.3.1 SBC Application

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

The following table describes the main configuration concepts and terminology.

<table>
<thead>
<tr>
<th>Configuration Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Group</td>
<td>The IP Group is a logical representation of the SIP entity (UA) with which the device receives and sends calls. The SIP entity can be a server (e.g., IP PBX or SIP Trunk) or it can be a group of users (e.g., LAN IP phones). For servers, the IP Group is typically used to define the address of the entity (by its associated Proxy Set). IP Groups are used in IP-to-IP routing rules to denote the source and destination of the call.</td>
</tr>
<tr>
<td>Proxy Set</td>
<td>The Proxy Set defines the actual address (IP address or FQDN) of SIP entities that are servers (e.g., IP PBX). As the IP Group represents the SIP entity, to associate an address with the SIP entity, the Proxy Set is assigned to the IP Group. You can assign the same Proxy Set to multiple IP Groups (belonging to the same SRD).</td>
</tr>
<tr>
<td>SIP Interface</td>
<td>The SIP Interface represents a Layer-3 network. It defines a local listening port for SIP signaling traffic on a local, logical IP network interface. The term local implies that it’s a logical port and network interface on the device. The SIP Interface is used to receive and send SIP messages with a specific SIP entity (IP Group). Therefore, you can create a SIP Interface for each SIP entity in the VoIP network with which...</td>
</tr>
</tbody>
</table>
1. Introduction

### Configuration Terms

<table>
<thead>
<tr>
<th>Configuration Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>your device needs to communicate. For example, if your VoIP network consists of three SIP entities -- a SIP Trunk, a LAN IP PBX, and remote WAN users -- a SIP Interface can be created for each of these Layer-3 networks. The SIP Interface is associated with the SIP entity, by assigning it to an SRD that is in turn, assigned to the IP Group of the SIP entity.</td>
<td></td>
</tr>
<tr>
<td><strong>Media Realm</strong></td>
<td>The Media Realm defines a local UDP port range for RTP (media) traffic on any one of the device's logical IP network interfaces. The Media Realm is used to receive and send media traffic with a specific SIP entity (IP Group). The Media Realm can be associated with the SIP entity, by assigning the Media Realm to the IP Group of the SIP entity, or by assigning it to the SIP Interface associated with the SIP entity.</td>
</tr>
<tr>
<td><strong>SRD</strong></td>
<td>The SRD is a logical representation of your entire SIP-based VoIP network (Layer 5) containing groups of SIP users and servers. The SRD is in effect, the foundation of your configuration to which all other previously mentioned configuration entities are associated. For example, if your VoIP network consists of three SIP entities -- a SIP Trunk, a LAN IP PBX, and remote WAN users -- the three SIP Interfaces defining these Layer-3 networks would all assigned to the same SRD. Typically, only a single SRD is required and this is the recommended configuration topology. As the device provides a default SRD, in a single SRD topology, the device automatically assigns the SRD to newly created configuration entities. Thus, in such scenarios, there is no need to get involved with SRD configuration. Multiple SRDs are required only for multi-tenant deployments, where it &quot;splits&quot; the device into multiple logical devices. For multiple SRDs, the SRD can be configured with a Sharing Policy. The Sharing Policy simply means whether the SRD's resources (SIP Interfaces, IP Groups, and Proxy Sets) can be used by other SRDs. For example, if all tenants route calls with the same SIP Trunking service provider, the SRD of the SIP Trunk would be configured as a Shared Sharing Policy. SRDs whose resources are not shared, would be configured with an Isolated Sharing Policy.</td>
</tr>
<tr>
<td><strong>IP Profile</strong></td>
<td>The IP Profile is an optional configuration entity that defines a wide range of call settings for a specific SIP entity (IP Group). The IP Profile includes signaling and media related settings, for example, jitter buffer, voice coders, fax signaling method, SIP header support (local termination if not supported), and media security method. The IP Profile is in effect, the interoperability &quot;machine&quot; of the device, enabling communication between SIP endpoints that &quot;speak&quot; different call &quot;languages&quot;. The IP Profile is associated with the SIP entity, by assigning the IP Profile to the IP Group of the SIP entity.</td>
</tr>
<tr>
<td>Classification</td>
<td>Classification is the process that identifies the incoming call (SIP dialog request) as belonging to a specific SIP entity (IP Group). There are three chronological classification stages, where each stage is done only if the previous stage fails. The device first attempts to classify the SIP dialog by checking if it belongs to a user that is already registered in the device's registration database. If this stage fails, the device checks if the source IP address is defined for a Proxy Set and if yes, it classifies it to the IP Group associated with the Proxy Set. If this fails, the device</td>
</tr>
</tbody>
</table>

Version 7.2 29 Mediant VE/SE SBC
<table>
<thead>
<tr>
<th>Configuration Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration Terms</td>
<td>classifies the SIP dialog using the Classification table, which defines various characteristics of the incoming dialog that if matched, classifies the call to a specific IP Group. The main characteristics of the incoming call is the SIP Interface that is associated with the SRD for which the Classification rule is configured.</td>
</tr>
<tr>
<td>IP-to-IP Routing</td>
<td>IP-to-IP routing rules define the routes for routing calls between SIP entities. As the SIP entities are represented by IP Groups, the routing rules typically employ IP Groups to denote the source and destination of the call. For example, to route calls from the IP PBX to the SIP Trunk, the routing rule can be configured with the IP PBX as the source IP Group and the SIP Trunk as the destination IP Group. Instead of IP Groups, various other source and destination methods can be used. For example, the source can be a source host name while the destination can be an IP address or based on an LDAP query.</td>
</tr>
<tr>
<td>Inbound and Outbound Manipulation</td>
<td>Inbound and Outbound Manipulation lets you manipulate the user part of the SIP URI in the SIP message for a specific entity (IP Group). Inbound manipulation is done on messages received from the SIP entity; outbound manipulation is done on messages sent to the SIP entity. Inbound manipulation lets you manipulate the user part of the SIP URI for source (e.g., in the SIP From header) and destination (e.g., in the Request-URI line) in the incoming SIP dialog request. Outbound manipulation lets you manipulate the user part of the Request-URI for source (e.g., in the SIP From header) or destination (e.g., in the SIP To header) or calling name, in outbound SIP dialog requests. The Inbound and Outbound manipulation are associated with the SIP entity, by configuring the rules with incoming characteristics such as source IP Group and destination host name. The manipulation rules are also assigned a Routing Policy, which in turn, is assigned to IP-to-IP routing rules. As most deployments require only one Routing Policy, the default Routing Policy is automatically assigned to the manipulation rules and to the routing rules.</td>
</tr>
<tr>
<td>Routing Policy</td>
<td>Routing Policy logically groups routing and manipulation (inbound and outbound) rules to a specific SRD. It also enables Least Cost Routing (LCR) for routing rules and associates an LDAP server for LDAP-based routing. However, as multiple Routing Policies are required only for multi-tenant deployments, for most deployments only a single Routing Policy is required. When only a single Routing Policy is required, handling of this configuration entity is not required as a default Routing Policy is provided, which is automatically associated with all relevant configuration entities.</td>
</tr>
<tr>
<td>Call Admission Control</td>
<td>Call Admission Control (CAC) lets you configure the maximum number of permitted concurrent calls (SIP dialogs) per IP Group, SIP Interface, SRD, or user.</td>
</tr>
<tr>
<td>Accounts</td>
<td>Accounts are used to register or authenticate a “served” SIP entity (e.g., IP PBX) with a “serving” SIP entity (e.g., a registrar or proxy server). The device does this on behalf of the “served” IP Group. Authentication (SIP 401) is typically relevant for INVITE messages forwarded by the device to a “serving” IP Group. Registration is for REGISTER messages, which are initiated by the device on behalf of the “serving” SIP entity.</td>
</tr>
</tbody>
</table>
The associations between the configuration entities are summarized in the following figure:

**Figure 1-1: Association of Configuration Entities**

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

**Figure 1-2: SBC Configuration Terminology for Call Processing**

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

4. The device routes the dialog to an IP Group (destination), using the IP-to-IP Routing table. The destination SRD (and thus, SIP Interface and Media Realm) is the one assigned to the IP Group. Outbound manipulation can be applied to the outgoing dialog.
Part I

Getting Started with Initial Connectivity
2 Introduction

This part describes how to initially access the device's management interface and change its default IP address to correspond with your networking scheme.
This page is intentionally left blank.
3 Default OAMP IP Address

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

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

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

<table>
<thead>
<tr>
<th>IP Address</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Type</td>
<td>OAMP + Media + Control</td>
</tr>
<tr>
<td>IP Address</td>
<td>192.168.0.1</td>
</tr>
<tr>
<td>Prefix Length</td>
<td>24 (255.255.255.0)</td>
</tr>
<tr>
<td>Ethernet Device</td>
<td>vlan 1</td>
</tr>
<tr>
<td>Interface Name</td>
<td>O+A+M+P</td>
</tr>
</tbody>
</table>
This page is intentionally left blank.
4 Installing the Software

For installing the device, refer to the following documents:

- **Mediant Virtual Edition SBC**: *Mediant Virtual Edition SBC Installation Manual*
This page is intentionally left blank.
5 Changing Default IP Address to Suit your Network Addressing Scheme

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

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

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

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

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

- **To change the IP address through CLI:**
  1. Establish a CLI session with the device using any of the following connection methods:
     - **Server Edition:** Use a VGA monitor and keyboard to connect to the CLI management interface.
     - **Virtual Edition:** Connect to the Virtual Machine's (VM) console (e.g., in vSphere, click the Console tab).
  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, and then press Enter:
     - **Password:** Admin
  6. At the prompt, type the following commands to access the network interface configuration:
     - `# configure network`
     - `(config-network)# interface network-if 0`
     - `(network-if-0)#`

![Note: To ensure that you type the correct command syntax, use the Tab key to auto-complete partially entered commands.]
7. At the prompt, type the following commands to configure the IP address, prefix length and default gateway:

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

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

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

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

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

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

```
(config-network)# exit
```

11. At the prompt, make sure that Port #1 is connected (i.e., link is UP) by typing the following command:

```
# show network physical-port
```

The port is mapped to network-if-0, by default. For more information on mapping physical ports to the logical configuration ports, see "Configuring Ethernet Port Groups" on page 132.

Once you have assigned an IP address that suits your network environment, you can connect remotely with this IP address to the device's Web interface for management and configuration. To access the Web interface, see "Web-Based Management" on page 49.

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

- To modify and configure IP network interfaces, see "Configuring IP Network Interface" on page 137
- To configure the used physical Ethernet ports (speed, and mode), see "Configuring Physical Ethernet Ports" on page 130.
6 Licensing the Device

By default, the device is shipped with a pre-installed License Key that enables up to three call sessions only. After you have successfully installed the software (as described in the *Installation Manual*), follow the instructions in the *Installation Manual* for licensing your product and installing the License Key for the ordered features.
This page is intentionally left blank.
Part II

Management Tools
7 Introduction

This part describes the various management tools that you can use to configure the device:

- Embedded HTTP/S-based Web server - see "Web-based Management" on page 49
- Command Line Interface (CLI) - see "CLI-Based Management" on page 83
- Simple Network Management Protocol (SNMP) - see "SNMP-Based Management" on page 89
- Configuration ini file - see "INI File-Based Management" on page 97

Note:

- Some configuration settings can only be done using a specific management tool.
- For a list and description of all the configuration parameters, see "Configuration Parameters Reference" on page 831.
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 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.

Note:
- 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 License Key (see "License Key" on page 669).

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 11.0.13 or later)
  - Mozilla Firefox® (Versions 5.02 or later)
  - Google Chrome (Version 50 or later)
- 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.
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 Screen]

3. In the 'Username' and 'Password' fields, enter the username and password, respectively. The credentials are case-sensitive.

4. If you want the Web browser to remember your username and password, select the 'Remember Me' check box and then agree to the browser's prompt (depending on your browser). On your next login attempt, the 'Username' field is automatically populated with your username. Simply press the Tab or Enter key to auto-fill the 'Password' field, and then click Login.

5. Click Login.
8. Web-Based Management

Note:

• The default login username and password is "Admin" (case-sensitive). To change the login credentials, see "Configuring Management User Accounts" on page 73.

• By default, Web access is only through the IP address of the OAMP interface. However, you can allow access from all of the device’s IP network interfaces, by setting the EnableWebAccessFromAllInterfaces parameter to 1.

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

• Depending on your Web browser’s settings, a security warning box may be displayed. The reason for this is that the device’s certificate is not trusted by your PC. The browser may allow you to install the certificate, thus skipping the warning box the next time you connect to the device. If you are using Windows Internet Explorer, click View Certificate, and then Install Certificate. The browser also warns you if the host name used in the URL is not identical to the one listed in the certificate. To resolve this, add the IP address and host name (ACL_nnnnnn, where nnnnnn is the serial number of the device) to your hosts file, located at /etc/hosts on UNIX or C:\Windows\System32\Drivers\ETC\hosts on Windows; then use the host name in the URL (e.g., https://ACL_280152). Below is an example of a host file:

  127.0.0.1    localhost
  10.31.4.47   ACL_280152

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
<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Company logo.</td>
</tr>
<tr>
<td>2</td>
<td>Menu bar containing the menus.</td>
</tr>
</tbody>
</table>
| 3      | Toolbar providing frequently required command buttons.  
  - **Save**: Saves configuration changes to the device's flash memory (without resetting the device). If you make a configuration change, the button is surrounded by a red-colored border as a reminder to save your settings to flash memory, by clicking the button.  
  - **Reset**: Opens the Maintenance Actions page, which is used for performing various maintenance procedures such as resetting the device (see "Basic Maintenance" on page 649). If you make a configuration change that takes effect only after a device reset, the button is surrounded by a red-colored border as a reminder to save your settings to flash memory with a device reset; otherwise, your changes revert to previous settings if the device subsequently resets or powers off.  
  - **Actions**:  
    - **Configuration File**: Opens the Configuration File page, which is used for saving the ini file to a folder on your PC, or for loading an ini file to the device (see "Configuration File" on page 691).  
    - **Auxiliary Files**: Opens the Auxiliary Files page, which is used for loading Auxiliary files to the device (see "Loading Auxiliary Files through Web Interface" on page 656).  
    - **License Key**: Opens the License Key page, which is used for installing a new License Key file (see "Installing License Key through Web Interface" on page 673).  
    - **Software Upgrade**: Starts the Software Upgrade Wizard for upgrading the device's software (see "Software Upgrade Wizard" on page 683).  
    - **Switchover**: Opens the High Availability Maintenance page, which is used for switching between Active and Redundant devices (see High Availability Maintenance on page 652).  
    - **Configuration Wizard**: Opens the SBC Configuration Wizard, which is used for quick-and-easy configuration of the device (see SBC Configuration Wizard on page ). |
| 4      | Alarm bell icon, which displays the number of active alarms generated by the device. The color of the number of alarms display indicates the highest severity of an active alarm. If you click the icon, the Active Alarms table is displayed. For more information on the table, see Viewing Active Alarms. |
| 5      | Button displaying the username of the currently logged in user. If you click the button, information of the logged-in user is displayed (see "Viewing Logged-In User Information" on page 78) and the **Log Out** button is provided to log out the Web session (see "Logging Off the Web Interface" on page 66). |
| 6      | Tab bar containing tabs pertaining to the selected menu:  
  - Setup menu:  
    - IP Network  
    - Signaling & Media  
    - Administration  
  - Monitor menu: Monitor  
  - Troubleshoot menu: Troubleshoot |
| 7      | Back and Forward buttons that enable quick-and-easy navigation through previously opened pages. This is especially useful when you find that you need to return to a previously accessed page, and then need to go back to the page you just left. |
8. Web-Based Management

Item # | Description
--- | ---
| | • Back button: Goes back to the previously accessed page.
| | • Forward button: Opens the page that you initially left using the back button. The button is available only if you have used the Back button.

| 8 | Navigation pane, which displays the Navigation tree containing the commands (items) for opening the configuration pages (see "Navigation Tree" on page 53).
| 9 | SRD filter. When your configuration includes multiple SRDs, you can filter tables in the Web interface by a specific SRD. For more information, see "Filtering Tables in Web Interface by SRD" on page 333.
| 10 | Search box for searching parameter names and values (see "Searching for Configuration Parameters" on page 65).
| 11 | Work pane where configuration pages are displayed.

### 8.1.4 Accessing Configuration Pages from Navigation Tree

Accessing configuration pages is a three-fold process that consists of selecting a menu on the menu bar, a tab on the tab bar, and then a page item in the Navigation pane. The Navigation pane provides the Navigation tree, which is a tree-like structure of folders and page items that open configuration pages in the Work pane. The hierarchical structure and organization of the items in the Navigation tree allow you to easily drill-down and locate the required item.

The Navigation tree consists of the following areas:

- **Home**: (Callout #1) First ("home") page displayed when a menu-tab combination is initially selected. For example, the home page of the Setup menu - Administration tab combination is the Time & Date page.
- **Folders**: (Callout #2) Folders group items of similar functionality. To open and close a folder, simply click the folder name.
- **Items**: (Callout #3) Items open configuration pages. In some cases, an item may be listed under a sub-item. An item can open a page containing stand-alone parameters or a table. If it opens a page with stand-alone parameters, the item is displayed in italics. If it opens a page with a table, the item is displayed in regular font, or bold font to indicate an item that is commonly required.

![Figure 8-3: Navigation Tree (Example)](image)

The items of the Navigation tree depend on the menu-tab combination, selected from the menu bar and tab bar, respectively. The menus and their respective tabs are listed below:
Setup menu:
- IP Network tab
- Signaling & Media tab
- Administration tab

Monitor menu: Monitor tab

Troubleshoot menu: Troubleshoot tab

When you open the Navigation tree, folders containing commonly required items are opened by default, allowing quick access to their pages.

Items that open pages containing tables provide the following indications in the Navigation tree:

- Number of configured rows. For example, the item below indicates that two rows have been configured:

```
Ethernet Groups (2)
```

If you have filtered the Web interface display by SRD, the number reflects only the rows that are associated with the filtered SRD.

- Invalid row configuration. If you have configured a row with at least one invalid value, a red-colored icon is displayed next to the item, as shown in the following example:

```
Ethernet Groups (2) ●
```

If you hover your cursor over the icon, it displays the number of invalid rows (lines).

- Association with an invalid row: If you have associated a parameter of a row with a row of a different table that has an invalid configuration, the item appears with an arrow and a red-colored icon, as shown in the following example:

```
Ethernet Devices (2) ➔
```

If you hover your cursor over the icon, it displays the number of rows in the table that are associated with invalid rows.

- Folder containing an item with an invalid row: If a folder contains an item with an invalid row (or associated with an invalid row), the closed folder displays a red-colored icon, as shown in the following example:

```
CORE ENTRIES ●
```

If you hover your cursor over the icon, it displays the names of the items that are configured with invalid values. If you have filtered the Web interface display by SRD, only items with invalid rows that are associated with the filtered SRD are displayed.

To open a configuration page:

1. On the menu bar, click the required menu.
2. On the tab bar, click the required tab; the Navigation tree displays the items pertaining to the selected menu-tab combination.
3. In the Navigation pane, open the folder in which the required item is located. The folders are opened and closed by clicking the title of the folder. When opened, the folder's arrow is displayed as ▶; when closed, the arrow is displayed as ◄.
4. In the folder, click the required item; the page is displayed in the Work pane.

You can also easily navigate through previously accessed pages, using the Back and Forward buttons located above the Navigation pane:

- **Back** button: Click to go back to the previously accessed page or keep on clicking until you reach any other previously accessed page.
- **Forward** button: Click to open the page that you just left as a result of clicking the Back button.
These buttons are especially useful when you find that you need to return to a previously accessed page, and then need to go back to the page you just left.

**Note:** Depending on the access level (e.g., Monitor level) of your Web user account, certain pages may not be accessible or may be read-only (see "Configuring Management User Accounts" on page 73). For read-only privileges:

- Read-only pages with stand-alone parameters: "Read Only Mode" is displayed at the bottom of the page.
- Read-only pages with tables: Configuration buttons (e.g., **New** and **Edit**) are missing.

### 8.1.5 Configuring Stand-alone Parameters

Parameters that are not contained in a table are referred to as *stand-alone* parameters.

- If you change the value of a parameter (before clicking **Apply**), the parameter's field is highlighted, as shown in the example below:

```
6010
```

- If you change the value of a parameter from its default value and then click **Apply**, a dot appears next to the parameter's field, as shown in the example below:

```
• 6010
```

- If you change the value of a parameter that is displayed with a lightning-bolt ⚡ icon (as shown in the example below), you must save your settings to flash memory with a device reset for your changes to take effect. When you change such a parameter and then click **Apply**, the **Reset** button on the toolbar is encircled by a red border. If you click the button, the Maintenance Actions page opens, which provides commands for doing this (see "Basic Maintenance" on page 649).

```
6010
```

- Typically required parameters are displayed in bold font.
- If you enter an invalid value for a parameter and then click **Apply**, a message box appears notifying you of the invalid value. Click **OK** to close the message. The parameter reverts to its previous value and the field is surrounded by a colored border, as shown in the figure below:

```
6010
```

- To get help on a parameter, simply hover your mouse over the parameter's field and a pop-up help appears, displaying a brief description of the parameter.

The following procedure describes how to configure stand-alone parameters.

➢ **To configure a stand-alone parameter:**

1. Modify the parameter's value as desired.
2. Click **Apply**; the changes are saved to the device's volatile memory (RAM).
3. Save the changes to the device's non-volatile memory (flash):
   - If a device reset is not required:
a. On the toolbar, click **Save**; a confirmation message box appears:

   ![Save Configuration Confirmation Box](image)

   **Figure 8-4: Save Configuration Confirmation Box**

b. Click **Yes** to confirm; the changes are save to flash memory.

   - If a device reset is required:
     a. On the toolbar, click **Reset**; the Maintenance Actions page opens.
     b. Click **Reset**; the device saves the changes to flash memory and then resets.

   **Warning:** When you click **Apply**, your changes are saved only to the device’s volatile memory and thus, revert to their previous settings if the device later undergoes a hardware reset, a software reset (without saving to flash) or powers down. Therefore, make sure that you save your configuration to the device’s flash memory.

### 8.1.6 Configuring Table Parameters

A typical configuration table is shown below and subsequently described:

   ![Description of Tables](image)

   **Figure 8-5: Description of Tables**

<table>
<thead>
<tr>
<th>Item #</th>
<th>Button</th>
</tr>
</thead>
</table>

**Table 8-2: General Description of Configuration Tables**
8. Web-Based Management

### 8.1.6.1 Adding Table Rows

The following procedure describes how to add table rows. Before adding rows, the following GUI conventions are used:

- **Commonly required parameters are displayed in bold font.**
- **If you change the value of a parameter (before clicking Apply), the parameter’s field is highlighted,** as shown in the example below:

  ![Highlighted Field]

- **For indications of invalid values,** see "Invalid Value Indications" on page 60.

**To add a row:**

1. Click the **New** button, located on the table’s toolbar; a dialog box appears.
2. Configure the parameters of the row as desired. For information on configuring parameters that are assigned a value which is a row referenced from another table, see "Assigning Rows from Other Tables" on page 58.
3. Click **Apply** to add the row to the table or click **Cancel** to ignore your configuration.

4. If the **Save** button is surrounded by a red border, you must save your settings to flash memory, otherwise they are discarded if the device resets (without a save to flash) or powers off.

### 8.1.6.2 Assigning Rows from Other Tables

Some tables contain parameters whose value is an assigned row (referenced-row) from another table (referenced-table). For example, the IP Groups table contains the 'Proxy Set' parameter whose value is an assigned Proxy Set, configured in the Proxy Sets table. These parameter types provide a drop-down list for selecting the value and a **View** button, as shown in the example below:

![Figure 8-6: Parameter with View Button (Example)]

You can assign a referenced-row using one of the following methods:

- **Selecting a referenced-row from the drop-down list:**
  - Scroll down to the desired item and click it.
  - Search for the item by entering in the field the first few characters of the desired row, and then clicking it. The figure below shows an example of searched results for items (Proxy Sets) that begin with the letter "i":

![Figure 8-7: Searching a Row to Reference](image)

- **Selecting an existing referenced-row directly from the referenced-table:**
  a. Click **View**; the table (e.g., IP Groups table) and dialog box in which the button was clicked is minimized to the bottom-left corner of the Web interface and the referenced-table (e.g., Proxy Sets table) opens.
  b. Add a new row (e.g., Proxy Set), if required; otherwise, skip this step.
  c. Select the desired row in the row-referenced table (e.g., Proxy Sets table), and then click **Use selected row** located on the top-right of the table, as shown in the example below:

![Figure 8-8: Selecting Referenced Row](image)

- **Adding a new referenced-row:**
a. From the drop-down list, select the **Add new** option; as shown in the example below:

![Figure 8-9: Selecting Add new Option](image)

The table (e.g., IP Groups table) and dialog box in which the **Add new** option was selected is minimized to the bottom-left corner of the Web interface and a dialog box appears for adding a new row in the referenced-table (e.g., Proxy Sets table).

b. Configure the referenced-row and click **Apply**; the referenced-table (e.g., Proxy Sets table) closes and you are returned to the dialog box in which you selected the **Add new** option (e.g., IP Groups table), where the newly added row now appears selected.

You may want to access the referenced-table (e.g., Proxy Sets table) to simply view all its configured rows and their settings, without selecting one. To do this, click the **View** button. To return to the dialog box of the table (e.g., IP Groups table) in which you are making your configuration, click the arrow icon on the minimized dialog box to restore it to its previous size.

### 8.1.6.3 Modifying Table Rows

The following procedure describes how to modify (edit) the configuration of an existing table row. Remember that a gray-colored dot icon displayed next to a parameter's value (as shown in the example below), indicates that it was changed from its default value:

![6010]

➢ **To edit a table row:**

1. Select the row that you want to edit.
2. Click the **Edit** button, located on the table's toolbar; a dialog appears displaying the current configuration settings of the row.
3. Make your changes as desired, and then click **Apply**; the dialog box closes and your new settings are applied.
4. If the **Save** button is surrounded by a red border, you must save your settings to flash memory, otherwise they are discarded if the device resets (without a save to flash) or powers off.

### 8.1.6.4 Deleting Table Rows

The following procedure describes how to delete a row from a table.
To delete a table row:
1. Select the row that you want to delete.
2. Click the delete icon, located on the table's toolbar; a confirmation message box appears requesting you to confirm deletion, as shown in the example below:

   ![Delete Confirmation Message]

3. Click Yes, Delete; the row is removed from the table and the total number of configured rows that is displayed next to the page title and page item in the Navigation tree is updated to reflect the deletion.

   ![Note]

   Note: If the deleted row (e.g., a Proxy Set) was referenced in another table (e.g., IP Group), the reference is removed and replaced with an empty field. In addition, if the reference in the other table is for a mandatory parameter, the invalid icon is displayed where relevant. For example, if you delete a SIP Interface that you have assigned to a Proxy Set, the invalid icon appears alongside the Proxy Sets item in the Navigation tree as well as on the Proxy Sets page.

8.1.6.5 Invalid Value Indications

The Web interface provides the following indications of invalid values when configuring table rows:

- **Parameters configured with invalid values**: An invalid value is a value that is not permissible for the parameter. This can include incorrect syntax (string, numeral, or character) or an out-of-range value. If you enter an invalid value and then click Apply, the field is surrounded by a colored border, as shown in the example below.

   ![Invalid Value]

   If you hover your mouse over the field, a pop-up message appears providing the valid values. If you enter a valid value, the colored border is removed from the field. If you leave the parameter at the invalid value and click Apply, the parameter reverts to its previous value.

- **Mandatory parameters that reference rows of other configuration tables**:
  - **Adding a row**: If you do not configure the parameter and you click Apply, an error message is displayed at the bottom of the dialog box. If you click Cancel, the dialog box closes and the row is not added to the table. For example, if you do not configure the 'SIP Interface' field (mandatory) for a Proxy Set (in the Proxy Sets table), the below message appears::

    ![Error Message]

  - **Editing a row**: If you modify the parameter so that it's no longer referencing a row of another table (i.e., blank value), when you close the dialog box, the Invalid Line icon appears in the following locations:
Parameters that reference rows of other configuration tables that are configured with invalid values: If a row has a parameter that references a row of another table that has a parameter with an invalid value, the Invalid Reference Line icon is displayed in the following locations:

- 'Index' column of the row.
- Page title of the table. The total number of invalid rows in the table is also displayed with the icon.
- Item in the Navigation tree that opens the table.

For example, if you configure IP Group #0 (in the IP Groups table) with a parameter that references Proxy Set #0, which is configured with an invalid value, Invalid Reference Line icons are displayed for the IP Groups table, as shown below:

Figure 8-12: Invalid Reference Line Icons
8.1.6.6 Viewing Table Rows

Tables display a certain number of rows per page. If you have configured more than this number, you can use the table's navigation bar to scroll through the table pages, as shown below and described in the subsequent table:

Table 8-3: Table Navigation Bar Description

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | Navigation buttons to view previous table rows:  
  - Displays the previous table page  
  - Displays the first table page (i.e., page with at least the first index row) |
| 2      | Navigation buttons to view the next table rows:  
  - Displays the next table page  
  - Displays the last table page (i.e., page with last index row) |
| 3      | Currently displayed table page. To open a specific table page, enter the page number and then press the Enter key. |
| 4      | Total number of table pages. |
8.1.6.7 Sorting Tables by Column

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

➢ To sort table rows by column:

1. Click the name of the column by which you want to sort the table rows; the up-down arrows appear alongside the column name and the up button is displayed in a darker shade of color, indicating that the column is sorted in ascending order:

Figure 8-14: Table Sorted by Index in Ascending Order

2. To sort the column in descending order, click the column name again; only the down arrow is displayed in a darker shade of color, indicating that the column is sorted in descending order:

Figure 8-15: Table Sorted by Index in Descending Order

8.1.6.8 Changing Index Position of Table Rows

You can change the position (index) of rows in tables. This is done by using the up-down arrows located on the table’s toolbar.

Note:

- Changing row position can only done when the table is sorted by the 'Index' column and in ascending order; otherwise, the buttons are grayed out. For sorting table columns, see 'Sorting Tables by Column' on page 63.
- Changing row position is supported only by certain tables (e.g., IP-to-IP Routing table).

➢ To change the position of a row:

1. Click the 'Index' column header so that the rows are sorted in ascending order (e.g., 0.
1. Select the row that you want to move.

3. Do one of the following:
   - To move one index up (e.g., from Index 3 to 2): Click the up arrow; the row moves one index up in the table (e.g., to 2) and the row that originally occupied the index is moved one index down (e.g., to 3). In other words, the rows have swapped positions.
   - To move one index down (e.g., from Index 3 to Index 4): Click the down arrow; the row moves one index down in the table (e.g., to 4) and the row that originally occupied the index is moved one index up (e.g., to 3). In other words, the rows have swapped positions.

4. Continue clicking the required arrow until the row has moved to the desired location in the table.

8.1.6.9 Searching Table Entries

You can search for any parameter value (alphanumeric) in configuration tables, using the Search tool. The Search tool, located above each table, is shown below and described in the subsequent table:

![Figure 8-16: Table Search Tool](image)

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>'Specify Columns' drop-down list for selecting the table column (parameter) in which to do the search. By default, the search is done in all columns.</td>
</tr>
<tr>
<td>2</td>
<td>Search box to enter your search key (parameter value).</td>
</tr>
<tr>
<td>3</td>
<td>Magnifying-glass icon which when clicked performs the search.</td>
</tr>
</tbody>
</table>

To search for a table value:

1. If you want to perform the search on all table columns, skip this step; otherwise, from the 'Specify Columns' drop-down list, select the table column in which you want to perform the search; the name of the drop-down list changes to the name of the selected column.

2. In the Search box, enter the value for which you want to search.

3. Click the magnifying-glass icon to run the search. If the device finds the value, the table displays only the rows in which the value was found. You can then select any row and modify it by clicking the Edit button. If the search is unsuccessful, no rows are displayed.

4. To quit the Search tool and continue configuring rows, click the icon located in the Search box.
8.1.7 Searching for Configuration Parameters

You can search in the Web interface for parameter names (standalone or table parameters) and values. The search key can include the full parameter name (Web or ini file name) or a substring of it. If you search for a substring, all parameters containing the substring in their names are listed in the search result. For example, to search for the parameter 'Telnet Server TCP Port', you can use any of the following search keys:

- "Telnet Server TCP Port" (Web name)
- "TelnetServerPort" (ini file name)
- "Telnet"
- "Port"

When the device completes the search, it displays a list of found results based on the search key. Each possible result, when clicked, opens the page on which the parameter or value is located. You need to click the most appropriate result.

➢ To search for a parameter:

1. In the search box, enter the search key (parameter name or value).
2. Click the search icon; the Search Result window appears, listing found parameters based on your search key. Each searched result displays the following:
   - Navigation path (link) to the page on which the parameter appears
   - Parameter's name
   - Parameter's value
   - Brief description of parameter

![Figure 8-17: Search Result Window]

3. Click the link of the navigation path corresponding to the required found parameter to open the page on which the parameter appears.
8.1.8 Getting Help

The Web interface provides you with context-sensitive pop-up help of standalone parameters. When you hover your mouse over a parameter’s field, a pop-up appears with a short description of the parameter, as shown in the following example:

![Figure 8-18: Viewing Context-Sensitive Help for a Parameter](image)

8.1.9 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 menu bar, from the ‘Admin’ drop-down list, click **Log Out**; the following confirmation message box appears:

    ![Figure 8-19: Log Out Confirmation Box](image)

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

8.2 Customizing the Web Interface

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

- Corporate logo (see Replacing the Corporate Logo on page 66)
- Device’s (product) name (see Customizing the Product Name on page 69)
- Favicon (see Customizing the Favicon on page 70)
- Login welcome message (see Creating a Login Welcome Message on page 65)

**Note:**

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

8.2.1 Replacing the Corporate Logo

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

- **Web Login screen:**
  
  ![Figure 8-20: Corporate Logo on Web Login Screen](image)

  You can replace the logo with one of the following:
  - A different image (see Replacing the Corporate Logo with an Image on page 67)
  - Text (see Replacing the Corporate Logo with Text on page 68)

- **Menu bar:**

  ![Figure 8-21: Corporate Logo on Menu Bar](image)

  You can replace the logo with one of the following:
  - A different image (see Replacing the Corporate Logo with an Image on page 67)
  - Text (see Replacing the Corporate Logo with Text on page 68)

### 8.2.1.1 Replacing the Corporate Logo with an Image

You can replace the logo with a different image. The following figure displays an example where the default logo (top) is replaced with a different image (bottom):

![Figure 8-22: Customizing Web Logo Image](image)

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

**Figure 8-23: Customizing Web Logo**

![Image Load to Device](image.png)

5. Use the **Browse** button to select your logo file, and then click **Send File**; the device loads the file.

6. If you want to modify the width of the image, in the 'Logo Width' field, enter the new width (in pixels) and then click the **Set Logo Width** button.

7. On the left pane, click **Back to Main** to exit the Admin page.

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

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

### 8.2.1.2 Replacing the Corporate Logo with Text

You can replace the logo with text. The following figure displays an example where the logo (top) is replaced with the text, "My Logo Text" (bottom):

**Figure 8-24: Replacing Logo with Text**

![Replacing Logo with Text](image.png)

➢ **To replace the logo with text:**

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

   ```ini
   UseWebLogo = 1
   WebLogoText = < your text >
   ```

2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 655).

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

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

- **Web Login screen:**

  ![](image1.png) Figure 8-25: Customizing Product Name (Example)

  - **Ini file "Board" field:**
    
    Board: My Product Name

  - **CLI prompt:**
    
    My Product Name(config-system)#

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

    1. Create an ini file that includes the following parameter settings:
      
      ```
      UseProductName = 1
      UserProductName = < name >
      ```

    2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 655).

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

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

To customize the favicon:

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

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

Note:
- The logo image file type can be ICO, GIF, or PNG.
- The maximum size of the image file can be 16 Kbytes.
- Ignore the ini Parameters option, which is located on the left pane of the Admin page.
8.2.4 Creating a Login Welcome Message

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

![Figure 8-28: Creating Login Welcome Message](image)

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

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

  2. Load the ini file using the Auxiliary Files page (see Loading Auxiliary Files on page 655).
  3. Reset the device with a save-to-flash for your settings to take effect.

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

8.3 Configuring Additional Management Interfaces

The Additional Management Interfaces table lets you configure up to 64 management interfaces, in addition to the OAMP management interface in the IP Interfaces table.
Multiple management interfaces lets you access the device’s management interfaces (e.g., Web interface and CLI) remotely through different IP addresses. Each additional management interface can be configured to use a specific network interface (Control and/or Media type) and TLS Context, and can be configured to restrict access through HTTPS only.

Note:
- To allow access to the device’s management interfaces through all network interfaces in the IP Interfaces table, see the EnableWebAccessFromAllInterfaces parameter. This parameter does not specify a TLS Context nor a connectivity protocol (HTTP or HTTPS).
- Currently, additional management interfaces are not supported for REST API (ARM).

The following procedure describes how to configure additional management interfaces through the Web interface. You can also configure it through ini file (AdditionalManagementInterfaces) or CLI (configure system > additional-mgmt-if).

To configure additional management interfaces:
1. Open the Additional Management Interfaces table (Setup menu > Administration tab > Web & CLI folder > Additional Management Interfaces).
2. Click New; the following dialog box is displayed:

   ![Additional Management Interfaces Table - Add Dialog Box](image)

   - Configure an additional management interface according to the parameters described in the table below.
3. Click Apply, and then save your settings to flash memory.

   Additional Management Interfaces Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[AdditionalManagementInterfaces_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Interface Name</td>
<td>Assigns an IP network interface (from the IP Interfaces table) to the management interface.</td>
</tr>
<tr>
<td>interface-name</td>
<td></td>
</tr>
<tr>
<td>TLS Context Name</td>
<td></td>
</tr>
<tr>
<td>HTTP Only</td>
<td></td>
</tr>
<tr>
<td>[Use global definition]</td>
<td></td>
</tr>
</tbody>
</table>
8. Web-Based Management

### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [AdditionalManagementInterfaces_InterfaceName] | Network Interfaces on page 137. **Note:**
  - Only Control- and/or Media-type IP network interfaces can be associated with additional management interfaces.
  - An IP network interface can be associated with only **one** additional management interface. |

| TLS Context Name tls-context-name [AdditionalManagementInterfaces_TLSContextName] | Assigns a TLS Context (from the TLS Contexts table) to the management interface. A TLS Context provides secure TLS-based management access. For more information on TLS Contexts, see Configuring TLS Certificate Contexts on page 105. |

| HTTPS Only https-only-val [AdditionalManagementInterfaces_HTTPSOnly] | Defines the protocol required for accessing the management interface.  
  - [0] HTTP and HTTPS = The management interface can be accessed over a secured (HTTPS) and an unsecured (HTTP) connection.  
  - [1] HTTPS Only = The management interface can be accessed only over a secured (HTTPS) connection.  
  - [2] Use global definition = The type of management connection (HTTP and HTTPS, or HTTPS Only) depends on the configuration of the global parameter, HTTPSOnly (see Configuring Secured (HTTPS) Web on page 80). |

---

### 8.4 Configuring Management User Accounts

The Local Users table lets you configure up to 10 management user accounts for the device's Web interface and CLI. You configure each user account with login credentials (username and password) and with a management user level which defines the level of read and write privileges. The table below describes the different types of user levels:

#### Table 8-5: Description of Management User Levels

| User Level           | Numeric Representation in RADIUS | Privileges \n|----------------------|---------------------------------|---------------------------------------------------------------|
|----------------------|---------------------------------|---------------------------------------------------------------|
| **Security Administrator** | 200                              | Read/write privileges for all Web pages. This user level can create all other user levels and is the only one that can create the first Master user.  
  **Note:** At least one Security Administrator user must exist. |
| **Master**           | 220                              | Read/write privileges for all Web pages. This user level can create all user levels, including additional Master users and Security Administrators. It can delete all users except the last Security Administrator.  
  **Note:** Only Master users can delete Master users. If only one Master user exists, it can be deleted only by itself. |
| **Administrator**    | 100                              | Read/write privileges for all Web pages, except security-related pages and the Local Users table where this user... |
Mediant VE/SE SBC

User Level | Numeric Representation in RADIUS | Privileges
---|---|---
Monitor | 50 | Read-only privileges and access to security-related pages is blocked.

Note: Only Security Administrator and Master users can configure users in the Local Users table. Administrator users have read-only privileges and Monitor users are denied access to the table. However, Administrator and Monitor users can change their login credentials in the Web Settings page (see "Configuring Web Session and Access Settings" on page 79).

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

Table 8-6: Default User Accounts

<table>
<thead>
<tr>
<th>User Level</th>
<th>Username (Case-Sensitive)</th>
<th>Password (Case-Sensitive)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Administrator</td>
<td>&quot;Admin&quot;</td>
<td>&quot;Admin&quot;</td>
</tr>
<tr>
<td>Monitor</td>
<td>&quot;User&quot;</td>
<td>&quot;User&quot;</td>
</tr>
</tbody>
</table>

Note:
- For security, it's recommended that you change the default username and password of the default users.
- To restore the device to the default users (and with their default usernames and passwords), configure the ini file ResetWebPassword parameter to 1. If you have configured any other accounts, they are deleted.
- If you delete a user who is currently in an active Web session, the user is immediately logged off the device.
- Up to five users can be concurrently logged in to the Web interface; they can all be the same user.
- You can set the entire Web interface to read-only (regardless of Web user access levels), using the ini file parameter DisableWebConfig (see "Web and Telnet Parameters" on page 831).
- You can define additional Web user accounts using a RADIUS server (see "RADIUS Authentication" on page 231).

The following procedure describes how to configure user accounts through the Web interface. You can also configure it through ini file (WebUsers) or CLI (configure system > user).
To configure management user accounts:

1. Open the Local Users table (Setup menu > Administration tab > Web & CLI folder > Local Users).
2. Click New; the following dialog box is displayed:

   Figure 8-30: Local Users Table - Dialog Box

3. Configure a user account according to the parameters described in the table below.
4. Click Apply, and then save your settings to flash memory.

Table 8-7: Local Users Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index [WebUsers_Index]</td>
<td>Defines an index number for the new table row. Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Username user [WebUsers_Username]</td>
<td>Defines the Web user’s username. The valid value is a string of up to 40 alphanumeric characters, including the period &quot;.&quot;, underscore &quot;_&quot;, and hyphen &quot;-&quot; signs.</td>
</tr>
</tbody>
</table>
| Password password [WebUsers_Password] | Defines the Web user’s password. The valid value is a string of 8 to 40 ASCII characters. To ensure strong passwords, adhere to the following password complexity requirements:  
  - Contain at least eight characters.  
  - Contain at least two letters that are upper case (e.g., A).  
  - Contain at least two letters that are lower case (e.g., a).  
  - Contain at least two numbers (e.g., 4).  
  - Contain at least two symbols (non-alphanumeric characters) (e.g., $, #, %).  
  - No spaces.  
  - Contain at least four new characters that were not used in the previous password.  
Note: To enforce the password complexity requirements mentioned above, configure the EnforcePasswordComplexity to 1. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Level</strong></td>
<td>Defines the user’s access level.</td>
</tr>
<tr>
<td>privilege</td>
<td>- Monitor = (Default) Read-only user. This user can only view Web pages and access to security-related pages is denied.</td>
</tr>
<tr>
<td>[WebUsers_UserLevel]</td>
<td>- Administrator = Read/write privileges for all pages except security-related pages including the Local Users table where this user has read-only privileges.</td>
</tr>
<tr>
<td></td>
<td>- Security Administrator = Full read/write privileges for all pages. This user also functions as a security administrator.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>- At least one Security Administrator must exist. You cannot delete the last remaining Security Administrator.</td>
</tr>
<tr>
<td></td>
<td>- The first Master user can be added only by a Security Administrator user.</td>
</tr>
<tr>
<td></td>
<td>- Additional Master users can be added, edited and deleted only by Master users.</td>
</tr>
<tr>
<td></td>
<td>- If only one Master user exists, it can be deleted only by itself.</td>
</tr>
<tr>
<td></td>
<td>- Master users can add, edit, and delete Security Administrators (except the last Security Administrator).</td>
</tr>
<tr>
<td></td>
<td>- Only Security Administrator and Master users can add, edit, and delete Administrator and Monitor users.</td>
</tr>
<tr>
<td><strong>SSH Public Key</strong></td>
<td>Defines a Secure Socket Shell (SSH) public key for RSA public-key authentication (PKI) of the remote user when logging into the device’s CLI through SSH. Connection to the CLI is established only when a successful handshake with the user’s private key occurs.</td>
</tr>
<tr>
<td>public-key</td>
<td>The valid value is a string of up to 512 characters. By default, no value is defined.</td>
</tr>
<tr>
<td>[WebUsers_SSHPublicKey]</td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>- For more information on SSH and for enabling SSH, see Enabling SSH with RSA Public Key for CLI on page 84.</td>
</tr>
<tr>
<td></td>
<td>- To configure whether SSH public keys are optional or mandatory, use the SSHRequirePublicKey parameter.</td>
</tr>
<tr>
<td></td>
<td>- If not configured, the settings of the global parameter, SSHAdminKey is used.</td>
</tr>
<tr>
<td><strong>Status</strong></td>
<td>Defines the status of the user.</td>
</tr>
<tr>
<td>status</td>
<td>- 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.</td>
</tr>
<tr>
<td>[WebUsers_Status]</td>
<td>- Valid = User can log in to the Web interface as normal.</td>
</tr>
<tr>
<td></td>
<td>- Failed Login = The state is automatically set for users that exceed a user-defined number of failed login attempts, set by the 'Deny Access on Fail Count' parameter (see 'Configuring Web Session and Access Settings' on page 79). 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 Security Administrator or Master.</td>
</tr>
</tbody>
</table>
### 8. Web-Based Management

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Inactivity</td>
<td>The state is automatically set for users that have not accessed the Web interface for a user-defined number of days, set by the 'User Inactivity Timer' (see 'Configuring Web Session and Access Settings' on page 79). 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.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The Inactivity status is applicable only to Administrator and Monitor users; Security Administrator and Master users can be inactive indefinitely.</td>
</tr>
<tr>
<td></td>
<td><strong>For security, it is recommended to set the status of a newly added user to New in order to enforce password change.</strong></td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td></td>
</tr>
<tr>
<td>Password Age</td>
<td>Defines the duration (in days) of the validity of the password. When the 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.</td>
</tr>
<tr>
<td>Security Password Age</td>
<td></td>
</tr>
<tr>
<td>session-limit</td>
<td>Defines the maximum number of concurrent Web interface sessions allowed for the specific user. For example, if configured to 2, the same user account can be logged into the device’s Web interface (i.e., same username-password combination) from two different management stations (i.e., IP addresses) at any one time. Once the user logs in, the session is active until the user logs off (by clicking the Log off icon on the toolbar) or until the session expires if the user is inactive for a user-defined duration (see the 'Session Timeout' parameter below). The valid value is 0 to 5. The default is 2. <strong>Note:</strong> Up to five users can be concurrently logged in to the Web interface.</td>
</tr>
<tr>
<td>Session Timeout</td>
<td>Defines the duration (in minutes) of inactivity of a logged-in user in the Web interface, after which the user is automatically logged off the Web session. In other words, the session expires when the user has not performed any operations (activities) in the Web interface for the configured timeout duration. The valid value is 0 to 100000. A value of 0 means no timeout. The default value is according to the settings of the WebSessionTimeout global parameter (see 'Configuring Web Session and Access Settings' on page 79).</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>To enable this feature, see the 'Deny Access On Fail Count' parameter in 'Configuring Web Session and Access Settings' on page 79.</strong></td>
</tr>
<tr>
<td></td>
<td>The 'Deny Authentication Timer' parameter relates to failed Web logins from specific IP addresses.</td>
</tr>
</tbody>
</table>
8.5 Displaying Login Information upon Login

You can enable the device to display login information immediately upon Web login.

➢ To enable display of user login information upon login:
1. Open the Web Settings page (Setup menu > Administration tab > Web & CLI folder > Web Settings).
2. Under the Security group, from the 'Display Last Login Information' drop-down list, select Enable.
3. Click Apply.
Once enabled, each time you login to the device, the Login Information window is displayed, as shown in the example below:

Figure 8-31: Login Information Window

To close the window, click Close.

8.6 Viewing Logged-In User Information

The username of the currently logged in user is displayed in the top-right corner of the Web interface. If you click the username (e.g., "Admin"), a pop-up callout appears:

Figure 8-32: Logged-in User Information

The following information is displayed:
- 'Access Level': User level of the currently logged in user (e.g., Security Administrator).
- 'Session Time': Duration of the current Web session (starting from login).

The Log Out button is also provided for logging out of the Web session (see "Logging Off the Web Interface" on page 66).
8.7 Configuring Web Session and Access Settings

The following procedure describes how to configure security features related to Web user sessions and access.

Note: You can only perform the configuration described in this section if you are a management user with Security Administrator level or Master level. For more information, see “Configuring Management User Accounts” on page 73.

To configure Web user sessions and access security:

1. Open the Web Settings page (Setup menu > Administration tab > Web & CLI folder > Web Settings).
2. Under the Session group, configure the following parameters:

   ![Figure 8-33: Configuring Web User Sessions](image)

   - 'Password Change Interval': Duration (in minutes) of the validity of the Web login passwords. When the duration expires, the user must change the password in order to log in again.
   - 'User Inactivity Timeout': If the user has not logged into the Web interface within this duration, the status of the user becomes inactive and the user can no longer access the Web interface. The user can only log in to the Web interface if its status is changed (to New or Valid) by a Security Administrator or Master user (see "Configuring Management User Accounts" on page 73).
   - 'Session Timeout': Duration (in minutes) of inactivity (i.e., no actions are performed in the Web interface) of a logged-in user, after which the Web session expires and the user is automatically logged off the Web interface and needs to log in again to continue the session. You can also configure the functionality per user in the Local Users table (see "Configuring Management User Accounts" on page 73), which overrides this global setting.

3. Under the Security group, configure the following parameters:

   ![Figure 8-34: Configuring Web User Security](image)

   - 'Deny Authentication Timer': Interval (in seconds) that the user needs to wait before logging in from the same IP address after reaching the maximum number of failed login attempts (see next step).
8.8 Changing Login Password for Administrator and Monitor Users

If you are logged in as a user with Administrator level or Monitor level, you can change your login password by performing the following procedure.

**Note:**
- Users with Security Administrator level or Master level can change passwords for themselves and for other users in the Local Users table (see "Configuring Management User Accounts" on page 73).
- You can only change the password if the duration configured in the 'Password Change Interval' has elapsed (see "Configuring Web Session and Access Settings" on page 79).

➢ To change the login password:

1. Open the Web Settings page (Setup menu > Administration tab > Web & CLI folder > Web Settings).

Figure 8-35: Changing Login Password for Administrator and Monitor User Levels

![](image)

2. In the 'Current Password' field, type in your current login password.
3. In the 'New Password' field, type in your new password.
4. In the 'Confirm New Password' field, type in your new password again.
5. Click Change; you are logged off the Web session and prompted to login in again with your new login password.

8.9 Configuring Secured (HTTPS) Web

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

➢ To configure secure (HTTPS) Web access:

1. Open the Web Settings page (Setup menu > Administration tab > Web & CLI folder > Web Settings).
2. Under the General group, configure the following:
3. From the 'Secured Web Connection (HTTPS)' drop-down list, select **HTTPS Only**.

4. To enable two-way authentication whereby both management client and server are authenticated using X.509 certificates, from the ' Require Client Certificates for HTTPS connection' drop-down list, select **Enable**.

5. In the 'HTTPS Cipher String' field, enter the cipher string for HTTPS (in OpenSSL cipher list format).

6. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

For more information on secure Web-based management including TLS certificates, see "TLS for Remote Device Management" on page 118.

### 8.10 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.

### 8.11 Configuring Web and Telnet Access List

The Access List table lets you restrict access to the device's management interfaces (Web, Telnet and SSH) by specifying IP addresses (up to ten) of management clients that are permitted to access the device. Access to the device's management interfaces from undefined IP addresses is denied. If you don't specify any IP addresses, this security feature is inactive and the device can be accessed from any IP address.
The following procedure describes how to configure the Access List through the Web interface. You can also configure it through ini file (WebAccessList_x).

**Note:**
- Configure the IP address of the computer from which you are currently logged into the device as the first authorized IP address in the Access List. If you configure any other IP address, access from your computer will be immediately denied.
- If you configure network firewall rules in the Firewall table (see "Configuring Firewall Rules" on page 165), you must configure a firewall rule that permits traffic from IP addresses configured in the Access List table.

**➢ To add IP addresses to the Access List:**

1. Open the Access List table (Setup menu > Administration tab > Web & CLI folder > Access List).

   ![Figure 8-36: Access List - Adding IP Address](image)

2. In the ‘Add an authorized IP address’ field, configure an IP address, and then click Add New Entry; the IP address is added to the table.

   ![Figure 8-37: Web & Telnet Access List Table](image)

If you have configured IP addresses in the Access List and you no longer want to restrict access to the management interface based on the Access List, delete all the IP addresses in the table, as described in the following procedure.

**Note:** When deleting all the IP addresses, make sure that you delete the IP address of the computer from which you are currently logged into the device, last; otherwise, access from your computer will be immediately denied.

**➢ To delete an IP address from the Access List:**

1. Select the Delete Row check box corresponding to the IP address that you want to delete.
2. Click Delete Selected Addresses.
9 CLI-Based Management

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

Note:
- By default, CLI is disabled (for security purposes).
- The CLI can only be accessed by management users with the following user levels:
  - Administrator
  - Security Administrator
  - Master
- For more information on using the CLI and CLI commands, refer to the CLI Reference Guide.

9.1 Enabling CLI

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

9.1.1 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 clear text. 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. To configure such a message, see "Creating a Login Welcome Message" on page 65.

➢ To enable Telnet:

1. Open the CLI Settings page (Setup menu > Administration tab > Web & CLI folder > CLI Settings).

2. Configure the following parameters:
   - 'Embedded Telnet Server': Select Enable Unsecured or Enable Secured (i.e., SSL) to enable Telnet.
   - 'Telnet Server TCP Port': Enter the port number of the embedded Telnet server.
• "Telnet Server Idle Timeout": Enter the duration of inactivity in the Telnet session after which the session automatically ends.

3. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

For a detailed description of the Telnet parameters, see "Telnet Parameters" on page 837.

### 9.1.2 Enabling SSH with RSA Public Key for CLI

Unless configured for TLS, Telnet is not secure as it requires passwords to be transmitted in clear text. To overcome this, you can use Secure SHell (SSH) 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/](http://www.chiark.greenend.org.uk/~sgtatham/putty/). By default, SSH uses the same username and password as the device’s 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 (public key) between "ssh-rsa" and "rsa-key-…..", as shown in the example below:

   ![Figure 9-1: Selecting Public RSA Key in PuTTY](image)

2. You can use the public key per management user or for all management users:
• Per user: Open the Local Users table (see Configuring Management User Accounts on page 73), and then for the required user, paste the public key that you copied in Step 1.d into the 'SSH Public Key' field, as shown below:

**Figure 9-2: Pasting Public RSA Key per User in Local Users Table**

```markdown
<table>
<thead>
<tr>
<th>GENERAL</th>
<th>SECURITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>0</td>
</tr>
<tr>
<td>Username</td>
<td>Admin</td>
</tr>
<tr>
<td>Password</td>
<td>*</td>
</tr>
<tr>
<td>User Level</td>
<td>Security Administrator</td>
</tr>
<tr>
<td>SSH public key</td>
<td>AAAANR12Y26AAAAABPMAJ1H5S</td>
</tr>
<tr>
<td>Status</td>
<td>Valid</td>
</tr>
</tbody>
</table>
```

• For all users: Open the CLI Settings page (Setup menu > Administration tab > Web & CLI folder > CLI Settings), and then paste the public key that you copied in Step 1.d into the 'Admin Key' field, as shown below:

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

```
<table>
<thead>
<tr>
<th>Secure Shell (SSH)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable SSH Server</td>
</tr>
<tr>
<td>Server Port</td>
</tr>
<tr>
<td>Admin Key</td>
</tr>
<tr>
<td>Public Key</td>
</tr>
<tr>
<td>Max Payload Size</td>
</tr>
<tr>
<td>Max Binary-Packet Size</td>
</tr>
<tr>
<td>Enable Last Login Message</td>
</tr>
<tr>
<td>Max Login Attempts</td>
</tr>
<tr>
<td>Maximum SSH Sessions</td>
</tr>
</tbody>
</table>
```

3. On the CLI Settings page, do the following:
   a. From the 'Enable SSH Server' drop-down list, select **Enable**.
   b. For additional security, you can configure the 'Require Public Key' field to **Enable**. This ensures that SSH access is only possible by using the RSA key and not by username and password.
   c. Configure the other SSH parameters as required. For a description of these parameters, see SSH Parameters on page 873.
   d. Click **Apply**.

4. 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.

5. Connect to the device with SSH using the username "Admin"; RSA key negotiation
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:

   ```bash
   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. You can use the public key per management user or for all users:
   - Per user: Open the Local Users table (see Configuring Management User Accounts on page 73), and then for the required user, paste the public key that you copied in Step 2 into the 'SSH Public Key' field.
   - For all users: Open the CLI Settings page, and then paste the public key that you copied in Step 2 into the 'Admin Key' field.

4. Connect to the device with SSH, using the following command (where xx.xx.xx.xx is the device's IP address):

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

   RSA-key negotiation occurs automatically and no password is required.

9.2 Configuring Maximum Telnet/SSH Sessions

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

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

To configure the maximum number of concurrent Telnet and SSH sessions:

1. Open the CLI Settings page (Setup menu > Administration tab > Web & CLI folder > CLI Settings).

2. For Telnet: Under the Telnet group, in the 'Maximum Telnet Sessions' field, enter the maximum number of concurrent sessions.

3. For SSH: Under the SSH group, in the 'Maximum SSH Sessions' field, enter the maximum number of concurrent sessions.

4. Click Apply.

9.3 Establishing a CLI Session

You can access the device's CLI using any of the following methods:

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

- **Secure SHell (SSH):** The device can be accessed through its Ethernet interface by the SSH protocol using SSH client software. A popular and freeware SSH client software is Putty, which can be downloaded from [http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html](http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html)
9. CLI-Based Management

**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. To configure login credentials and management user accounts, see "Configuring Management User Accounts" on page 73.

**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 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), 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:**

1. Establish a CLI session with the device.
2. Run the following command:

```
# 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.
9.5 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:
   1. Establish a CLI session with the device.
   2. Run the following command:
      
      ```
      # clear user <session ID>
      ```

      Where `<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 87).

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

9.6 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 from 0 to 65,535, or determined by resizing the terminal window by mouse-dragging the window's border.

➢ To specify the number of displayed output lines:
   1. Establish a CLI session with the device.
   2. Access the System menu:
      
      ```
      # configure system
      ```

   3. At the prompt, type the following command:
      
      ```
      (config-system)# cli-terminal
      ```

   4. At the prompt, type the following command:
      
      ```
      <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 displayed output lines by dragging terminal window:
   1. Establish a CLI session with the device.
   2. Access the System menu:
      
      ```
      # configure system
      ```

   3. At the prompt, type the following command:
      
      ```
      (config-system)# cli-terminal
      ```

   4. At the prompt, type the following command:
      
      ```
      <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.
10 SNMP-Based Management

The device provides an embedded SNMP agent that lets you manage it using AudioCodes Element Management System (EMS) or a third-party SNMP manager. The SNMP agent supports standard and proprietary Management Information Base (MIBs). All supported MIB files are supplied to customers as part of the release. The SNMP agent can send unsolicited SNMP trap events to the SNMP manager.

Note:
- By default, SNMP-based management is enabled.
- For more information on the device's SNMP support such as SNMP trap alarms and events, refer to the SNMP Reference Guide.
- For more information on AudioCodes EMS, refer to the EMS User's Manual.

10.1 Disabling SNMP

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

➢ To disable SNMP:

1. Open the SNMP Community Settings page (Setup menu > Administration tab > SNMP folder > SNMP Community Settings).
2. From the 'Disable SNMP' drop-down list (DisableSNMP parameter), select Yes:

3. Click Apply.

10.2 Configuring SNMP Community Strings

SNMP community strings determine the access privileges (read-only and read-write) of SNMP clients with the device's SNMP agent. You can configure up to five read-only SNMP community strings and up to five read-write SNMP community strings. The device's SNMP agent accepts SNMP Get (read-only) and Set (read-write) requests only if the correct community string is used in the request.

You can also configure a unique password-like community string used for sending SNMP traps. The device sends the traps with the community string.

Note:
- SNMP community strings are applicable only to SNMPv1 and SNMPv2c; SNMPv3 uses username-password authentication along with an encryption key (see "Configuring SNMP V3 Users" on page 94).
- You can enhance security by configuring Trusted Managers (see "Configuring SNMP Trusted Managers" on page 93). A Trusted Manager is an IP address from which the SNMP agent accepts Get and Set requests.
For detailed descriptions of the SNMP parameters, see "SNMP Parameters" on page 838.

➢ To configure SNMP community strings:

1. Open the SNMP Community Settings page (Setup menu > Administration tab > SNMP folder > SNMP Community Settings).

2. Configure SNMP community strings for access privileges:

   - Under the Read Only Community Strings group, configure read-only community strings (see the table below).
   - Under the Read/Write Community Strings group, configure read-write community strings (see the table below).

3. Configure a community string for SNMP traps: Under the Misc. Settings group, in the 'Trap Community String' field, configure a community string (see the table below).

   **Figure 10-1: Configuring SNMP Trap Community String**

   Trap Community String: trapuser

4. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

   To delete a community string, delete the configured string, click Apply, and then reset the device with a save-to-flash for your settings to take effect.

   **Table 10-1: SNMP Community String Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Read-Only Community Strings</strong>&lt;br&gt;configure system &gt; snmp settings &gt; ro-community-string [SNMPReadOnlyCommunityString_x]</td>
<td>Defines read-only SNMP community strings. Up to five read-only community strings can be configured. The valid value is a string of up to 30 characters that can include only the following: &lt;ul&gt;  • Upper- and lower-case letters (a to z, and A to Z)  • Numbers (0 to 9)  • Hyphen (-)  • Underline (_) &lt;/ul&gt; For example, &quot;Public-comm_string1&quot;. The default is &quot;public&quot;.</td>
</tr>
<tr>
<td><strong>Read-Write Community Strings</strong>&lt;br&gt;configure system &gt; snmp settings &gt; rw-community-string [SNMPReadWriteCommunityString_x]</td>
<td>Defines read-write SNMP community strings. Up to five read-write community strings can be configured. The valid value is a string of up to 30 characters that can include only the following: &lt;ul&gt;  • Upper- and lower-case letters (a to z, and A to Z)  • Numbers (0 to 9)  • Hyphen (-)  • Underline (_) &lt;/ul&gt; For example, &quot;Private-comm_string1&quot;. The default is &quot;private&quot;.</td>
</tr>
<tr>
<td><strong>Trap Community String</strong>&lt;br&gt;configure system &gt; snmp trap &gt; community-string [SNMPTrapCommunityString]</td>
<td>Defines the community string for SNMP traps. The valid value is a string of up to 30 characters that can include only the following: &lt;ul&gt;  • Upper- and lower-case letters (a to z, and A to Z)  • Numbers (0 to 9)  • Hyphen (-)  • Underline (_) &lt;/ul&gt; For example, &quot;Trap-comm_string1&quot;. The default is &quot;trapuser&quot;.</td>
</tr>
</tbody>
</table>

### 10.3 Configuring SNMP Trap Destinations with IP Addresses

The SNMP Trap Destinations table lets you to configure up to five SNMP trap managers to receive traps sent by the device. The SNMP manager is defined by IP address and port. 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.

The following procedure describes how to configure SNMP trap destinations through the Web interface. You can also configure it through ini file (SNMPManager) or CLI (configure system > snmp trap-destination).
To configure SNMP trap destinations:

1. Open the SNMP Trap Destinations table (Setup menu > Administration tab > SNMP folder > SNMP Trap Destinations).

2. Configure the SNMP trap manager according to the table below.
3. Select the check boxes corresponding to the configured SNMP managers that you want to enable.
4. Click Apply.

Note:
- Rows whose corresponding check boxes are cleared revert to default settings when you click Apply.
- To enable the sending of the trap event, acPerformanceMonitoringThresholdCrossing, which is sent every time a threshold (high or low) of a performance monitored SNMP object is crossed, configure the ini file parameter PM_EnableThresholdAlarms to 1.
- Instead of configuring SNMP trap managers with an IP address in dotted-decimal notation, you can configure a single SNMP trap manager with an FQDN (see "Configuring an SNMP Trap Destination with FQDN" on page 93.

Table 10-2: SNMP Trap Destinations Table Parameters Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(check box)</td>
<td>Enables the SNMP manager to receive traps and checks the validity of the configured destination (IP address and port number).</td>
</tr>
<tr>
<td>[SNMPManagerIsUsed_x]</td>
<td></td>
</tr>
<tr>
<td>IP Address</td>
<td>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.</td>
</tr>
<tr>
<td>[SNMPManagerTableIP_x]</td>
<td></td>
</tr>
<tr>
<td>Trap Port</td>
<td>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.</td>
</tr>
<tr>
<td>[SNMPManagerTrapPort_x]</td>
<td></td>
</tr>
<tr>
<td>Trap User</td>
<td>Associates a trap user with the trap destination. This determines the trap format, authentication level, and encryption level.</td>
</tr>
<tr>
<td>[SNMPManagerTrapUser]</td>
<td></td>
</tr>
</tbody>
</table>
### 10.4 Configuring an SNMP Trap Destination with FQDN

Instead of configuring SNMP trap destinations (managers) with IP addresses in dotted-decimal notation in the SNMP Trap Destinations table (see "Configuring SNMP Trap Destination with IP Addresses" on page 91), you can configure a single SNMP trap manager with an FQDN (e.g., mngr.corp.mycompany.com). The device sends the traps to the DNS-resolved IP address. The resolved IP address replaces the IP address of the last row (SNMP Manager 5) in the SNMP Trap Destinations table (and the last trap manager entry in the snmpTargetAddrTable in the snmpTargetMIB).

**Note:** If you configure an FQDN for an SNMP trap manager:
- The device ignores your configuration in the SNMP Trap Destinations table.
- Only one SNMP trap manager can be configured.

#### To configure an SNMP trap destination with an FQDN:
1. Open the SNMP Community Settings page (Setup menu > Administration tab > SNMP folder > SNMP Community Settings).
2. Under the Misc. Settings group, in the 'Trap Manager Host Name' field (SNMPTrapManagerHostName parameter), enter the FQDN.
3. Click Apply.

### 10.5 Configuring SNMP Trusted Managers

The SNMP Trusted Managers table lets you configure up to five SNMP Trusted Managers. 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 (see "Configuring SNMP Community Strings" on page 89). You can enhance security by configuring Trusted Managers, which is an IP address from which the device's SNMP agent accepts and processes SNMP requests. If no SNMP Trusted Manager is configured, any SNMP manager can access the device (as long as the community string is correct).

The following procedure describes how to configure SNMP Trusted Managers through the Web interface. You can also configure it through ini file (SNMPTtrustedMgr_x) or CLI (configure system > snmp settings > trusted-managers).
To configure SNMP Trusted Managers:

1. Open the SNMP Trusted Managers table (Setup menu > Administration tab > SNMP folder > SNMP Trusted Managers).

![Figure 10-3: SNMP Trusted Managers Table](image)

<table>
<thead>
<tr>
<th>DELETE</th>
<th>TRUSTED MANAGERS IP ADDRESS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SNMP Trusted Manager 1: 0.0.0.0</td>
</tr>
<tr>
<td></td>
<td>SNMP Trusted Manager 2: 0.0.0.0</td>
</tr>
<tr>
<td></td>
<td>SNMP Trusted Manager 3: 0.0.0.0</td>
</tr>
<tr>
<td></td>
<td>SNMP Trusted Manager 4: 0.0.0.0</td>
</tr>
<tr>
<td></td>
<td>SNMP Trusted Manager 5: 0.0.0.0</td>
</tr>
</tbody>
</table>

2. Configure an IP address (in dotted-decimal notation) for one or more SNMP Trusted Managers.
3. Select the check boxes corresponding to the configured SNMP Trusted Managers that you want to enable.
4. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

10.6 Enabling SNMP Traps for Web Activity

You can enable the device to send SNMP traps to notify of management users’ activities in the Web interface. A trap is sent each time an activity is done by a user. To configure the types of Web activities that you want reported, see "Configuring Reporting of Management User Activities" on page 799.

To enable traps to SNMP manager for Web activity:

1. Open the SNMP Community Settings page (Setup menu > Administration tab > SNMP folder > SNMP Community Settings).
2. Under the Misc. Settings group, from the ‘Activity Trap’ drop-down list (EnableActivityTrap), select Enable.

![Figure 10-4: Enabling Trap for Web User Activities](image)

3. Click Apply.

10.7 Configuring SNMP V3 Users

The SNMPv3 Users table lets you configure up to 10 SNMP v3 users for authentication and privacy.

The following procedure describes how to configure SNMP v3 users through the Web interface. You can also configure it through ini file (SNMPUsers) or CLI (configure system > snmp v3-users).

Note: If you delete a user that is associated with a trap destination (see "Configuring SNMP Trap Destinations with IP Addresses" on page 91), the trap destination becomes disabled and the trap user reverts to default (i.e., SNMPv2).
To configure an SNMP v3 user:

1. Open the SNMPv3 Users table (Setup menu > Administration tab > SNMP folder > SNMP V3 Users).
2. Click New; the following dialog box appears:

   **Figure 10-5: SNMPv3 Users Table - Dialog Box**

   ![SNMPv3 Users Table - Dialog Box](image)

3. Configure the SNMP V3 parameters according to the table below.
4. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

**Table 10-3: SNMPv3 Users Table Parameters Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[SNMPUsers_Index]</td>
<td></td>
</tr>
<tr>
<td>User Name</td>
<td>Name of the SNMP v3 user.</td>
</tr>
<tr>
<td>[SNMPUsers_Username]</td>
<td></td>
</tr>
<tr>
<td>Authentication Protocol</td>
<td>Authentication protocol of the SNMP v3 user.</td>
</tr>
</tbody>
</table>
| auth-protocol [SNMPUsers_AuthProtocol] | [0] None (default)  
|                            | [1] MD5                                               |
|                            | [2] SHA-1                                             |
| Privacy Protocol           | Privacy protocol of the SNMP v3 user.                 |
| priv-protocol [SNMPUsers_PrivProtocol] | [0] None (default)  
| Privacy Key                |                                                       |
| [SNMPUsers_PrivKey]        |                                                       |
| Group                      |                                                       |
| [SNMPUsers_Group]          |                                                       |

**Note:** Each row must be configured with a unique index.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>auth-key</td>
<td>or long hex string. Keys are always persisted as long hex strings and keys are localized.</td>
</tr>
<tr>
<td>[SNMPUsers_AuthKey]</td>
<td></td>
</tr>
<tr>
<td>Privacy Key</td>
<td>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.</td>
</tr>
<tr>
<td>priv-key</td>
<td></td>
</tr>
<tr>
<td>[SNMPUsers_PrivKey]</td>
<td></td>
</tr>
<tr>
<td>Group</td>
<td>The group with which the SNMP v3 user is associated.</td>
</tr>
<tr>
<td>group</td>
<td></td>
</tr>
<tr>
<td>[SNMPUsers_Group]</td>
<td></td>
</tr>
<tr>
<td>Note: All groups can be used to send traps.</td>
<td></td>
</tr>
</tbody>
</table>
11 INI File-Based Management

You can configure the device through an ini file, which is a text-based file with an *.ini file extension name, created using any standard text-based editor such as Notepad. Once you have created an ini file with all your configuration settings, you need to install (load) it to the device to apply the configuration. For a list of the ini file parameters, see "Configuration Parameters Reference" on page 831.

11.1 INI File Format

There are two types of ini file parameters:
- Individual parameters - see "Configuring Individual ini File Parameters" on page 97
- Table parameters - see "Configuring Table ini File Parameters" on page 97

11.1.1 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 ";".

```
[optional 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 99.

11.1.2 Configuring Table ini File Parameters

Table ini file parameters allow you to configure tables, which include multiple parameters (columns) and row entries (indices). The table ini file parameter is composed of the following elements:
- Table title: 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).

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

```
[ CodersGroup0 ]
FORMAT CodersGroup0_Index = CodersGroup0_Name, CodersGroup0_pTime,
CodersGroup0_rate, CodersGroup0_PayloadType, CodersGroup0_Sce,
CodersGroup0_CoderSpecific;
CodersGroup0 0 = g711Alaw64k, 20, 0, 255, 0, 0;
CodersGroup0 1 = eg711Ulaw, 10, 0, 71, 0, 0;
[ \CodersGroup0 ]
```

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

11.1.3 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.

11.2 Configuring an ini File

There are different methods that you can use for configuring an 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 652).
  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 ini files to the device, see "Loading an ini File to the Device" on page 100.

Note:
- If you save an ini file from the device and a table row is configured with invalid values, the ini file displays the row prefixed with an exclamation mark (!), for example:
  !CpMediaRealm 1 = "ITSP", "Voice", ",", 60210, 2, 6030, 0, ",", ",";
- To restore the device to default settings through the ini file, see "Restoring Factory Defaults" on page 725.
11.3 Loading an ini File to the Device

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

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

- **Web interface:**
  - Auxiliary Files page (see "Loading Auxiliary Files" on page 655): The device updates its configuration according to the loaded ini file while preserving the remaining current configuration.
  - Configuration File page (see "Configuration File" on page 691): 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.

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`.

11.4 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).

11.5 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>).
11.6 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
12 Configuring SSL/TLS Certificates

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

Note:
- The device is shipped with an active, default TLS setup. 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. To configure NTP, see "Configuring Automatic Date and Time using SNTP" on page 121.
- Only **Base64 (PEM)** encoded X.509 certificates can be loaded to the device.

12.1 Configuring TLS Certificate Contexts

The TLS Contexts table lets you configure up to 100 TLS certificates, referred to as **TLS Contexts**. The Transport Layer Security (TLS), also known as Secure Socket Layer (SSL) can be used to secure the device's SIP signaling connections or SIP over TLS (SIPS), Web (HTTPS) sessions, Telnet sessions and SSH sessions. 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 (configured in row index 0 and called "default"), which includes a self-generated random private key and a self-signed server certificate. The subject name of the default certificate is "ACL_nnnnnnn", where nnnnnnn denotes the serial number of the device.

Note:
- The default TLS Context cannot be deleted.
- The default TLS Context can be used for SIPS or any other supported application such as Web (HTTPS), Telnet, and SSH.
- If you configure new TLS Contexts, you can use them only for SIPS.
- If a TLS Context for an existing TLS connection is changed during the call by the user agent, the device ends the connection.

You can configure each TLS Context with the following:

- TLS version (SSL 3.0, TLS 1.0, TLS 1.1 and TLS 1.2)
- Encryption ciphers for server and client - DES, RC4 compatible, Advanced Encryption Standard (AES)
- 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. To configure TLS certificate expiry check, see "Configuring TLS Server Certificate Expiry Check" on page 120.
- Online Certificate Status Protocol (OCSP). Some Public-Key Infrastructures (PKI) can revoke a certificate after it has been issued. You can configure the device to check whether a peer's certificate has been revoked, using the OCSP. When OCSP is enabled, the device queries the OCSP server for revocation information whenever a
peer certificate is received (TLS client mode, or TLS server mode with mutual authentication).

**Note:**
- 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.

- Private key - externally created and then uploaded to device.
- Different levels of security strength (key size) per TLS certificate.
- 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).

To use a TLS Context for SIPS, assign it to a Proxy Set and/or SIP Interface associated with the IP Group for which you want to employ TLS 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. To configure Proxy Sets, see "Configuring Proxy Sets" on page 359.
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. To configure SIP Interfaces, see "Configuring SIP Interfaces" on page 337.
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.

The following procedure describes how to configure a TLS Context through the Web interface. You can also configure it through ini file (TLSContexts) or CLI (configure system > tls <ID>).

**To configure a TLS Context:**
1. Open the TLS Contexts table (Setup menu > IP Network tab > Security folder > TLS Contexts).
2. Click New to add a new TLS Context or Edit to modify the default TLS Context at Index 0; the following dialog box appears (for adding a TLS Context):
3. Configure the TLS Context according to the parameters described in the table below.

4. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

<table>
<thead>
<tr>
<th>Table 12-1: TLS Contexts Parameter Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>Index</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>TLS Version</td>
</tr>
<tr>
<td>DTLS Version</td>
</tr>
</tbody>
</table>

**TLS Contexts_Index**

- **tls <ID>**
- **[TLSContexts_Index]**

**TLS Version**

- **tls-version**
- **[TLSContexts_TLSVersion]**

**DTLS Version**

- **[TLSContexts_DTLSVersion]**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cipher Server ciphers-server [TLSContexts_ServerCipherString]</td>
<td>Defines the supported cipher suite for the TLS server (in OpenSSL cipher list format). The default is AES:RC4. For valid values, visit the OpenSSL website at <a href="https://www.openssl.org/docs/man1.0.2/apps/ciphers.html">https://www.openssl.org/docs/man1.0.2/apps/ciphers.html</a>.</td>
</tr>
<tr>
<td>Cipher Client ciphers-client [TLSContexts_ClientCipherString]</td>
<td>Defines the supported cipher suite for TLS clients. The default is DEFAULT. For possible values and additional details, visit the OpenSSL website at <a href="https://www.openssl.org/docs/man1.0.2/apps/ciphers.html">https://www.openssl.org/docs/man1.0.2/apps/ciphers.html</a>.</td>
</tr>
</tbody>
</table>
| Strict Certificate Extension Validation require-strict-cert [TLSContexts_RequireStrictCert] | Enables the validation of the extensions (keyUsage and extendedKeyUsage) of peer certificates. The 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 |
| DH Key Size dh-key-size [TLSContexts_DHKeySize] | Defines the Diffie-Hellman (DH) key size (in bits). DH is an algorithm used chiefly for exchanging cryptography keys used in symmetric encryption algorithms such as AES.  
  - [1024] 1024 (default)  
  - [2048] 2048 |
| OCSP Server ocsp-server [TLSContexts_OcspEnable] | Enables or disables certificate checking using OCSP.  
  - [0] Disable (default)  
  - [1] Enable |
| Primary OCSP Server ocsp-server-primary [TLSContexts_OcspServerPrimary] | Defines the IP address (in dotted-decimal notation) of the primary OCSP server. The default is 0.0.0.0. |
| Secondary OCSP Server ocsp-server-secondary [TLSContexts_OcspServerSecondary] | Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional). The default is 0.0.0.0. |
| OCSP Port ocsp-port [TLSContexts_OcspServerPort] | Defines the OCSP server's TCP port number. The default port is 2560. |
| OCSP Default Response ocsp-default-response [TLSContexts_OcspDefaultResponse] | Determines whether the device allows or rejects peer certificates if it cannot connect to the OCSP server.  
  - [0] Reject (default)  
  - [1] Allow |
12.2 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). The DNS name is used to access the device and therefore, must be listed in the server certificate.
2. Open the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 105).
3. In the table, select the required TLS Context, and then click the Change Certificate link located below the table; the Change Certificates page appears.
4. Under the Certificate Signing Request group, do the following:
   a. In the 'Subject Name [CN]' field, enter the DNS name.

   Note: For the Subject Name, you can use the IP address of the device instead of a qualified DNS name. However, it is not recommended since the IP address is subject to change and may not uniquely identify the device.

   a. From the 'Signature Algorithm' drop-down list, select the hash function algorithm (SHA-1, SHA-256, or SHA-512) with which to sign the certificate.
   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 12-1: Certificate Signing Request Group*

<table>
<thead>
<tr>
<th>Subject Name [CN]</th>
<th>Headquarters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Unit [OU]</td>
<td>Corporate</td>
</tr>
<tr>
<td>Company name [O]</td>
<td>Poughkeepsie</td>
</tr>
<tr>
<td>Locality or city name [L]</td>
<td>New York</td>
</tr>
<tr>
<td>State [ST]</td>
<td>US</td>
</tr>
<tr>
<td>Country code [C]</td>
<td></td>
</tr>
<tr>
<td>Signature Algorithm</td>
<td>SHA-1</td>
</tr>
</tbody>
</table>

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). Make sure 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/MQswCQYDVQQGEwJGUjETMBEGA1UEChMKQ2VydGlwb3N0ZTEbMBkGA1UEAxMSQ2VydGlwb3N0ZSBTZXJ2ZXJ2ZXVyMB4XDTk4MDYyNDA4MDAwMFoXDTE4MDYyNDA4MDAwMFowPzELMAkGA1UEBhMCRlIxEzARBgNVBAoTCkNlcnRpcG9zdGUxGzAZBgNVBAMTEkNlcnRpcG9zdGUgU2VydmV1cjCCASEwDQYJKoZIhvcNAQEBBQADggEOADCCAQkCggEAPqd4Mz+iR4spWldGRx8bQrh2k0WnNm`+Yhb7+4Q67ecf1jhanH7GcN/SXsfx7jJpreeWULf7v7Cvpr4R7q1JcmhHIntmf7JPM5n6cDBv17uSW63er7NkVnMFFhwR1QaGFLMybFkzaeGrvFmdk3JRefiXdmuOe+FhJgHYezYHF44LvPRwshrzi9+Ag3o8pWdgwJuzVD1UPF1jMa+LPwvRExFcUW+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 **Load File**:

*Figure 12-2: Loading Device Certificate*

8. After the certificate successfully loads to the device, save the configuration with a device reset.
9. Verify that the private key is correct:
   a. Open the TLS Contexts table.
   b. Select the required TLS Context index row.
   c. Click the **Certificate Information** link located below the table.
   d. Make sure that the 'Private key' field displays "OK"; otherwise (i.e., displays "Does not match certificate"), consult with your security administrator.

   **Figure 12-3: Verifying Private Key**

   ![Certificate Information]

<table>
<thead>
<tr>
<th>GENERAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate subject: CN=ACL_5469038</td>
</tr>
<tr>
<td>Certificate issuer: CN=ACL_5469038</td>
</tr>
<tr>
<td>Time to expiration: 7290 days</td>
</tr>
<tr>
<td>Key size: 1024 bits</td>
</tr>
<tr>
<td>Private key: OK</td>
</tr>
</tbody>
</table>

   **Note:**
   - The certificate replacement process can be repeated whenever necessary (e.g., the new certificate expires).
   - You can also load the device certificate through the device's Automatic Provisioning mechanism, using the HTTPSCertFileName ini file parameter.

### 12.3 Viewing Certificate Information

You can view information of TLS certificates installed on the device per TLS Context.

- **To view certificate information:**
  1. Open the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 105).
2. Select a TLS Context row, and then click the Certificate Information link located below the table; the Certificate Information page appears:

Figure 12-4: Viewing Certificate Information

<table>
<thead>
<tr>
<th>PRIVATE KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key size: 1024 bits</td>
</tr>
<tr>
<td>Status: OK</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CERTIFICATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certificate:</td>
</tr>
<tr>
<td>Data:</td>
</tr>
<tr>
<td>Version: 1 (0x0)</td>
</tr>
<tr>
<td>Serial Number: 0 (0x0)</td>
</tr>
<tr>
<td>Signature Algorithm: sha256WithRSAEncryption</td>
</tr>
<tr>
<td>Issuer: CN=ACL_5967925</td>
</tr>
<tr>
<td>Validity:</td>
</tr>
<tr>
<td>Not Before: Jan 1 00:00:38 2010 GMT</td>
</tr>
<tr>
<td>Not After: Dec 27 00:38:38 2029 GMT</td>
</tr>
<tr>
<td>Subject: CN=ACL_5967925</td>
</tr>
<tr>
<td>Signature Algorithm: rsaEncryption</td>
</tr>
<tr>
<td>Public-Key: (1024 bit)</td>
</tr>
<tr>
<td>Modulus:</td>
</tr>
<tr>
<td>Exponent: 65537 (0x10001)</td>
</tr>
<tr>
<td>Signature Algorithm: sha256WithRSAEncryption</td>
</tr>
</tbody>
</table>
12.4 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 table (see “Configuring TLS Certificate Contexts” on page 105).

3. In the table, select the required TLS Context index row, and then click the Change Certificate link located below the table; the Change Certificates page appears.

4. Scroll down to the Upload Certificate Files From Your Computer group:

   Figure 12-5: Upload Certificate Files from your Computer Group

   5. (Optional) In the 'Private key pass-phrase' field, enter the password (passphrase) of the encrypted private key file. If not encrypted with a passphrase, leave the field blank.

   6. Click the Browse button corresponding to the 'Send Private Key file ...' text, navigate to the private key file (Step 1), and then click Load File.

   7. If the security administrator has provided you with a device certificate file, load it using the Browse button corresponding to the 'Send Device Certificate file ...' text.

   8. After the files successfully load to the device, save the configuration with a device reset.

9. Verify that the private key is correct:
   a. Open the TLS Contexts table.
   b. Select the required TLS Context index row.
   c. Click the Certificate Information link located below the table.
   d. Make sure that the 'Private key' field displays "OK"; otherwise (i.e., displays "Does not match certificate"), consult with your security administrator.

   Figure 12-6: Verifying Private Key
12.5 Generating Private Keys for TLS Contexts

You can let the device generate the private key for a TLS Context. 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 table (see "Configuring TLS Certificate Contexts" on page 105).
2. In the table, select the required TLS Context index row, and then click the Change Certificates link located below the table; the Change Certificates page appears.
3. Scroll down to the Generate New Private Key and Self-signed Certificate group:

Figure 12-7: Generate new private key and self-signed certificate Group

4. From the 'Private Key Size' drop-down list, select the desired private key size (in bits) for RSA public-key encryption for newly self-signed generated keys:
   - 512
   - 768
   - 1024 (default)
   - 2048
   - 4096
5. (Optional) In the ‘Private key pass-phrase’ field, enter a password (passphrase) to encrypt the private key file. If you don't want to encrypt the file, make the field blank. The default passphrase is "audc". The passphrase can be up to 32 characters.
6. Click Generate Private-Key; a message appears requesting you to confirm key generation.
7. Click **OK** to confirm key generation; the device generates a new private key, indicated by a message in the **Certificate Signing Request** group:

**Figure 12-8: Indication of Newly Generated Private Key**

<table>
<thead>
<tr>
<th>Certificate Signing Request</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subject Name [CN]</strong></td>
</tr>
<tr>
<td><strong>Organizational Unit [OU]</strong></td>
</tr>
<tr>
<td><strong>Company name [O]</strong></td>
</tr>
<tr>
<td><strong>Locality or city name [L]</strong></td>
</tr>
<tr>
<td><strong>State [ST]</strong></td>
</tr>
<tr>
<td><strong>Country code [C]</strong></td>
</tr>
<tr>
<td><strong>Signature Algorithm</strong></td>
</tr>
</tbody>
</table>

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

**A new 512-bit Private-Key was generated for Context-ID=2**

Please save the configuration.

8. Continue with the certificate configuration by either creating a CSR or generating a new self-signed certificate.

9. Save the configuration with a device reset for the new certificate to take effect.

### 12.6 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 of the following:
   - You have a unique DNS name for the device (e.g., dns_name.corp.customer.com). The 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 table (see "Configuring TLS Certificate Contexts" on page 105).

3. In the table, select the required TLS Context index row, and then click the **Change Certificate** link located below the table; the Change 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 12-9: Generate new private key and self-signed certificate Group](image)

6. Click **Generate Self-Signed Certificate**; a message appears requesting you to confirm generation.

7. Click **OK** to confirm generation; the device generates a new self-signed certificate displaying the new subject name, indicated by a message in the **Certificate Signing Request** group:

![Figure 12-10: Generated Self-Signed Certificate](image)

8. Save the configuration with a device reset for the new certificate to take effect.

### 12.7 Importing Certificates 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. 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.
You can also import multiple TLS root certificates in bulk from a single file. Each certificate in the file must be Base64 encoded (PEM). When copying-and-pasting the certificates into the file, each Base64 ASCII encoded certificate string must be enclosed between "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----".

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

To import certificates into device’s Trusted Root Certificate Store:
1. Open the TLS Contexts table (see “Configuring TLS Certificate Contexts” on page 105).
2. In the table, select the required TLS Context index row, and then click the Trusted Root Certificates link located below the table; the Trusted Certificates table appears.
3. Click the Import button, and then browse to and select the certificate file.
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 folder 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**.

### 12.8 Configuring Mutual TLS Authentication

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

#### 12.8.1 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 calls by enabling mutual authentication on the SIP Interface associated with the calls. 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" (SIPSRequireClientCertificate) parameter (see "Configuring TLS for SIP" on page 170).

- **To configure mutual TLS authentication for SIP messaging:**
  1. Enable two-way authentication on the specific SIP Interface:
     a. In the SIP Interfaces table (see "Configuring SIP Interfaces" on page 337), configure the 'TLS Mutual Authentication' parameter to **Enable** for the specific SIP Interface.
     b. Reset the device with a save-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 Certificates table (certificate root store) so that the device can authenticate the client (see "Importing Certificates and Certificate Chain into Trusted Certificate Store" on page 116).
     - Make sure that the TLS certificate is signed by a CA that the SIP client trusts so that the client can authenticate the device.

#### 12.8.2 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 Certificates table (certificate root store). The Trusted Root Certificate file may contain more than one CA certificate combined, using a text editor.
To enable mutual TLS authentication for HTTPS:

1. On the Web Settings page (see "Configuring Secured (HTTPS) Web" on page 80), configure the 'Secured Web Connection (HTTPS)' parameter to **HTTPS Only**. The setting ensures that you have a method for accessing the device in case the client certificate doesn't work. Restore the previous setting after testing the configuration.

2. In the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 105), select the required TLS Context row, and then click the **Trusted Root Certificates** link located below the table; the Trusted Certificates table appears.

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

4. Wait until the import operation finishes successfully.

5. On the Web Settings page, configure the 'Require Client Certificates for HTTPS connection' parameter to **Enable**.

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

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

- If the user has a client certificate from a CA that is listed in the Trusted Root Certificate file, the connection is accepted and the user is prompted for the system password.

- If both the CA certificate and the client certificate appear in the Trusted Root Certificate file, the user is not prompted for a password (thus, providing a single-sign-on experience - the authentication is performed using the X.509 digital signature).

- If the user does not have a client certificate from a listed CA or does not have a client certificate, the connection is rejected.

**Note:**

- 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 with your security administrator.

- The root certificate can also be loaded through the device's Automatic Provisioning mechanism, 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 105).
12.9 Configuring TLS Server Certificate Expiry Check

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

➢ To configure TLS certificate expiry checks and notification:

1. Open the TLS Contexts table (see “Configuring TLS Certificate Contexts” on page 105).
2. Select the required TLS Context index row, and then click the Change Certificate link located below the table; the Change Certificates page appears.
3. Scroll down the page to the TLS Expiry Settings group:

   **Figure 12-13: TLS Expiry Settings Group**

<table>
<thead>
<tr>
<th>TLS Expiry Settings</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS Expiry Check Start (days)</td>
<td>60</td>
</tr>
<tr>
<td>TLS Expiry Check Period (days)</td>
<td>7</td>
</tr>
</tbody>
</table>

4. In the 'TLS Expiry Check Start' field, enter the number of days before the installed TLS server certificate is to expire when the device sends an SNMP trap event to notify of this.
5. 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.
6. Click the Submit TLS Expiry Settings button.
13 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.

13.1 Configuring Automatic Date and Time using SNTP

The device's Simple Network Time Protocol (SNTP) client functionality generates requests and reacts to the resulting responses using the NTP Version 3 protocol definitions (according to RFC 1305). Through these requests and responses, the device, as an NTP client, synchronizes the system time to a time source within the network, thereby eliminating any potential issues should the local system clock "drift" during operation. The NTP client follows a simple process in managing system time: 1) the NTP client requests an NTP update, 2) receives an NTP response and then 3) 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, the update interval is every 24 hours based on when the system was restarted.

You can also configure the device to authenticate and validate 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 through the Web interface. For detailed descriptions of the configuration parameters, see "NTP and Daylight Saving Time Parameters" on page 855.

➢ To configure SNTP through the Web interface:

1. Open the Time & Date page (Setup menu > Administration tab > Time & Date), and then scroll down to the NTP Server group:

   Figure 13-1: Configuring NTP Server

<table>
<thead>
<tr>
<th>NTP Server</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary NTP Server Address (IP or FQDN)</td>
<td>0.0.0.0</td>
</tr>
<tr>
<td>Secondary NTP Server Address (IP or FQDN)</td>
<td></td>
</tr>
<tr>
<td>NTP Update Interval</td>
<td>Hours: 24 Minutes: 0</td>
</tr>
<tr>
<td>NTP Authentication Key Identifier</td>
<td>0</td>
</tr>
<tr>
<td>NTP Authentication Secret Key</td>
<td></td>
</tr>
</tbody>
</table>

2. Configure the NTP server address:
   - In the 'Primary NTP Server Address' (NTPServerIP) field, configure the primary NTP server's address (IP or FQDN).
   - (Optional) In the 'Secondary NTP Server Address' (NTPSecondaryServerIP) field, configure the backup NTP server.

3. In the 'NTP Update Interval' (NTPUpdateInterval) field, configure the period after which the date and time of the device is updated.

4. 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.
5. Verify that the device has received the correct date and time from the NTP server. The date and time is displayed in the 'UTC Time' read-only field under the Time Zone group.

**Note:** If the device does not receive a response from the NTP server, it polls the NTP server for 10 minutes. If there is still no response after this duration, the device declares the NTP server as unavailable and raises an SNMP alarm (acNTPServerStatusAlarm). The failed response could be due to incorrect configuration.

### 13.2 Configuring Date and Time Manually

You can manually configure the date and time of the device instead of using an NTP server (as described in "Configuring Automatic Date and Time using SNTP" on page 121).

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

1. Open the Time & Date page (Setup menu > Administration tab > Time & Date), and then scroll down to the Local Time group:

   ![Figure 13-2: Configuring Manual Date and Time](image)

2. Configure the current date and time of the geographical location in which the device is installed:
   - **Date:**
     - 'Year' in yyyy format (e.g., "2015")
     - 'Month' in mm format (e.g., "3" for March)
     - 'Day' in dd format (e.g., "27")
   - **Time:**
     - 'Hours' in 24-hour format (e.g., "4" for 4 am)
     - 'Minutes' in mm format (e.g., "57")
     - 'Seconds' in ss format (e.g., "45")

3. Click **Apply**; the date and time is displayed in the 'UTC Time' read-only field.

**Note:**
- If the device is configured to obtain date and time from an NTP server, the fields under the Local Time group are read-only, displaying the date and time received from the NTP server.
- After performing a hardware reset, the date and time are returned to default values and thus, you should subsequently update the date and time.

### 13.3 Configuring the Time Zone

You can configure the time zone in which the device is deployed. This is referred to as the Coordinated Universal Time (UTC) time offset and defines how many hours the device is
13. Date and Time

from Greenwich Mean Time (GMT). For example, Germany Berlin is one hour ahead of GMT (UTC/GMT is +1 hour) and therefore, you would configure the offset to "1". USA New York is five hours behind GMT (UTC/GMT offset is -5 hours) and therefore, you would configure the offset as a minus value "-5".

➢ To configure the time zone:

1. Open the Time & Date page (Setup menu > Administration tab > Time & Date), and then scroll down to the Time Zone group:

   ![Figure 13-3: Configuring UTC Offset](image)

2. In the 'UTC Offset' fields (NTPServerUTCOffset), configure the time offset in relation to the UTC. For example, if your region is GMT +1 (an hour ahead), enter "1" in the 'Hours' field.

3. Click Apply; the updated time is displayed in the 'UTC Time' read-only field and the fields under the Local Time group.

13.4 Configuring Daylight Saving Time

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

➢ To configure DST through the Web interface:

1. Open the Time & Date page (Setup menu > Administration tab > Time & Date), and then scroll down to the Time Zone group:

   ![Figure 13-4: Configuring Daylight Savings](image)

2. From the 'Day Light Saving Time' (DayLightSavingTimeEnable) drop-down list, select Enable.

3. From the 'DST Mode' drop-down list, select the range type for configuring the start and end dates for DST:
• **Day of year**: The range is configured by exact date (day number of month), for example, from March 30 to October 30. If 'DST Mode' is set to **Day of year**, in the 'Start Time' (DayLightSavingTimeStart) and 'End Time' (DayLightSavingTimeEnd) drop-down lists, configure the period for which DST is relevant.

• **Day of month**: The range is configured by month and day type, for example, from the last Sunday of March to the last Sunday of October. If 'DST Mode' is set to **Day of month**, in the 'Day of Month Start' and 'Day of Month End' drop-down lists, configure the period for which DST is relevant.

4. In the 'Offset' (DayLightSavingTimeOffset) field, configure the DST offset in minutes.

5. If the current date falls within the DST period, verify that it has been successful applied to the device's current date and time. You can view the device's date and time in the 'UTC Time' read-only field.
Part IV

General VoIP Configuration
14 Network

This section describes network-related configuration.

14.1 Building and Viewing your Network Topology

The Network view lets you easily build and view your voice network topology entities, including IP network interfaces, Ethernet Devices (VLANs), Ethernet Groups, and physical Ethernet ports. The Topology view graphically displays these entities and the associations between them, giving you a better understanding of your network topology and configuration. You can use the Network view as an alternative to configuring the entities in their respective Web pages or you can use it in combination.

➢ To access the Network view:

- Click the Network View home icon (Setup menu > IP Network tab > Network View).

The areas of the Network view is shown in the example below and described in the subsequent table.

Note: The below figure is used only as an example; your device may show different Ethernet Groups and Ethernet ports.

Figure 14-1: Network View (Example)
### Table 14-1: Description of Network View

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | Configures and displays IP Interfaces. The IP Interface appears as an icon, displaying the application type ("OCM" for OAMP, "C" for Control, and "M" for Media), row index number, name, and IP address, as shown in the example below:  
If you click the icon, a drop-down menu appears listing the following commands:  
- **Edit**: Opens a dialog box in the IP Interfaces table to modify the IP Interface.  
- **View List**: Opens the IP Interfaces table, allowing you to configure IP Interfaces.  
- **Delete**: Opens the IP Interfaces table where you are prompted to confirm deletion of the IP Interface.  
**To add an IP Interface:**  
1. Click **Add IP Interface**: the IP Interfaces table opens with a new dialog box for adding an IP Interface to the next available index row.  
2. Configure the IP Interface as desired, and then click **Apply**: the IP Interfaces table closes and you are returned to the Network View, displaying the newly added IP Interface.  
For more information on configuring IP Interfaces, see "Configuring IP Network Interfaces" on page 137. |
| 2      | Configures and displays Ethernet Devices. The Ethernet Device appears as an icon, displaying the row index number, name, VLAN ID and whether its tagged or untagged, as shown in the example below:  
If you click the icon, a drop-down menu appears listing the following commands:  
- **Edit**: Opens a dialog box in the Ethernet Devices table to modify the Ethernet Device.  
- **View List**: Opens the Ethernet Devices table, allowing you to configure all Ethernet Devices.  
- **Delete**: Opens the Ethernet Devices table where you are prompted to confirm deletion of the Ethernet Device.  
**To add an Ethernet Device:**  
3. Click **Add VLAN**: the Ethernet Devices table opens with a new dialog box for adding an Ethernet Device to the next available index row.  
4. Configure the Ethernet Devices as desired, and then click **Apply**: the Ethernet Devices table closes and you are returned to the Network View, displaying the newly added |
### Configuring Ethernet Groups

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Ethernet Device. For more information on configuring Ethernet Devices, see &quot;Configuring Underlying Ethernet Devices&quot; on page 134.</td>
</tr>
<tr>
<td>3</td>
<td>Configures and displays Ethernet Groups. The Ethernet Groups appear as icons, displaying the row index number and name, as shown in the example below:</td>
</tr>
</tbody>
</table>

- Ethernet ports associated with Ethernet Groups are indicated by lines connecting between them, as shown in the example below:

To edit an Ethernet Group:

1. Click the Ethernet Group icon, and then from the drop-down menu, choose **Edit**; the Ethernet Groups table opens with a dialog box for editing the Ethernet Group.
2. Configure the Ethernet Group as desired, and then click **Apply**; the Ethernet Groups table closes and you are returned to the Network View.

For more information on configuring IP Interfaces, see "Configuring Ethernet Port Groups" on page 132.

To open the Ethernet Groups table, click any Ethernet Group icon, and then from the drop-down menu, choose **View List**. You can then view and edit all the Ethernet Groups in the table.
4 Configures and displays the device's Ethernet ports.

**To configure an Ethernet port:**
1. Click the required port icon, and then from the drop-down menu, choose **Edit**; the Physical Ports table opens with a dialog box for editing the Ethernet port.
2. Configure the Ethernet Port as desired, and then click **Apply**; the Physical Ports table closes and you are returned to the Network View.

For more information on configuring Ethernet ports, see "Configuring Underlying Ethernet Devices" on page 134.

The Ethernet ports appear as icons, displaying the row index number, description, and port string number, as shown in the example below:

The connectivity status of the port is indicated by the color of the icon:
- Green: Network connectivity exists through port (port connected to network).
- Red: No network connectivity through port (e.g., cable disconnected).

To refresh the status indication, click the **Refresh Network View** button (described below in Item #5).

To open the Physical Ports table, click any port icon, and then from the drop-down menu, choose **View List**. You can then view and edit all the ports in the table.

5 If you keep the Network view page open for a long time, you may want to click the **Refresh Network View** button to refresh the connectivity status display of the Ethernet ports.

### 14.2 Configuring Physical Ethernet Ports

The Physical Ports table lets you configure the device's Ethernet ports. This includes configuring port speed and duplex mode (half or full), and a brief description of the port. The table also displays the status of the port as well as the port group (*Ethernet Group*) to which the port belongs. For more information on Ethernet Groups, see "Configuring Ethernet Port Groups" on page 132.

The names of the ports displayed in the device's management tools (e.g., Web interface) are different to the labels of the physical ports on the chassis. The figure below shows the mapping between the two:

You can also view the mapping of the ports, using the following CLI command:

```
# show network physical-port
```

**Note:** Each Ethernet port must have a unique VLAN ID in scenarios where the ports are connected to the same switch.

The following procedure describes how to configure Ethernet ports through the Web interface. You can also configure it through ini file (PhysicalPortsTable) or CLI (configure network > physical-port).

- **To configure the physical Ethernet ports:**
  1. Open the Physical Ports table (Setup menu > IP Network tab > Core Entities folder >
2. Select a port that you want to configure, and then click **Edit**; the following dialog box appears:

![Port Configuration Dialog Box]

3. Configure the port according to the parameters described in the table below.

4. Click **Apply**, and then save your settings to flash memory.

### Table 14-2: Physical Ports Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>(Read-only) Displays the index number for the table row.</td>
</tr>
<tr>
<td>Name</td>
<td>(Read-only) Displays the Ethernet port number. See the figure in the beginning of this section for the mapping between the GUI port number and the physical port on the chassis.</td>
</tr>
<tr>
<td>Description</td>
<td>Defines a description of the port. By default, the value is &quot;User Port #&lt;row index&gt;&quot;. Note: Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>Mode</td>
<td>(Read-only) Displays the mode of the port.</td>
</tr>
<tr>
<td>Speed and Duplex</td>
<td>Defines the speed and duplex mode of the port.</td>
</tr>
<tr>
<td><strong>Ethernet Group</strong></td>
<td></td>
</tr>
<tr>
<td>Member of Ethernet Group</td>
<td>(Read-only) Displays the Ethernet Group to which the port belongs.</td>
</tr>
<tr>
<td>Group Status</td>
<td>(Read-only) Displays the status of the port:</td>
</tr>
</tbody>
</table>

To assign the port to a different Ethernet Group, see "Configuring Ethernet Port Groups" on page 132.
### 14.3 Configuring Ethernet Port Groups

The Ethernet Groups table lets you configure Ethernet Groups. An Ethernet Group represents a physical Ethernet port(s) on the device. You can assign an Ethernet Group with one, two, or no ports (members). When two ports are assigned to an Ethernet Group, 1+1 Ethernet port redundancy can be implemented in your network. This provides port redundancy within the Ethernet Group, whereby if a port is disconnected the device switches over to the other port in the Ethernet Group. If you configure an Ethernet Group with only one port, the Ethernet Group operates as a single port (no redundancy).

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

The maximum number of Ethernet Groups that you can configure is the same as the number of Ethernet ports provided by the device. Thus, the device supports up to 12 Ethernet Groups where each contains one port, or 6 Ethernet Groups where each contain two ports. By default, each Ethernet Group is assigned one port.

You can assign Ethernet ports to IP network interfaces. This is done by first configuring an Ethernet Device with the required Ethernet Group containing the port or ports (see "Configuring Underlying Ethernet Devices" on page 134). Then by assigning the Ethernet Device to the IP network interface in the IP Interfaces table (see "Configuring IP Network Interfaces" on page 137). This enables physical separation of network interfaces, providing a higher level of segregation of sub-networks. Equipment connected to different physical ports is not accessible to one another; the only connection between them can be established by cross connecting them with media streams (VoIP calls).

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

#### Note:

- If you want to assign a port to a different Ethernet Group, you must first remove the port from its current Ethernet Group. To remove the port, configure the 'Member' field so that no port is selected or select a different port.
- When implementing 1+1 Ethernet port redundancy, each port in the Ethernet Group (port pair) must be connected to a different switch (but in the same subnet).
- For Mediant Virtual Edition (VE), Ethernet port redundancy is not relevant as the virtual NIC ("port") is logical and always available. Therefore, it is sufficient to configure the Ethernet Group with only one port member. To employ Ethernet port redundancy, you need to configure the virtual switch of the hypervisor for Ethernet port redundancy (or bonding) with the physical NICs. Refer to your hypervisor’s support material on how to do this.

#### To configure Ethernet Groups:

1. Open the Ethernet Groups table (Setup menu > IP Network tab > Core Entities)
1. Select the Ethernet Group that you want to configure, and then click **Edit**; the following dialog box appears:

![Ethernet Groups Dialog Box](image)

2. Configure the Ethernet Group according to the parameters described in the table below.

3. Click **Apply**, and then save your settings to flash memory.

### Table 14-3: Ethernet Groups Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>(Read-only) Displays the index number for the table row.</td>
</tr>
<tr>
<td>Name group [EtherGroupTable_Group]</td>
<td>(Read-only) Displays the Ethernet Group number.</td>
</tr>
<tr>
<td>Mode mode [EtherGroupTable_Mode]</td>
<td>Defines the mode of operation of the ports in the Ethernet Group. This applies only to Ethernet Groups containing two ports.</td>
</tr>
<tr>
<td></td>
<td>[3] 2RX/1TX = Both ports in the Ethernet Group can receive packets, but only one port can transmit. The transmitting port is determined arbitrarily by the device. If the selected port fails at a later stage, a switchover to the redundant port is done, which begins to transmit and receive.</td>
</tr>
<tr>
<td></td>
<td>[4] 2RX/2TX = Both ports in the Ethernet Group can receive and transmit packets. This option is applicable only to the Maintenance interface for High Availability (HA) deployments. For more information, see Network Topology Types and Rx/Tx Ethernet Port Group Settings on page 603.</td>
</tr>
<tr>
<td></td>
<td>[5] Single = (Default) Select this option if the Ethernet Group contains only one port.</td>
</tr>
<tr>
<td></td>
<td>[6] None = Select this option to remove all ports from the Ethernet Group.</td>
</tr>
<tr>
<td>Note:</td>
<td>Select the <strong>2RX/1TX</strong> option. In such a setup, the ports can be connected to the same LAN switch or each to a different switch where both are in the same subnet.</td>
</tr>
<tr>
<td></td>
<td>It is recommended to use the <strong>2RX/1TX</strong> option. In such a setup, the ports can be connected to the same LAN switch or each to a different switch where both are in the same subnet.</td>
</tr>
<tr>
<td>Member 1</td>
<td>Assigns the first port to the Ethernet Group. To assign no port, set this parameter to None.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>member1 [EtherGroupTable_Member1]</td>
<td>field to None. Note: Before you can re-assign a port to a different Ethernet Group, you must first remove the port from its current Ethernet Group. To remove the port, either set this field to None or to a different port.</td>
</tr>
<tr>
<td>Member 2 member2 [EtherGroupTable_Member2]</td>
<td>Assigns the second port to the Ethernet Group. To assign no port, set this field to None. Note: Before you can re-assign a port to a different Ethernet Group, you must first remove the port from its current Ethernet Group. To remove the port, either set this field to None or to a different port.</td>
</tr>
</tbody>
</table>

### 14.4 Configuring Underlying Ethernet Devices

The Ethernet Devices table lets you configure up to 1,024 Ethernet Devices. An Ethernet Device represents a Layer-2 bridging device and is assigned a VLAN ID and an Ethernet Group (Ethernet port group). Multiple Ethernet Devices can be associated with the same Ethernet Group. The Ethernet Device (VLAN) can be configured with a VLAN tagging policy, which determines whether the Ethernet Device accepts tagged or untagged packets received on the Ethernet port associated with the Ethernet Device.

Once configured, assign the Ethernet Device to an IP network interface in the IP Interfaces table ('Underlying Device' field) and/or with a static route in the Static Routes table ('Device Name' field). You can assign the same Ethernet Device to multiple IP network interfaces and thereby, implement multi-homing (multiple addresses on the same interface/VLAN).

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

- Name: "vlan 1"
- VLAN ID: 1
- Ethernet Group: GROUP 1
- Tagging Policy: Untagged
- MTU: 1500

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

You can view configured Ethernet Devices that have been successfully applied to the device (saved to flash) in the Ethernet Device Status table. This page is accessed by clicking the Ethernet Device Status Table button located at the bottom of the Ethernet Devices table. The Ethernet Device Status table can also be accessed from the Navigation tree (see "Viewing Ethernet Device Status" on page 755).

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

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

➢ **To configure an Ethernet Device:**

1. Open the Ethernet Devices table (Setup menu > IP Network tab > Core Entities)
folder > Ethernet Devices).

2. Click New; the following dialog box appears:

![Ethernet Devices dialog box]

3. Configure an Ethernet Device according to the parameters described in the table below.

4. Click Apply.

**Table 14-4: Ethernet Devices Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index [DeviceTable_Index]</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name [DeviceTable_DeviceName]</td>
<td>Defines a name for the Ethernet Device. The name is used to associate the Ethernet Device with an IP network interface in the IP Interfaces table ('Underlying Device' field - see &quot;Configuring IP Network Interfaces&quot; on page 137) and/or with a static route in the Static Routes table ('Device Name' field - see &quot;Configuring Static IP Routing&quot; on page 144).</td>
</tr>
<tr>
<td>VLAN ID [DeviceTable_VlanID]</td>
<td>Defines a VLAN ID for the Ethernet Device. The valid value is 1 to 3999. The default is 1. <strong>Note:</strong> You can configure the same VLAN ID for multiple Ethernet Devices.</td>
</tr>
<tr>
<td>Underlying Interface [DeviceTable_UnderlyingInterface]</td>
<td>Assigns an Ethernet Group to the Ethernet Device. To configure Ethernet Groups, see Configuring Ethernet Port Groups on page 132. <strong>Note:</strong> The parameter is mandatory.</td>
</tr>
</tbody>
</table>
| Tagging [DeviceTable_Tagging] | Defines VLAN tagging for the Ethernet Device.  
  - [0] Untagged = (Default for pre-configured Ethernet Device) The Ethernet Device accepts untagged packets and packets with the same VLAN ID as the Ethernet Device. Incoming untagged packets are assigned the VLAN ID of the Ethernet Device. The Ethernet Device sends these VLAN packets untagged (i.e., removes the VLAN ID).  
  - [1] Tagged = (Default for new Ethernet Devices) The Ethernet Device accepts packets that have the same VLAN ID as the
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Device and sends packets with this VLAN ID. For all Ethernet Devices that are associated with the same Ethernet Group (see 'Underlying Interface' parameter above) and configured to <strong>Tagged</strong>, incoming untagged packets received on this Ethernet Group are discarded. <strong>Note:</strong> Only one Ethernet Device can be configured as Untagged per associated Ethernet Group. In other words, if multiple Ethernet Devices are associated with the same Ethernet Group, only one of these Ethernet Devices can be configured to <strong>Untagged</strong>; all the others must be configured to <strong>Tagged</strong>.</td>
<td></td>
</tr>
</tbody>
</table>
| **MTU**<br>mtu [DeviceTable_MTU] | Defines the Maximum Transmission Unit (MTU) in bytes per VLAN (Ethernet Device). The valid value is 68 to 65,535 (subject to hardware NIC limitations which depend on NIC manufacturer and model). The default is 1,500. **Note:**  
  - MTU is not applicable to SBC Direct Media traffic and to debug recording traffic.  
  - If your first Ethernet Device is configured with an untagged VLAN, its MTU value is the maximum MTU that can be configured for all other Ethernet Devices that are associated with the same Ethernet Group. In other words, if you configure additional Ethernet Devices (tagged VLANs) that are associated with the same Ethernet Group, their MTUs must be equal to or less than the MTU of the first Ethernet Device (untagged VLAN). For example, if the untagged VLAN is configured with MTU of 100 bytes, you can configure a tagged VLAN with an MTU value of either 100 bytes or less.  
  - If your first Ethernet Device is configured with a tagged VLAN and you later configure an additional Ethernet Device with an untagged VLAN that is associated with the same Ethernet Group, the MTU of the untagged VLAN must be equal to or greater than the highest MTU value configured out of all the Ethernet Devices (VLANs) associated with the Ethernet Group. For example, if VLAN 1 is configured with the highest MTU (100 bytes) out of all your VLANs, you can configure an untagged VLAN with an MTU value of either 100 bytes or greater. |
14.5 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 14-2: Multiple Network Interfaces

The device is shipped with a default OAMP interface (see "Default OAMP IP Address" on page 37). The IP Interfaces table lets you change this OAMP interface and configure additional network interfaces for control and media, if necessary. You can configure up to 1,024 interfaces, consisting of up to 1,023 Control and Media interfaces including a Maintenance interface if your device is deployed in a High Availability (HA) mode, and 1 OAMP interface. Each IP interface is configured with the following:

- **Application type allowed on the interface:**
  - Control: call control signaling traffic (i.e., SIP)
  - Media: RTP traffic
  - Operations, Administration, Maintenance and Provisioning (OAMP): management (i.e., Web, CLI, and SNMP based management)
Maintenance: This interface is used in HA mode when two devices are deployed for redundancy, and represents one of the LAN interfaces or Ethernet Groups on each device used for the Ethernet connectivity between the two devices. For more information on HA and the Maintenance interface, see Configuring High Availability on page 595.

- IP address (IPv4 or IPv6) and subnet mask (prefix length)
- To configure Quality of Service (QoS), see "Configuring the QoS Settings" on page 155.
- 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
- (Optional) Primary and secondary domain name server (DNS) addresses for resolving FQDNs into IP addresses.
- Ethernet Device: Layer-2 bridging device and assigned a VLAN ID. As the Ethernet Device is associated with an Ethernet Group, this is useful for setting trusted and untrusted networks on different physical Ethernet ports. Multiple entries in the IP Interfaces 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 IP Interfaces table is the Static Routes table, which lets you configure static routing rules for non-local hosts/subnets. For more information, see "Configuring Static IP Routing" on page 144.

Note: Before configuring IP interfaces, it is recommended that you read the IP interface configuration guidelines in "IP Interfaces Table Configuration Guidelines" on page 141.

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

➢ To configure IP network interfaces:

1. Open the IP Interfaces table (Setup menu > IP Network tab > Core Entities folder > IP Interfaces).
2. Click New; the following dialog box appears:

   **Figure 14-3: IP Interfaces Table - Dialog Box**

3. Configure the IP network interface according to the parameters described in the table below.
4. Click Apply.

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

To view configured IP network interfaces that are currently active, click the IP Interface Status Table link located at the bottom of the table. For more information, see "Viewing Active IP Interfaces" on page 755.

**Table 14-5: IP Interfaces Table Parameters Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>network-if [InterfaceTable_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a name for the interface. The valid value is a string of up to 16 characters. If you do not configure a name, the device automatically assigns the name using the syntax &quot;InterfaceTable_&lt;row index&gt;&quot;. For example, if you add a new interface to row index 2, the name is &quot;InterfaceName_2&quot;. The name of the default OAMP interface is &quot;O+M+C+P&quot;. <strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>Application Type</td>
<td>Defines the applications allowed on the IP interface.</td>
</tr>
<tr>
<td>application-type [InterfaceTable_ApplicationTypes]</td>
<td>[0] OAMP = Operations, Administration, Maintenance and Provisioning (OAMP) applications (e.g., Web, Telnet, SSH, and SNMP).</td>
</tr>
<tr>
<td></td>
<td>[1] Media = Media (i.e., RTP streams of voice).</td>
</tr>
<tr>
<td></td>
<td>[2] Control = Call Control applications (e.g., SIP).</td>
</tr>
<tr>
<td></td>
<td>[4] OAMP + Control = OAMP and Call Control applications.</td>
</tr>
<tr>
<td></td>
<td>[6] OAMP + Media + Control = All application types are allowed on the interface.</td>
</tr>
<tr>
<td></td>
<td>[99] MAINTENANCE = Only the Maintenance application for HA is allowed on this interface.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Only one IP network interface can be configured with OAMP in this table. To configure additional management interfaces, see Configuring Additional Management Interfaces on page 71.</td>
</tr>
<tr>
<td>Ethernet Device</td>
<td>Assigns an Ethernet Device to the IP interface. An Ethernet Device</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>underlying-dev [InterfaceTable_UnderlyingDevice]</td>
<td>is a VLAN associated with a physical Ethernet port (Ethernet Group). To configure Ethernet Devices, see Configuring Underlying Ethernet Devices on page 134. By default, no value is defined. <strong>Note:</strong> The parameter is mandatory.</td>
</tr>
<tr>
<td>IP Address</td>
<td></td>
</tr>
<tr>
<td>Interface Mode</td>
<td>Defines the method that the interface uses to acquire its IP address.</td>
</tr>
<tr>
<td>mode [InterfaceTable_InterfaceMode]</td>
<td>[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.</td>
</tr>
<tr>
<td>IP Address ip-address [InterfaceTable_IPAddress]</td>
<td>Defines the IPv4/IPv6 address in dotted-decimal notation. By default, no value is defined. <strong>Note:</strong> The parameter is mandatory.</td>
</tr>
<tr>
<td>Prefix Length [InterfaceTable_PrefixLength]</td>
<td>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. The default is 16.</td>
</tr>
<tr>
<td>Default Gateway gateway [InterfaceTable_Gateway]</td>
<td>Defines the IP address of the default gateway for the IP 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. By default, no value is defined.</td>
</tr>
<tr>
<td>DNS</td>
<td></td>
</tr>
</tbody>
</table>

**User's Manual** 140 **Document #: LTRT-41877**
### 14. Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary DNS primary-dns [InterfaceTable_PrimaryDNSserverIPAddress]</td>
<td>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.</td>
</tr>
<tr>
<td>Secondary DNS secondary-dns [InterfaceTable_SecondaryDNSserverIPAddress]</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 14.5.1 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. For more information on NTP, see "Configuring Automatic Date and Time using SNTP" on page 121.

### 14.5.2 IP Interfaces Table Configuration Guidelines

Adhere to the following guidelines when configuring network interfaces in the IP Interfaces table:

- 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 Routes table.
- The interface name **must** be configured (mandatory) and must be unique for each interface.
- Each network interface must be assigned an Ethernet Device.
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 IP Interfaces 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.

### 14.5.3 Networking Configuration Examples

This section provides configuration examples of networking interfaces.

#### 14.5.3.1 One VoIP Interface for All Applications

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

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

   Table 14-6: Example of Single VoIP Interface in IP Interfaces table

<table>
<thead>
<tr>
<th>Index</th>
<th>Application Type</th>
<th>Interface Mode</th>
<th>IP Address</th>
<th>Prefix Length</th>
<th>Default Gateway</th>
<th>Ethernet Device</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OAMP, Media &amp; Control</td>
<td>IPv4</td>
<td>192.168.0.2</td>
<td>16</td>
<td>192.168.0.1</td>
<td>1</td>
<td>myInterface</td>
</tr>
</tbody>
</table>

2. **Static Routes 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 14-7: Example of Static Routes Table

<table>
<thead>
<tr>
<th>Destination</th>
<th>Prefix Length</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>201.201.0.0</td>
<td>16</td>
<td>192.168.11.10</td>
</tr>
<tr>
<td>202.202.0.0</td>
<td>16</td>
<td>192.168.11.1</td>
</tr>
</tbody>
</table>

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

#### 14.5.3.2 VoIP Interface per Application Type

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

1. **IP Interfaces 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 14-8: Example of VoIP Interfaces per Application Type in IP Interfaces table

<table>
<thead>
<tr>
<th>Index</th>
<th>Application Type</th>
<th>Interface Mode</th>
<th>IP Address</th>
<th>Prefix Length</th>
<th>Default Gateway</th>
<th>Ethernet Device</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OAMP</td>
<td>IPv4</td>
<td>192.168.0.2</td>
<td>16</td>
<td>192.168.0.1</td>
<td>1</td>
<td>ManagementIF</td>
</tr>
</tbody>
</table>
2. **Static Routes table:** A routing rule is required to allow remote management from a host in 176.85.49.0 / 24:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Prefix Length</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>176.85.49.0</td>
<td>24</td>
<td>192.168.11.1</td>
</tr>
</tbody>
</table>

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

### 14.5.3.3 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. **IP Interfaces table:**

<table>
<thead>
<tr>
<th>Index</th>
<th>Application Type</th>
<th>Interface Mode</th>
<th>IP Address</th>
<th>Prefix Length</th>
<th>Default Gateway</th>
<th>Ethernet Device</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OAMP</td>
<td>IPv4 Manual</td>
<td>192.168.0.2</td>
<td>16</td>
<td>192.168.0.1</td>
<td>1</td>
<td>Mgmt</td>
</tr>
<tr>
<td>1</td>
<td>Media &amp; Control</td>
<td>IPv4 Manual</td>
<td>200.200.85.14</td>
<td>24</td>
<td>200.200.85.1</td>
<td>201</td>
<td>MediaCntrl1</td>
</tr>
<tr>
<td>2</td>
<td>Media &amp; Control</td>
<td>IPv4 Manual</td>
<td>200.200.86.14</td>
<td>24</td>
<td>200.200.86.1</td>
<td>202</td>
<td>MediaCntrl2</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Destination</th>
<th>Prefix Length</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td>176.85.49.0</td>
<td>24</td>
<td>192.168.0.10</td>
</tr>
</tbody>
</table>
3. The NTP application is configured (through the ini file) to serve as OAMP applications:

   EnableNTPasOAM = 1

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

14.5.3.4 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 14-12: Configured Default Gateway Example

<table>
<thead>
<tr>
<th>Index</th>
<th>Application Type</th>
<th>Interface Mode</th>
<th>IP Address</th>
<th>Prefix Length</th>
<th>Default Gateway</th>
<th>Ethernet Device</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>OAMP</td>
<td>IPv4 Manual</td>
<td>192.168.0.2</td>
<td>16</td>
<td>192.168.0.1</td>
<td>100</td>
<td>Mgmt</td>
</tr>
</tbody>
</table>

A separate Static Routes 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 14-13: Separate Static Routes Table Example

<table>
<thead>
<tr>
<th>Destination</th>
<th>Prefix Length</th>
<th>Gateway</th>
<th>Underlying Device</th>
</tr>
</thead>
<tbody>
<tr>
<td>17.17.0.0</td>
<td>16</td>
<td>192.168.10.1</td>
<td>100</td>
</tr>
<tr>
<td>171.79.39.0</td>
<td>24</td>
<td>200.200.85.10</td>
<td>200</td>
</tr>
</tbody>
</table>

14.6 Configuring Static IP Routes

The Static Routes table lets you configure up to 30 static IP routing rules. Static routes let you communicate with LAN networks that are not located behind the Default Gateway that is specified for an IP network interface in the IP Interfaces table, from which the packets are sent. Before sending an IP packet, the device searches the Static Routes table for an entry that matches the requested destination host/network. If an entry is found, the device sends the packet to the gateway that is configured for the static route. If no explicit entry is
found, the packet is sent to the Default Gateway as configured for the IP interface in the IP Interfaces table.

You can view the status of configured static routes in the IP Routing Status table. This table can be accessed by clicking the **Static Routes Status Table** link located at the bottom of the Static Routes table (see "Viewing Static Routes Status" on page 756).

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

➢ **To configure static IP routes:**
1. Open the Static Routes table (Setup menu > IP Network tab > Core Entities folder > Static Routes).
2. Click **New**; the following dialog box appears:

![Static Routes Table](image)

3. Configure a static route according to the parameters described in the table below. 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 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 are ignored. To reach a specific host, enter its IP address in the 'Destination' field and "32" in the 'Prefix Length' field.
4. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

⚠️ **Note:** Only static routing rules that are inactive can be deleted.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[StaticRouteTable_Index]</td>
<td>The valid value is 0 to 29.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description [StaticRouteTable_Description]</td>
<td>Defines a name for the rule. The valid value is a string of up to 20 characters.</td>
</tr>
<tr>
<td>Destination [StaticRouteTable_Destination]</td>
<td>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.</td>
</tr>
<tr>
<td>Prefix Length [StaticRouteTable_PrefixLength]</td>
<td>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.</td>
</tr>
<tr>
<td>Ethernet Output Name [StaticRouteTable_DeviceName]</td>
<td>Associates an IP network interface through which the static route's Gateway is reached. The association is done by assigning the parameter the same Ethernet Device that is assigned to the IP network interface in the IP Interfaces table ('Ethernet Device' parameter). To configure IP network interface, see Configuring IP Network Interfaces on page 137. To configure Ethernet Devices, see Configuring Underlying Ethernet Devices on page 134.</td>
</tr>
</tbody>
</table>
| 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. **Note:**  
- 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). |

### 14.6.1 Configuration Example of Static IP Routes

An example of the use for static routes is shown in the figure below. In the example, 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 Routes table must be associated with the IP network interface in the IP Interfaces table. This is done by configuring the 'Ethernet Output Name' field in the Static Routes table to the same value as configured in the 'Ethernet Device' field in the IP Interfaces table.
The static route's Gateway address in the Static Routes table is in the same subnet as the IP address of the IP network interface in the IP Interfaces table.

**Figure 14-4: Example of using a Static Route**

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

### 14.6.2 Troubleshooting the Static Routes Table

When adding a new static route to the Static Routes 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 Routes 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 has been configured in two different static routing rules.
- 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.

14.7 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.

14.7.1 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 device provides the following solutions, listed in priority of the method used by the device:

- If configured, uses the NAT Translation table which configures NAT per IP network interface - see Configuring NAT Translation per IP Interface on page 149.
- 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 IP Interfaces table.
14. Network

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

**Figure 14-5: Device behind NAT and NAT Issues**

![Diagram of NAT problem](image)

14.7.1.1 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 - public*) when the device is located behind NAT. The device's NAT traversal mechanism replaces the source IP address of SIP messages sent from a specified VoIP interface to a public IP address. This allows, for example, the separation of VoIP traffic between different ITSPs and topology hiding of internal IP addresses from the "public" network. Each IP network interface, configured in the IP Interfaces table can be associated with a NAT rule, translating the source IP address and port of the outgoing packet into the NAT address (IP address and port range).

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

➢ **To configure NAT translation rules:**

1. Open the NAT Translation table (Setup menu > IP Network tab > Core Entities folder > NAT Translation).
2. Click **New**; the following dialog box appears:

![Figure 14-6: NAT Translation Table - Dialog Box](image)

3. Configure a NAT translation rule according to the parameters described in the table below.

4. Click **Apply**, and then save your settings to flash memory.

<table>
<thead>
<tr>
<th><strong>Table 14-15: NAT Translation Table Parameter Descriptions</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td><strong>Source</strong></td>
</tr>
<tr>
<td>Index</td>
</tr>
<tr>
<td>Index</td>
</tr>
<tr>
<td>[NATTranslation_Index]</td>
</tr>
<tr>
<td>Source Interface</td>
</tr>
<tr>
<td>src-interface-name</td>
</tr>
<tr>
<td>[NATTranslation_SrcIPInterfaceName]</td>
</tr>
<tr>
<td>Source Start Port</td>
</tr>
<tr>
<td>src-start-port</td>
</tr>
<tr>
<td>[NATTranslation_SourceStartPort]</td>
</tr>
<tr>
<td>Source End Port</td>
</tr>
<tr>
<td>src-end-port</td>
</tr>
<tr>
<td>[NATTranslation_SourceEndPort]</td>
</tr>
<tr>
<td><strong>Target</strong></td>
</tr>
<tr>
<td>Target IP Address</td>
</tr>
<tr>
<td>target-ip-address</td>
</tr>
<tr>
<td>[NATTranslation_TargetIPAddress]</td>
</tr>
<tr>
<td>Target Start Port</td>
</tr>
<tr>
<td>target-start-port</td>
</tr>
<tr>
<td>[NATTranslation_TargetStartPort]</td>
</tr>
<tr>
<td>Parameter</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>Target End Port</td>
</tr>
<tr>
<td>target-end-port</td>
</tr>
</tbody>
</table>

14.7.2  Remote UA behind NAT

This section describes configuration for scenarios where the device sends signaling and media packets to a remote UA that is located behind NAT.

14.7.2.1 SIP Signaling Messages

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

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 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 Groups table (see "Configuring IP Groups" on page 345). If this feature is disabled, the device’s NAT detection is according to the settings of the global parameter, 'SIP NAT Detection' parameter (see below procedure).

➢ To enable the NAT Detection feature (global):

1. Open the Transport Settings page (Setup menu > Signaling & Media tab > SIP Definitions folder > Transport Settings).
2. From the 'SIP NAT Detection' drop-down list (SIPnatDetection), select Enable:

   ![Figure 14-7: Enabling SIP NAT Detection](image)

3. Click Apply.

14.7.2.2 Media (RTP/RTCP/T.38)

When a remote UA initiates a call and is not located behind a NAT server, the device sends the media (RTP, RTCP, and T.38) packets to the remote UA using the IP
address:port (UDP) indicated in the SDP body of the SIP message received from the UA. However, if the UA is located behind NAT, the device sends the RTP with the IP address of the UA (i.e., private IP address) as the destination instead of that of the NAT server. Thus, the RTP will not reach the UA. To resolve this NAT traversal problem, the device offers the following features:

- First Incoming Packet Mechanism - see "First Incoming Packet Mechanism" on page 152
- RTP No-Op packets according to the avt-rtp-noop draft - see "No-Op Packets" on page 153

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

**Figure 14-8: Remote UA behind NAT**

![Diagram of network architecture](image)

### 14.7.2.2.1 First Incoming Packet Mechanism

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

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

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

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

Note: If the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA does it determine whether the UA is behind NAT.

- [1] Disable NAT: (Default) The device considers the UA as not located behind NAT.
and sends media packets to the UA using the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message.

- [2] Force NAT: The device always considers the UA as behind NAT and sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA. The device only sends packets to the UA after it receives the first packet from the UA (to obtain the IP address).

- [3] NAT by Signaling = The device identifies whether or not the UA is located behind NAT based on the SIP signaling. The device assumes that if signaling is behind NAT that the media is also behind NAT, and vice versa. If located behind NAT, the device sends media as described in option [2] Force NAT; if not behind NAT, the device sends media as described in option [1] Disable NAT. This option is applicable only to SBC calls. If the parameter is configured to this option, Gateway calls use option [0] Enable NAT Option, by default.

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

1. Open the Media Settings page (Setup menu > Signaling & Media tab > Media folder > Media Settings), and then from the 'NAT Traversal' drop-down list (NATMode), select the required NAT option:

   ![Figure 14-9: Configuring NAT Traversal for Media](image)

2. Click Apply.

14.7.2.2 No-Op Packets

The device can send No-Op packets to verify Real-Time Transport Protocol (RTP) and T.38 connectivity, and to keep NAT bindings and Firewall pinholes open. The No-Op packets can be sent in RTP and T.38 formats:

- **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"). The IETF document defines a No-Op payload format for RTP. The draft defines the RTP payload type as dynamic. You can configure the payload type as described in the following procedure (default 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).

➢ To configure the No-Op packet feature:

1. Enable the feature, using the NoOpEnable ini file parameter.

2. Configure the time interval during which the device sends No-Op packets when silence occurs (i.e., no RTP or T.38 traffic), using the NoOpInterval ini file parameter.

3. For RTP No-Op packets, configure the payload type of the No-Op packets, using the RTPNoOpPayloadType ini file parameter.

**Note:**

- The No-OP Packet feature requires DSP resources.
- Receipt of No-Op packets is always supported.
14.7.2.2.3 Fax Transmission behind NAT

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

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

14.7.2.2.4 ICE Lite

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

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

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

These addresses:ports (local, STUN, TURN and any other network address) of the client are termed "candidates". Each client sends its' candidates to the other in the SDP body of the INVITE message. Peers then perform connectivity checks per candidate of the other peer, using STUN binding requests sent on the RTP and RTCP ports. ICE tries each candidate and selects the one that works (i.e., media can flow between the clients). The following figure shows a simple illustration of ICE:
connectivity checks of ICE candidates and responds to them with STUN responses. Note that in the response to the INVITE message received from the remote endpoint, the device sends only a single candidate for its own IP address. This is the IP address of the device that the client uses. To support ICE, the SBC leg interfacing with the ICE-enabled client (SIP entity) must be enabled for ICE. This is done using the IP Profile parameter, IPProfile_SBCIceMode (see "Configuring IP Profiles" on page 417).

As the ICE technique has been defined by the WebRTC standard as mandatory for communication with the WebRTC client, ICE support by the device is important for deployments implementing WebRTC. For more information on WebRTC, see "WebRTC" on page 566. Once a WebRTC session (WebSocket) is established for SIP signaling between the device and the WebRTC client, the client's IP address needs to be discovered by the SBC device using the ICE technique.

14.8 Configuring Quality of Service

This section describes how to configure Layer-2 and Layer-3 Quality of Service (QoS).

14.8.1 Configuring Class-of-Service QoS

The QoS Settings page lets you configure Layer-3 Class-of-Service Quality of Service (QoS). This configures Differentiated Services (DiffServ) values for each CoS. 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):

- 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 mapping of an application to its CoS and traffic type is shown in the table below:
### Table 14-16: Traffic/Network Types and Priority

<table>
<thead>
<tr>
<th>Application</th>
<th>Traffic / Network Types</th>
<th>Class-of-Service (Priority)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debugging interface</td>
<td>Management</td>
<td>Bronze</td>
</tr>
<tr>
<td>Telnet</td>
<td>Management</td>
<td>Bronze</td>
</tr>
<tr>
<td>Web server (HTTP)</td>
<td>Management</td>
<td>Bronze</td>
</tr>
<tr>
<td>SNMP GET/SET</td>
<td>Management</td>
<td>Bronze</td>
</tr>
<tr>
<td>Web server (HTTPS)</td>
<td>Management</td>
<td>Bronze</td>
</tr>
<tr>
<td>RTP traffic</td>
<td>Media</td>
<td>Media Premium</td>
</tr>
<tr>
<td>RTCP traffic</td>
<td>Media</td>
<td>Media Premium</td>
</tr>
<tr>
<td>T.38 traffic</td>
<td>Media</td>
<td>Media Premium</td>
</tr>
<tr>
<td>SIP</td>
<td>Control</td>
<td>Control Premium</td>
</tr>
<tr>
<td>SIP over TLS (SIPS)</td>
<td>Control</td>
<td>Control Premium</td>
</tr>
<tr>
<td>Syslog</td>
<td>Management</td>
<td>Bronze</td>
</tr>
<tr>
<td>SNMP Traps</td>
<td>Management</td>
<td>Bronze</td>
</tr>
<tr>
<td>DNS client</td>
<td>Varies according to DNS settings:</td>
<td>Depends on traffic type:</td>
</tr>
<tr>
<td></td>
<td>▪ OAMP</td>
<td>▪ Control: Control Premium</td>
</tr>
<tr>
<td></td>
<td>▪ Control</td>
<td>▪ Management: Bronze</td>
</tr>
<tr>
<td>NTP</td>
<td>Varies according to the interface type associated with NTP (see &quot;Assigning NTP Services to Application Types&quot; on page 141):</td>
<td>Depends on traffic type:</td>
</tr>
<tr>
<td></td>
<td>▪ OAMP</td>
<td>▪ Control: Control Premium</td>
</tr>
<tr>
<td></td>
<td>▪ Control</td>
<td>▪ Management: Bronze</td>
</tr>
</tbody>
</table>

➢ **To configure DiffServ (Layer-3 QoS) values per CoS:**

1. Open the QoS Settings page (**Setup** menu > **IP Network** tab > **Quality** folder > **QoS Settings**).
2. Click **New**; the following dialog box appears:

   ![Figure 14-10: Configuring Class of Service](image)

3. Configure DiffServ values per CoS according to the parameters described in the table below.
4. Click **Apply**, and then save your settings to flash memory.
### Table 14-17: QoS Settings Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media Premium QoS media-qos</td>
<td>Defines the DiffServ value for Premium Media CoS content. The valid range is 0 to 63. The default is 46. <strong>Note:</strong> You can also configure the parameter per IP Profile (IpProfile_IPDiffServ).</td>
</tr>
<tr>
<td>Control Premium QoS control-qos</td>
<td>Defines the DiffServ value for Premium Control CoS content (Call Control applications). The valid range is 0 to 63. The default is 40. <strong>Note:</strong> You can also configure the parameter per IP Profile (IpProfile_SigIPDiffServ).</td>
</tr>
<tr>
<td>Gold QoS gold-qos</td>
<td>Defines the DiffServ value for Gold CoS content (streaming applications). The valid range is 0 to 63. The default is 26.</td>
</tr>
<tr>
<td>Bronze QoS bronze-qos</td>
<td>Defines the DiffServ value for Bronze CoS content (OAMP applications). The valid range is 0 to 63. The default is 10.</td>
</tr>
</tbody>
</table>

### 14.8.2 Configuring DiffServ-to-VLAN Priority Mapping

The QoS Mapping table lets you configure up to 64 DiffServ-to-VLAN priority mapping for Layer 3 and Layer-2 Quality of Service (QoS). 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. Layer-2 802.1Q frames have a 2-byte field called Tag Control Information. The three most significant bits of this 2-byte field represents the Class of Service (CoS) value. Layer-2 QoS is represented by this CoS value which is from 0 to 7 (thus 8 values). 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). 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.

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

- **To configure DiffServ-to-VLAN priority mapping:**
  1. Open the QoS Mapping table (Setup menu > IP Network tab > Quality folder > QoS Mapping).
2. Click **New**; the following dialog box appears:

   **Figure 14-11: QoS Mapping Table - Add Dialog Box**

3. Configure a DiffServ-to-VLAN priority mapping rule according to the parameters described in the table below.

4. Click **Apply**, and then save your settings to flash memory.

**Table 14-18: QoS Mapping Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td><strong>Differentiated Services</strong></td>
<td>Defines a DiffServ value.</td>
</tr>
<tr>
<td><strong>diff-serv</strong></td>
<td>The valid value is 0 to 63. The default is 0.</td>
</tr>
<tr>
<td><strong>VLAN Priority</strong></td>
<td>Defines the VLAN priority level.</td>
</tr>
<tr>
<td><strong>vlan-priority</strong></td>
<td>The valid value is 0 to 7. The default is 0.</td>
</tr>
</tbody>
</table>

### 14.9 Configuring ICMP Messages

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

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

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

This feature is applicable to IPv4 and IPv6 addressing schemes.

The following procedure describes how to configure ICMP messaging through the Web interface. You can also configure it through ini file - DisableICMPUnreachable (ICMP Unreachable messages) and DisableICMPRedirects (ICMP Redirect messages).
To configure handling of ICMP messages:

1. Open the Network Settings page (Setup menu > IP Network tab > Advanced folder > Network Settings).

   ![Figure 14-12: Configuring ICMP Messaging]

2. Under the ICMP group, do the following:
   - To enable sending and receipt of ICMP Redirect messages, configure the 'Send and Receive ICMP Redirect Messages' parameter to Enable.
   - To enable sending of ICMP Destination Unreachable messages, configure the 'Send ICMP Unreachable Messages' parameter to Enable.

3. Click Apply.

14.10 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 159
- Internal SRV table - see "Configuring the Internal SRV Table" on page 160

14.10.1 Configuring the Internal DNS Table

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

**Note:** The device first attempts to resolve a domain name using the 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 137).

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

To configure the device’s DNS table:

1. Open the Internal DNS table (Setup menu > IP Network tab > DNS folder > Internal DNS).
2. Click **New**; the following dialog box appears:

![Figure 14-13: Internal DNS Table - Add Dialog Box](image)

3. Configure a DNS rule according to the parameters described in the table below.

4. Click **Apply**.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td><strong>Domain Name</strong></td>
<td>Defines the host name to be translated. The valid value is a string of up to 31 characters.</td>
</tr>
<tr>
<td><strong>First IP Address</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Second IP Address</strong></td>
<td>Defines the second IP address (in dotted-decimal format notation) to which the host name is translated.</td>
</tr>
<tr>
<td><strong>Third IP Address</strong></td>
<td>Defines the third IP address (in dotted-decimal format notation) to which the host name is translated.</td>
</tr>
</tbody>
</table>

### 14.10.2 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.
The following procedure describes how to configure the Internal SRV table through the Web interface. You can also configure it through ini file (SRV2IP) or CLI (configure network > dns srv2ip).

To configure the device’s SRV table:
1. Open the Internal SRV table (Setup menu > IP Network tab > DNS folder > Internal SRV).
2. Click New; the following dialog box appears:

![Internal SRV Table - Add Dialog Box](image)

3. Configure an SRV rule according to the parameters described in the table below.
4. Click Apply, and then save your settings to flash memory.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Domain Name</td>
<td>Defines the host name to be translated.</td>
</tr>
<tr>
<td>domain-name [Srv2ip_InternalDomain]</td>
<td>The valid value is a string of up to 31 characters. By default, no value is defined.</td>
</tr>
<tr>
<td>Transport Type</td>
<td>Defines the transport type.</td>
</tr>
<tr>
<td>transport-type [Srv2ip_TransportType]</td>
<td>- [0] UDP (default)</td>
</tr>
<tr>
<td></td>
<td>- [1] TCP</td>
</tr>
<tr>
<td></td>
<td>- [2] TLS</td>
</tr>
</tbody>
</table>
### 14.11 Robust Receipt of Media Streams by Media Latching

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

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

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

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

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

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

> **To configure media latching:**

1. Open the Media Settings page (Setup menu > Signaling & Media tab > Media folder > Media Settings), and then from the 'Inbound Media Latch Mode' drop-down list...
(InboundMediaLatchMode), configure the media latch mode:

![Inbound Media Latch Mode](Dynamic)

2. If you configure the parameter to Dynamic or Dynamic-Strict:
   - 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'

![ROBUSTNESS Table](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)

3. Click Apply, and then save your settings to flash memory.

### 14.12 Multiple Routers Support

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

When a set of routers operating within the same subnet serve as devices to that network and intercommunicate using a dynamic routing protocol, the routers can determine the shortest path to a certain destination and signal the remote host the existence of the better route. Using multiple router support, the device can utilize these router messages to change its next hop and establish the best path.
Note: Multiple Routers support is an integral feature that doesn’t require configuration.
15 Security

This section describes the VoIP security-related configuration.

15.1 Configuring Firewall Settings

The Firewall table lets you configure up to 500 firewall rules, which define network traffic filtering rules (access list) for incoming traffic. 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 device searches the table from top to bottom until the first matching rule is found. The matched rule can 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.
**Note:**

- The rules configured by the Firewall table apply to a very low-level network layer and overrides all 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 device's Web and Telnet management interfaces in the Access List table (see "Configuring Web and Telnet Access List" on page 81), you must configure a firewall rule that permits traffic from these IP addresses.

- Only users with Security Administrator or Master access levels can configure firewall rules.

- The device supports dynamic firewall pinholes for media (RTP/RTCP) traffic negotiated in the SDP offer-answer of SIP calls. The pinhole allows the device to ignore its firewall and accept the traffic on the negotiated port. The device automatically closes the pinhole once the call terminates. Therefore, it is unnecessary to configure specific firewall rules to allow traffic through specific ports. For example, if you have configured a firewall rule to block all media traffic in the port range 6000 to 7000 and a call is negotiated to use the local port 6010, the device automatically opens port 6010 to allow the call.

- Setting the 'Prefix Length' field to 0 means that the rule applies to all packets, regardless of the defined IP address in the 'Source IP' field. Thus, it is highly recommended to set the parameter to a value other than 0.

- It is recommended to add a rule at the end of your table that blocks all traffic and to add firewall rules above it that allow required traffic (with bandwidth limitations). To block all traffic, use the following firewall rule:
  - Source IP: 0.0.0.0
  - Prefix Length: 0 (i.e., rule matches all IP addresses)
  - Start Port - End Port: 0-65535
  - Protocol: Any
  - Action Upon Match: Block

- If you are using the High Availability feature and you have configured "block" rules, ensure that you also add "allow" rules for HA traffic. For more information, see Configuring Firewall Allowed Rules on page 610.

The following procedure describes how to configure firewall rules through the Web interface. You can also configure it through ini file (AccessList) or CLI (configure network > access-list).

- **To configure a firewall rule:**
  1. Open the Firewall table (Setup menu > IP Network tab > Security folder> Firewall).
2. Click **New**; the following dialog box appears:

**Figure 15-1: Firewall Table - Add Dialog Box**

3. Configure a firewall rule according to the parameters described in the table below.

4. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

**Table 15-1: Firewall Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Match</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td><strong>Source IP</strong></td>
<td></td>
</tr>
<tr>
<td>source-ip</td>
<td>Defines the IP address (or DNS name) or a specific host name of the source network from where the device receives the incoming packet.</td>
</tr>
<tr>
<td>[AccessList_Source_IP]</td>
<td>The default is 0.0.0.0.</td>
</tr>
<tr>
<td><strong>Source Port</strong></td>
<td></td>
</tr>
<tr>
<td>src-port</td>
<td>Defines the source UDP/TCP ports of the remote host from where the device receives the incoming packet.</td>
</tr>
<tr>
<td>[AccessList_Source_Port]</td>
<td>The valid range is 0 to 65535. The default is 0.</td>
</tr>
<tr>
<td><strong>Prefix Length</strong></td>
<td></td>
</tr>
<tr>
<td>prefixLen</td>
<td>(Mandatory) Defines the IP network mask - 32 for a single host or the appropriate value for the source IP addresses.</td>
</tr>
<tr>
<td>[AccessList_PrefixLen]</td>
<td>- A value of 8 corresponds to IPv4 subnet class A (network mask of 255.0.0.0).</td>
</tr>
<tr>
<td></td>
<td>- A value of 16 corresponds to IPv4 subnet class B (network mask of 255.255.0.0).</td>
</tr>
<tr>
<td></td>
<td>- A value of 24 corresponds to IPv4 subnet class C (network mask of 255.255.255.0).</td>
</tr>
<tr>
<td></td>
<td>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’.</td>
</tr>
<tr>
<td></td>
<td>The default is 0 (i.e., applies to all packets). You <strong>must</strong> change this value to any of the above options.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>A value of 0 applies to all packets, regardless of the defined IP address. Therefore, you must set the parameter to a value other than</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>0.</td>
<td></td>
</tr>
<tr>
<td>Start Port</td>
<td>Defines the first UDP/TCP port in the range of ports on the device on which the incoming packet is received. From the perspective of the remote IP entity, this is the destination port. To configure the last port in the range, see the 'End Port' parameter (below). The valid range is 0 to 65535. <strong>Note:</strong> When the protocol type isn't TCP or UDP, the entire range must be provided.</td>
</tr>
<tr>
<td>End Port</td>
<td>Defines the last UDP/TCP port in the range of ports on the device on which the incoming packet is received. From the perspective of the remote IP entity, this is the destination port. To configure the first port in the range, see the 'Start Port' parameter (above). The valid range is 0 to 65535 (default). <strong>Note:</strong> When the protocol type isn't TCP or UDP, the entire range must be provided.</td>
</tr>
<tr>
<td>Protocol</td>
<td>Defines the protocol type (e.g., <strong>UDP</strong>, <strong>TCP</strong>, <strong>ICMP</strong>, <strong>ESP</strong> or <strong>Any</strong>) or the IANA protocol number in the range of 0 (Any) to 255. The default is <strong>Any</strong>. <strong>Note:</strong> The parameter also accepts the abbreviated strings &quot;SIP&quot; and &quot;HTTP&quot;. Specifying these strings implies selection of the TCP or UDP protocols and the appropriate port numbers as defined on the device.</td>
</tr>
</tbody>
</table>
| Use Specific Interface          | Determines whether you want to apply the rule to a specific network interface defined in the IP Interfaces table (i.e., packets received from that defined in the Source IP field and received on this network interface):  
  - [0] Disable (default)  
  - [1] Enable  
  **Note:**  
  - 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                  | 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 IP Interfaces table in "Configuring IP Network Interfaces" on page 137. |
| Action                          | Defines the firewall action to be performed upon rule match.  
  - "Allow" = (Default) Permits the packets.  
  - "Block" = Rejects the packets |
| Packet Size                     | Defines the maximum allowed packet size. The valid range is 0 to 65535. **Note:** When filtering fragmented IP packets, this field relates to the overall (re-assembled) packet size, and not to the size of each fragment. |
### 15. Security

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Byte Rate</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Burst Bytes</strong></td>
<td>Defines the tolerance of traffic rate limit (number of bytes). The default is 0.</td>
</tr>
<tr>
<td><strong>Statistics</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Match Count</strong></td>
<td>(Read-only) Displays the number of packets accepted or rejected by the rule.</td>
</tr>
</tbody>
</table>

The table below provides an example of configured firewall rules:

**Table 15-2: Configuration Example of Firewall Rules**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Firewall Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Source IP</strong></td>
<td>12.194.231.76 12.194.230.7 0.0.0.0 192.0.0.0 0.0.0.0</td>
</tr>
<tr>
<td><strong>Prefix Length</strong></td>
<td>16 16 0 8 0</td>
</tr>
<tr>
<td><strong>Start Port and End Port</strong></td>
<td>0-65535 0-65535 0-65535 0-65535 0-65535</td>
</tr>
<tr>
<td><strong>Protocol</strong></td>
<td>Any Any icmp Any Any</td>
</tr>
<tr>
<td><strong>Use Specific Interface</strong></td>
<td>Enable Enable Disable Enable Disable</td>
</tr>
<tr>
<td><strong>Interface Name</strong></td>
<td>WAN WAN None Voice-Lan None</td>
</tr>
<tr>
<td><strong>Byte Rate</strong></td>
<td>0 0 40000 40000 0</td>
</tr>
<tr>
<td><strong>Burst Bytes</strong></td>
<td>0 0 50000 50000 0</td>
</tr>
<tr>
<td><strong>Action Upon Match</strong></td>
<td>Allow Allow Allow Block</td>
</tr>
</tbody>
</table>

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.

15.2 Configuring TLS for SIP

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

![Note: When a TLS connection with the device is initiated by a SIP client, the device also responds using TLS, regardless of whether or not TLS was configured.]

➢ To configure SIPS:

1. Configure a TLS Context as required (see "Configuring TLS Certificate Contexts" on page 105).
2. Assign the TLS Context to a Proxy Set or SIP Interface (see "Configuring Proxy Sets" on page 359 and "Configuring SIP Interfaces" on page 337, respectively).
3. Configure a SIP Interface with a TLS port number.

![Figure 15-2: Configuring TLS]

For a description of the TLS parameters, see "TLS Parameters" on page 871.

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), configure the 'Enable SIPS' (EnableSIPS) parameter to Enable on the Transport Settings page (Setup menu > Signaling & Media tab > SIP Definitions folder > Transport Settings):

![Figure 15-3: Enabling SIPS]
15.3 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 178.

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).
15.3.1 Enabling IDS

The following procedure describes how to enable IDS.

➢ To enable IDS:

1. Open the IDS General Settings page (Setup menu > Signaling & Media tab > Intrusion Detection folder > IDS General Settings).

   ![Figure 15-4: Enabling IDS]

   

2. From the 'Intrusion Detection System' drop-down list, select Enable.

3. Click Apply, and then reset the device with a save-to-flash for the setting to take effect.

15.3.2 Configuring IDS Policies

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

1. IDS Policies (parent table): Defines a name and provides a 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

Note: The default IDS Policies are read-only and cannot be modified.

The following procedure describes how to configure IDS Policies through the Web interface. You can also configure it through ini file or CLI:

- IDS Policy table: IDSPolicy (ini file) or configure voip > ids policy (CLI)
- IDS Rules table: IDSRule (ini file) or configure voip > ids rule (CLI)
To configure an IDS Policy:

1. Open the IDS Policies table (Setup menu > Signaling & Media tab > Intrusion Detection folder > IDS Policies); the table displays the pre-configured IDS policies.

   Figure 15-5: IDS Policies Table with Default Rules

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>DOS</td>
<td>dox-attacks</td>
</tr>
<tr>
<td>1</td>
<td>DEFAULT_FEU</td>
<td>Default policy for FEU</td>
</tr>
<tr>
<td>2</td>
<td>DEFAULT_PROXY</td>
<td>Default policy for proxies</td>
</tr>
<tr>
<td>3</td>
<td>DEFAULT_GLOBAL</td>
<td>Default policy for global scope</td>
</tr>
</tbody>
</table>

2. Click New; the following dialog box appears:

   Figure 15-6: IDS Policies Table - Add Dialog Box

3. Configure an IDS Policy name according to the parameters described in the table below.

4. Click Apply.

   Table 15-3: IDS Policies Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index policy [IDSPolicy_Index]</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>Name rule [IDSPolicy_Name]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td>Description description [IDSPolicy_Description]</td>
<td>Defines a brief description for the IDS Policy. The valid value is a string of up to 100 characters.</td>
</tr>
</tbody>
</table>

5. In the IDS Policies table, select the required IDS Policy row, and then click the IDS Rule link located below the table; the IDS Rule table opens.
6. Click **New**; the following dialog box appears:

   **Figure 15-7: IDS Rule Table - Add Dialog Box**

   ![IDS Rule Table - Add Dialog Box](image)

   The figure above shows a configuration example: If 15 malformed SIP messages ('Reason') are received within a period of 30 seconds ('Threshold Window'), a minor alarm is sent ('Minor-Alarm Threshold'). Every 30 seconds, the rule's counters are cleared ('Threshold Window'). If more than 25 malformed SIP messages are received within this period, the device blacklists for 60 seconds the remote IP host ('Deny Threshold') from where the messages were received.

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

8. Click **Apply**, and then save your settings to flash memory.

9. For example

   **Table 15-4: IDS Rule Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table record.</td>
</tr>
<tr>
<td>rule-id [IDSRule_RuleID]</td>
<td></td>
</tr>
<tr>
<td><strong>Reason</strong></td>
<td>Defines the type of intrusion attack (malicious event).</td>
</tr>
<tr>
<td>reason [IDSRule_Reason]</td>
<td></td>
</tr>
<tr>
<td>[0] Any</td>
<td>All events listed below are considered as attacks and are counted together.</td>
</tr>
<tr>
<td>[2] Malformed message</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Message exceeds a user-defined maximum message length (50K)</td>
</tr>
<tr>
<td></td>
<td>Any SIP parser error</td>
</tr>
<tr>
<td></td>
<td>Message Policy match (see &quot;Configuring SIP Message Policy Rules&quot;)</td>
</tr>
<tr>
<td></td>
<td>Basic headers not present</td>
</tr>
<tr>
<td></td>
<td>Content length header not present (for TCP)</td>
</tr>
<tr>
<td></td>
<td>Header overflow</td>
</tr>
<tr>
<td></td>
<td>Local authentication (&quot;Bad digest&quot; errors)</td>
</tr>
<tr>
<td></td>
<td>Remote authentication (SIP 401/407 is sent if original authentication fails)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>message includes authentication)</td>
</tr>
<tr>
<td></td>
<td>• [4] Dialog establish failure =</td>
</tr>
<tr>
<td></td>
<td>✓ Classification failure (see &quot;Configuring Classification Rules&quot; on page 493). This also applies to calls rejected by the device based on a registered users policy (configured by the SRD_BlockUnRegUsers or SIPInterface_BlockUnRegUsersblocks parameters).</td>
</tr>
<tr>
<td></td>
<td>✓ Routing failure</td>
</tr>
<tr>
<td></td>
<td>✓ Other local rejects (prior to SIP 180 response)</td>
</tr>
<tr>
<td></td>
<td>✓ Remote rejects (prior to SIP 180 response)</td>
</tr>
<tr>
<td></td>
<td>✓ Malicious signature pattern detected (see &quot;Configuring Malicious Signatures&quot; on page 559)</td>
</tr>
<tr>
<td></td>
<td>• [5] Abnormal flow =</td>
</tr>
<tr>
<td></td>
<td>✓ Requests and responses without a matching transaction user (except ACK requests)</td>
</tr>
<tr>
<td></td>
<td>✓ Requests and responses without a matching transaction (except ACK requests)</td>
</tr>
<tr>
<td>Threshold Scope</td>
<td>Defines the source of the attacker to consider in the device's detection count.</td>
</tr>
<tr>
<td>threshold-scope</td>
<td>[IDSRule_ThresholdScope]</td>
</tr>
<tr>
<td></td>
<td>• [0] Global = All attacks regardless of source are counted together during the threshold window.</td>
</tr>
<tr>
<td></td>
<td>• [2] IP = Attacks from each specific IP address are counted separately during the threshold window.</td>
</tr>
<tr>
<td></td>
<td>• [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.</td>
</tr>
<tr>
<td>Threshold Window</td>
<td>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.</td>
</tr>
<tr>
<td>threshold-window</td>
<td>[IDSRule_ThresholdWindow]</td>
</tr>
<tr>
<td></td>
<td>The valid range is 1 to 1,000,000. The default is 1.</td>
</tr>
<tr>
<td>Alarms</td>
<td></td>
</tr>
<tr>
<td>Minor-Alarm Threshold</td>
<td>Defines the threshold that if crossed a minor severity alarm is sent.</td>
</tr>
<tr>
<td>minor-alrm-thr</td>
<td>[IDSRule_MinorAlarmThreshold]</td>
</tr>
<tr>
<td></td>
<td>The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.</td>
</tr>
<tr>
<td>Major-Alarm Threshold</td>
<td>Defines the threshold that if crossed a major severity alarm is sent.</td>
</tr>
<tr>
<td>major-alrm-thr</td>
<td>[IDSRule_MajorAlarmThreshold]</td>
</tr>
<tr>
<td></td>
<td>The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.</td>
</tr>
<tr>
<td>Critical-Alarm Threshold</td>
<td>Defines the threshold that if crossed a critical severity alarm is sent.</td>
</tr>
<tr>
<td>critical-alrm-thr</td>
<td>[IDSRule_CriticalAlarmThreshold]</td>
</tr>
<tr>
<td></td>
<td>The valid range is 1 to 1,000,000. A value of 0 or -1 means not defined.</td>
</tr>
<tr>
<td>Deny</td>
<td></td>
</tr>
<tr>
<td>Deny Threshold</td>
<td>Defines the threshold that if crossed, the device blocks (blacklists)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| deny-thr [IDSRule_DenyThreshold] | the remote host (attacker).  
The default is -1 (i.e., not configured).  
**Note:** The parameter is applicable only if the ‘Threshold Scope’ parameter is set to IP or IP+Port. |
| Deny Period | deny-period [IDSRule_DenyPeriod] | Defines the duration (in sec) to keep the attacker on the blacklist, if configured using the 'Deny Threshold' parameter.  
The valid range is 0 to 1,000,000. The default is -1 (i.e., not configured).  
**Note:** The parameter is applicable only if the ‘Threshold Scope’ parameter is set to IP or IP+Port. |

### 15.3.3 Assigning IDS Policies

The IDS Matches 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). To configure SIP Interfaces, see "Configuring SIP Interfaces" on page 337.
- **Proxy Sets**: For detection of malicious attacks from specified Proxy Set(s). To configure Proxy Sets, see "Configuring Proxy Sets" on page 359.
- **Subnet addresses**: For detection of malicious attacks from specified subnet addresses.

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

The following procedure describes how to configure the IDS Match table through the Web interface. You can also configure it through ini file (IDSMatch) or CLI (configure voip > ids match).

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

1. Open the IDS Matches table (Setup menu > Signaling & Media tab > Intrusion Detection folder > IDS Matches).
2. Click New; the following dialog box appears:

**Figure 15-8: IDS Matches Table - Add Dialog Box**

<table>
<thead>
<tr>
<th>Index</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP Interface ID</td>
<td>1-2</td>
</tr>
<tr>
<td>Proxy Set ID</td>
<td></td>
</tr>
<tr>
<td>Subnet</td>
<td>10.0.1.0/16 &amp; 10.0.2.22</td>
</tr>
<tr>
<td>Policy</td>
<td>#4 [SIP Trunk]</td>
</tr>
</tbody>
</table>
The figure above shows a configuration example where the IDS Policy "SIP Trunk" is applied to SIP Interfaces 1 and 2, and to 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 **Apply**, and then save your settings to flash memory.

<table>
<thead>
<tr>
<th>Table 15-5: IDS Matches Table Parameter Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>Index [IDSMatch_Index]</td>
</tr>
<tr>
<td>SIP Interface ID sip-interface [IDSMatch_SIPInterface]</td>
</tr>
<tr>
<td>Proxy Set ID proxy-set [IDSMatch_ProxySet]</td>
</tr>
<tr>
<td>Subnet subnet [IDSMatch_Subnet]</td>
</tr>
</tbody>
</table>

Note:

- 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.

- 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)

- 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)

- 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.
15.3.4 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 172). 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 735). In this example, a Minor threshold alarm is cleared and replaced by a Major threshold alarm:

![Figure 15-9: IDS Alarms in Active Alarms Table](image)

- 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 through CLI:

- To view all active IDS alarms:
  ```
  # show voip ids active-alarm all
  ```

- To view all IP addresses that have crossed the threshold for an active IDS alarm:
  ```
  # show voip 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 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 797). An example of a Syslog message with IDS alarms and notifications is shown below:

**Figure 15-10: Syslog Message Example with IDS Alarms and Notifications**

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

**Table 15-6: Types of Malicious Events and Syslog Text String**

<table>
<thead>
<tr>
<th>Reason</th>
<th>Description</th>
<th>Syslog String</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Connection Abuse</strong></td>
<td>TLS authentication failure</td>
<td>abuse-tls-auth-fail</td>
</tr>
<tr>
<td><strong>Malformed Messages</strong></td>
<td>▪ Message exceeds a user-defined maximum message length (50K)</td>
<td>malformed-invalid-msg-len</td>
</tr>
<tr>
<td></td>
<td>▪ Any SIP parser error</td>
<td>malformed-parse-error</td>
</tr>
<tr>
<td></td>
<td>▪ Message policy match</td>
<td>malformed-message-policy</td>
</tr>
<tr>
<td></td>
<td>▪ Basic headers not present</td>
<td>malformed-miss-header</td>
</tr>
<tr>
<td></td>
<td>▪ Content length header not present (for TCP)</td>
<td>malformed-miss-content-len</td>
</tr>
<tr>
<td></td>
<td>▪ Header overflow</td>
<td>malformed-header-overflow</td>
</tr>
<tr>
<td><strong>Authentication Failure</strong></td>
<td>Local authentication (&quot;Bad digest&quot; errors)</td>
<td>auth-establish-fail</td>
</tr>
<tr>
<td></td>
<td>Remote authentication (SIP 401/407 is sent if original message includes authentication)</td>
<td>auth-reject-response</td>
</tr>
<tr>
<td><strong>Dialog Establishment Failure</strong></td>
<td>Classification failure</td>
<td>establish-classify-fail</td>
</tr>
<tr>
<td></td>
<td>Routing failure</td>
<td>establish-route-fail</td>
</tr>
<tr>
<td></td>
<td>Other local rejects (prior to SIP 180 response)</td>
<td>establish-local-reject</td>
</tr>
<tr>
<td></td>
<td>Remote rejects (prior to SIP 180 response)</td>
<td>establish-remote-reject</td>
</tr>
<tr>
<td></td>
<td>Malicious signature pattern detected</td>
<td>establish-malicious-signature-db-reject</td>
</tr>
<tr>
<td><strong>Abnormal Flow</strong></td>
<td>Requests and responses without a matching transaction user (except ACK requests)</td>
<td>flow-no-match-tu</td>
</tr>
<tr>
<td></td>
<td>Requests and responses without a matching transaction (except ACK requests)</td>
<td>flow-no-match-transaction</td>
</tr>
</tbody>
</table>
This page is intentionally left blank.
16 Media

This section describes the media-related configuration.

16.1 Configuring Voice Settings

The section describes various voice-related configuration such as voice volume and DTMF transport type. For a detailed description of these parameters, see "Configuration Parameters Reference" on page 831.

16.1.1 Configuring Voice Gain (Volume) Control

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

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

— To configure gain control through the Web interface:
  1. Open the Voice Settings page (Setup menu > Signaling & Media tab > Media folder > Voice Settings).
  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 Apply.

16.1.2 Configuring Echo Cancellation

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

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

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

The device also supports acoustic echo cancellation for SBC calls. These echoes are composed of undesirable acoustical reflections (non-linear) of the received signal (i.e., from the speaker) which find their way from multiple reflections such as walls and windows into the transmitted signal (i.e., microphone). Therefore, the party at the far end hears his / her echo. The device removes these echoes and sends only the near-end’s desired speech signal to the network (i.e., to the far-end party). The echo is composed of a linear part and a nonlinear part. However, in the Acoustic Echo 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 through the Web interface:

➢ **To configure echo cancellation:**

1. Configure line echo cancellation:
   a. Open the Voice Settings page (Setup menu > Signaling & Media tab > Media folder > Voice Settings).
   
   ![Figure 16-1: Enabling Echo Cancellation](image)

   - **Echo Canceller**: Enable

   b. From the 'Echo Canceller' drop-down list (EnableEchoCanceller), select Enable.

2. Enable acoustic echo cancellation for SBC calls:
   a. Open the Voice Settings page (Setup menu > Signaling & Media tab > Media folder > Voice Settings).
   b. Under the Network Echo Suppressor group:
      - '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
   c. In the Voice Settings page, configure the following parameters:
      - Open the IP Profiles table, and configure the 'Echo Canceller' parameter to Acoustic (see Configuring IP Profiles on page 417).
   d. 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:](image)

   - *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

### 16.2 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.
16. Media

**Note:**
- 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 193.
- For a detailed description of the parameters appearing on this page, see "Configuration Parameters Reference" on page 831.

➢ To access the fax and modem parameters:

1. Open the Fax/Modem/CID Settings page (Setup menu > Signaling & Media tab > Media folder > Fax/Modem/CID Settings).

![Figure 16-2: Fax/Modem/CID Settings Page]

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

### 16.2.1 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 193): 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).

### 16.2.2 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 184)
- G.711 Transport: switching to G.711 when fax/modem is detected (see "G.711 Fax / Modem Transport Mode" on page 186)
- Fax fallback to G.711 if T.38 is not supported (see "Fax Fallback" on page 187)
- Fax and modem bypass: a proprietary method that uses a high bit rate coder (see "Fax/Modem Bypass Mode" on page 188)
- NSE Cisco's Pass-through bypass mode for fax and modem (see "Fax / Modem NSE Mode" on page 189)
- Transparent with events: passing the fax / modem signal in the current voice coder with adaptations (see "Fax / Modem Transparent with Events Mode" on page 190)
- Transparent: passing the fax / modem signal in the current voice coder (see "Fax / Modem Transparent Mode" on page 190)
- RFC 2833 ANS Report upon Fax/Modem Detection (see "RFC 2833 ANS Report upon Fax/Modem Detection" on page 191)

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

### 16.2.2.1 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 184)
- 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 185)

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

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

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

### 16.2.2.1.1 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. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select **T.38 Relay**:

   ![Figure 16-3: Configuring Fax Signaling to T.38](image)

2. On 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).

### 16.2.2.1.2 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. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select **No Fax**:

   ![Figure 16-4: Configuring Fax Signaling to None](image)

2. On 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)
16.2.2.1.3 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 T38MaxBuffer=3000
  a=fmtp:100 T38MaxDatagram=122
  a=fmtp:100 T38FaxRateManagement=transferredTCF
  a=fmtp:100 T38FaxUdpEC=t38UDPRedundancy
  a=rtpmap: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.

16.2.2.2 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:
16. Media

- 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:

1. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select G.711 Transport:

   ![Figure 16-5: Configuring Fax Signaling to G.711](image)

   **Figure 16-5: Configuring Fax Signaling to G.711**

   **Fax Signaling Method**

   G.711 Transport

2. Click Apply.

16.2.2.3 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:

1. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select Fax Fallback:

   ![Figure 16-6: Configuring Fax Signaling to Fallback]

2. Click Apply.

16.2.2.4 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. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select No Fax.

2. On 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).
16. Media

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

16.2.2.5 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. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select **No Fax**.

2. On 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).
16.2.2.6 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. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select No Fax.

2. On 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).

16.2.2.7 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 407) 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. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select No Fax.
2. On 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 in the Coders table - see "Configuring Coder Groups" on page 407.
   c. 'Echo Canceller' (EnableEchoCanceller) - see "Configuring Echo Cancellation" on page 181.

   **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 188) or Transparent with Events modes (see "Fax / Modem Transparent with Events Mode" on page 190) for modem.

### 16.2.2.8 RFC 2833 ANS Report upon Fax/Modem Detection

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

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

1. Open the Gateway General Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway General Settings), and then from the 'Fax Signaling Method' drop-down list (IsFaxUsed), select **No Fax** or **Fax Fallback**.
2. On 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).

### 16.2.3 V.34 Fax Support

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

- Bypass mechanism for V.34 fax transmission (see "Bypass Mechanism for V.34 Fax Transmission" on page 192)
16.2.3.1 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. On the Fax/Modem/CID Settings page, do the following:
   a. Set the 'Fax Transport Mode' parameter to **Bypass** (FaxTransportMode = 2).
   b. Set the 'V.22 Modem Transport Type' parameter to **Enable Bypass** (V22ModemTransportType = 2).
   c. Set the 'V.23 Modem Transport Type' parameter to **Enable Bypass** (V23ModemTransportType = 2).
   d. Set the 'V.32 Modem Transport Type' parameter to **Enable Bypass** (V32ModemTransportType = 2).
   e. Set the 'V.34 Modem Transport Type' parameter to **Enable Bypass** (V34ModemTransportType = 2).

➢ **To use bypass mode for V.34 faxes, and T.38 for T.30 faxes:**
1. On 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).

16.2.3.2 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. On the Fax/Modem/CID Settings page, do the following:
   a. Set the 'Fax Transport Mode' parameter to **T.38 Relay** (FaxTransportMode = 1).
   b. Set the 'V.22 Modem Transport Type' parameter to **Disable** (V22ModemTransportType = 0).
   c. Set the 'V.23 Modem Transport Type' parameter to **Disable** (V23ModemTransportType = 0).
   d. Set the 'V.32 Modem Transport Type' parameter to **Disable** (V32ModemTransportType = 0).
   e. Set the 'V.34 Modem Transport Type' parameter to **Disable** (V34ModemTransportType = 0).

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**.
16. Media

e. Set the 'V.34 Modem Transport Type' parameter to **Disable**

(V34ModemTransportType = 0).

16.2.4 V.152 Support

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

For V.152 capability, the device supports T.38 as well as VBD codecs (i.e., G.711 A-law and G.711 μ-law). The selection of capabilities is performed using the Coder Groups table (see "Configuring Coder Groups" on page 407).

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 <udpPortA> 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.

16.3 Configuring RTP/RTCP Settings

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

16.3.1 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:

1. Open the RTP/RTCP Settings page (Setup menu > Signaling & Media menu > Media folder > RTP/RTCP Settings). The relevant parameters are listed under the General group, as shown below:
2. Set the ‘Dynamic Jitter Buffer Minimum Delay’ parameter (DJBufMinDelay) to the minimum delay (in msec) for the Dynamic Jitter Buffer.
4. Click Apply.

### 16.3.2 Configuring RFC 2833 Payload

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

1. Open the RTP/RTCP Settings page (Setup menu > Signaling & Media tab > Media folder > RTP/RTCP Settings).

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

3. Click Apply.

16.3.3 Configuring RTP Base UDP Port

You can configure the range (pool) of local UDP ports from which the device allocates ports to media (RTP, RTCP, and T.38) channels (legs). The maximum range of UDP ports is from 6,000 through to 65,535.

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

Within the port range, the device allocates the UDP ports per media channel (leg) in "jumps" (spacing) of 4, 5 (default) or 10, configured by the UdpPortSpacing parameter. For example, if the port range starts at 6000 and the UDP port spacing is 10, the available ports are 6000, 6010, 6020, 6030, and so on. Within the port range, the device assigns these ports randomly to the different media channels. For example, it allocates port 6000 to leg 1, port 6030 to leg 2, and port 6010 to leg 3.

You can configure the starting port (lower boundary) of the port range (default is 6000), using the BaseUDPPort parameter. Once configured, the port range is according to the following equation:

\[ \text{<BaseUDPPort parameter value> to 65,535} \]

Where, number of channels is the maximum number of purchased channels for the device (included in the installed License Key).

For example, if you configure the BaseUDPPort parameter to 6000, the port range is 6000 to 65,535.

You can also configure specific port ranges for specific SIP entities, using Media Realms (see "Configuring Media Realms" on page 319). You can configure each Media Realm with a different UDP port range and then associate the Media Realm with a specific IP Group, for example. However, the port range of the Media Realm must be within the range configured by the BaseUDPPort parameter.

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

To configure the RTP base UDP port:

1. Open the RTP/RTCP Settings page (Setup menu > Signaling & Media tab > Media
folder > RTP/RTCP Settings).

2. In the ‘RTP Base UDP Port’ field, configure the lower boundary of the UDP port range.

![Figure 16-7: Configuring RTP Base UDP Port](image)

3. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

![Note: The RTP port must be different from ports configured for SIP signaling traffic (i.e., ports configured for SIP Interfaces). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.](image)

### 16.4 Event Detection and Notification using X-Detect Header

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

The device can detect the following event types:

- **Answering Machine Detection (AMD):** Detects events that are related to the AMD feature. AMD detects whether an answering machine or live voice has answered the call. It can also be used to detect silence, or the beep sound played by an answering machine to indicate the end of the greeting message after which a voice message can be left. For more information on AMD, see “Answering Machine Detection (AMD)” on page 200.

- **Call Progress Tone (CPT):** Detects whether a specific tone, defined in the installed CPT file is received from the call. It can be used to detect the beep sound played by an answering machine (as mentioned above) and the busy, reorder and ring tones.

![Note: Event detection is supported only for calls using the G.711 coder.](image)

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

- **X-Detect header in the INVITE message received from the application server requesting a specific event detection:**
  
  ```
  X-Detect: Request=[event type to detect]
  ```

- **X-Detect header in the SIP response message -- SIP 183 (for early dialogs) or 200 OK (for confirmed dialogs) -- sent by the device to the application server specifying which of the requested events it can detect (absence of the X-Detect header indicates that the device cannot detect any of the events):**

  ```
  X-Detect: Response=[supported event types]
  ```

- **Each time the device detects the supported event, it sends an INFO message to the remote party with the following message body:**

  ```
  Content-Type: Application/X-Detect
  Type = [event type]
  Subtype = [subtype of each event type]
  ```
The table below lists the event types and subtypes that can be detected by the device. The text shown in the table are the strings used in the X-Detect header. The table also provides a summary of the required configuration. For SBC calls, event detection is enabled using the IPProfile_SBCHandleXDetect parameter in the IP Profiles table (see Configuring IP Profiles on page 417).

Table 16-1: Supported X-Detect Event Types

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Subtype</th>
<th>Description and Required Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD</td>
<td>Voice (live voice)</td>
<td>Event detection using the AMD feature. For more information, see Answering Machine Detection (AMD) on page 200.</td>
</tr>
<tr>
<td>AMD</td>
<td>Automata (answering machine)</td>
<td></td>
</tr>
<tr>
<td>AMD</td>
<td>Silence (no voice)</td>
<td></td>
</tr>
<tr>
<td>AMD</td>
<td>Unknown</td>
<td></td>
</tr>
<tr>
<td>AMD</td>
<td>Beep (greeting message of answering machine)</td>
<td></td>
</tr>
<tr>
<td>CPT</td>
<td>Busy</td>
<td>Event detection of tones using the CPT file. 1. Create a CPT file with the required tone types of the events that you want to detect. 2. Install the CPT file on the device. Note: To configure beep detection, see Detecting Answering Machine Beep on page 197.</td>
</tr>
<tr>
<td>CPT</td>
<td>Reorder</td>
<td></td>
</tr>
<tr>
<td>CPT</td>
<td>Ringtone</td>
<td></td>
</tr>
<tr>
<td>CPT</td>
<td>Beep (greeting message of answering machine)</td>
<td></td>
</tr>
</tbody>
</table>

16.4.1 Detecting Answering Machine Beeps

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

- **AMD-based Detection**: The device uses its beep detector that is integrated in the AMD feature. You can configure the beep detection timeout and beep detection sensitivity level (for more information, see "Configuring AMD" on page 202). To enable the AMD beep detection, the received INVITE message must contain an X-Detect header with the value "Request=AMD",

  \[X\text{-Detect: Request=AMD}\]

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

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

  \[X\text{-Detect: Request=CPT}\]

  For more information on the CPT file, see "Call Progress Tones File" on page 657.

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

- **AMD-detected Beep**: 

```
16.4.2 SIP Call Flow Examples of Event Detection and Notification

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

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

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac1566945480
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249>;tag=1c1505895240
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29758@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.7.20A.000.038
Content-Type: application/x-detect
Content-Length: 30
Type= AMD
SubType= AUTOMATA
```

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

```
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249>;tag=1c419779142
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow: REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.7.20A.000.038
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-START
```

3. Upon detection of the end of voice (i.e., end of the greeting message of the answering machine), the device sends the following INFO message to the application server:
INFO sip:sipp@172.22.2.9:5060 SIP/2.0
Via: SIP/2.0/UDP 172.22.168.249;branch=z9hG4bKac482466515
Max-Forwards: 70
From: sut <sip:3000@172.22.168.249;tag=1c419779142>
To: sipp <sip:sipp@172.22.2.9:5060>;tag=1
Call-ID: 1-29753@172.22.2.9
CSeq: 1 INFO
Contact: <sip:56700@172.22.168.249>
Supported: em,timer,replaces,path,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,UPDATE
User-Agent: Audiocodes-Sip-Gateway/v.7.20A.000.038
Content-Type: application/x-detect
Content-Length: 34
Type= PTT
SubType= SPEECH-END

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

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

1. The device receives a SIP message containing the X-Detect header from the remote application requesting beep detection:
   INVITE sip:101@10.33.2.53;user=phone SIP/2.0
   Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
   Max-Forwards: 70
   From: "anonymous"
   <sip:anonymous@anonymous.invalid>;tag=1c25298
   To: <sip:101@10.33.2.53;user=phone>
   Call-ID: 11923@10.33.2.53
   CSeq: 1 INVITE
   Contact: <sip:100@10.33.2.53>
   X-Detect: Request=AMD,CPT

2. The device sends a SIP response message to the remote party, listing the events in the X-Detect header that it can detect:
   SIP/2.0 200 OK
   Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
   From: "anonymous"
   <sip:anonymous@anonymous.invalid>;tag=1c25298
   To: <sip:101@10.33.2.53;user=phone>;tag=1c19282
   Call-ID: 11923@10.33.2.53
   CSeq: 1 INVITE
   Contact: <sip:101@10.33.2.53>
   X-Detect: Response=AMD,CPT

3. The device detects the beep of an answering machine and sends an INFO message to the remote party:
   INFO sip:101@10.33.2.53;user=phone SIP/2.0
   Via: SIP/2.0/UDP 10.33.2.53;branch=z9hG4bKac5906
   Max-Forwards: 70
   From: "anonymous"
   <sip:anonymous@anonymous.invalid>;tag=1c25298
   To: <sip:101@10.33.2.53;user=phone>
   Call-ID: 11923@10.33.2.53
   CSeq: 1 INVITE
   Contact: <sip:100@10.33.2.53>
   X-Detect: Response=AMD,CPT
   Content-Type: Application/X-Detect
16.5 Answering Machine Detection (AMD)

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

When the device detects what answered the call (human or machine), it can notify this detection type to, for example, a third-party application server used for automatic dialing applications. The X-Detect SIP header is used for requesting event detection and notification. For more information, see "Event Detection and Notification using X-Detect Header" on page 196. The device can also detect beeps played by an answering machine at the end of its greeting message. For more information, see "Detecting Answering Machine Beeps" on page 197.

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

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

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

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

Once you have provided AudioCodes with your database of recordings, AudioCodes compiles it into a loadable file. For a brief description of the file format and for installing the file on the device, see "AMD Sensitivity File" on page 667.
The device supports up to eight AMD algorithm suites called Parameter Suites, where each suite defines a range of detection sensitivity levels. Sensitivity levels refer to how accurately, based on AudioCodes' voice detection algorithms, the device can detect whether a human or machine has answered the call. Each level supports a different detection sensitivity to human and machine. For example, a specific sensitivity level may be more sensitive to detecting human than machine. In deployments where the likelihood of a call answered by an answering machine is low, it would be advisable to configure the device to use a sensitivity level that is more sensitive to human than machine. In addition, this allows you to tweak your sensitivity to meet local regulatory rules designed to protect consumers from automatic dialers (where, for example, the consumer picks up the phone and hears silence). Each suite can support up to 16 sensitivity levels (0 to 15), except for Parameter Suite 0, which supports up to 8 levels (0 to 7). The default, pre-installed AMD Sensitivity file, based on North American English, provides the following Parameter Suites:

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

As Parameter Suite 1 provides a greater range of detection sensitivity levels (i.e., higher detection resolution), this may be the preferable suite to use in your deployment. The detected AMD type (human or machine) and success of detecting it correctly are sent in CDR and Syslog messages. For more information, see "Syslog Fields for Answering Machine Detection (AMD)" on page 796.

The Parameter Suite and sensitivity level can be applied globally for all calls, or for specific calls using IP Profiles. For enabling AMD and selecting the Parameter Suite and sensitivity level, see "Configuring AMD" on page 202.

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

Table 16-2: Approximate AMD Normal Detection Sensitivity - Parameter Suite 0 (Based on North American English)

<table>
<thead>
<tr>
<th>AMD Detection Sensitivity</th>
<th>Performance</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Success Rate for Live Calls</td>
<td>Success Rate for Answering Machine</td>
</tr>
<tr>
<td>0 (Best for Answering Machine)</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>82.56%</td>
<td>97.10%</td>
</tr>
<tr>
<td>2</td>
<td>85.87%</td>
<td>96.43%</td>
</tr>
<tr>
<td>3</td>
<td>88.57%</td>
<td>94.76%</td>
</tr>
<tr>
<td>4</td>
<td>88.94%</td>
<td>94.31%</td>
</tr>
<tr>
<td>5</td>
<td>90.42%</td>
<td>91.64%</td>
</tr>
<tr>
<td>6</td>
<td>90.66%</td>
<td>91.30%</td>
</tr>
<tr>
<td>7 (Best for Live Calls)</td>
<td>94.72%</td>
<td>76.14%</td>
</tr>
</tbody>
</table>
### Table 16-3: Approximate AMD High Detection Sensitivity - Parameter Suite 1 (Based on North American English)

<table>
<thead>
<tr>
<th>AMD Detection Sensitivity</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Success Rate for Live Calls</td>
</tr>
<tr>
<td>0 (Best for Answering Machine)</td>
<td>72%</td>
</tr>
<tr>
<td>1</td>
<td>77%</td>
</tr>
<tr>
<td>2</td>
<td>79%</td>
</tr>
<tr>
<td>3</td>
<td>80%</td>
</tr>
<tr>
<td>4</td>
<td>84%</td>
</tr>
<tr>
<td>5</td>
<td>86%</td>
</tr>
<tr>
<td>6</td>
<td>87%</td>
</tr>
<tr>
<td>7</td>
<td>88%</td>
</tr>
<tr>
<td>8</td>
<td>90%</td>
</tr>
<tr>
<td>9</td>
<td>90%</td>
</tr>
<tr>
<td>10</td>
<td>91%</td>
</tr>
<tr>
<td>11</td>
<td>94%</td>
</tr>
<tr>
<td>12</td>
<td>94%</td>
</tr>
<tr>
<td>13</td>
<td>95%</td>
</tr>
<tr>
<td>14</td>
<td>96%</td>
</tr>
<tr>
<td>15 (Best for Live Calls)</td>
<td>97%</td>
</tr>
</tbody>
</table>

### 16.5.1 Configuring AMD

You can configure AMD for all calls using global AMD parameters or for specific calls using IP Profiles. The procedure below describes how to configure AMD for all calls. To configure AMD for specific calls, use the AMD parameters in the IP Profiles table (see "Configuring IP Profiles" on page 417).

- **To configure AMD for all calls:**
  1. Open the DSP Settings page (Setup menu > Signaling & Media tab > Media folder > DSP Settings):
  2. From the 'IPMedia Detectors' drop-down list (EnableDSPIPMDetectors), select Enable to enable AMD.
3. Scroll down to the Answer Machine Detector group:

**Figure 16-8: Configuring AMD**

<table>
<thead>
<tr>
<th>ANSWER MACHINE DETECTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Answer Machine Detector Sensitivity Parameter Suite</td>
</tr>
<tr>
<td>Answer Machine Detector Sensitivity</td>
</tr>
<tr>
<td>Answer Machine Detector Sensitivity Level</td>
</tr>
<tr>
<td>Answer Machine Detector Beep Detection Timeout</td>
</tr>
<tr>
<td>Answer Machine Detector Beep Detection Sensitivity</td>
</tr>
</tbody>
</table>

4. Select the AMD algorithm suite:
   a. In the 'Answer Machine Detector Sensitivity Parameter Suite' field, select the required Parameter Suite included in the installed AMD Sensitivity file.
   b. In the 'Answer Machine Detector Sensitivity' field, enter the required detection sensitivity level of the selected Parameter Suite.

5. Configure the answering machine beep detection:
   a. In the 'Answer Machine Detector Beep Detection Timeout' field (AMDBeepDetectionTimeout), enter the duration that the beep detector operates from when detection is initiated.
   b. In the 'Answer Machine Detector Beep Detection Sensitivity' field (AMDBeepDetectionSensitivity), enter the AMD beep detection sensitivity level.

6. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

For a complete list of AMD-related parameters, see "IP Media Parameters" on page 935.

16.6 **Automatic Gain Control (AGC)**

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

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

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

1. Open the DSP Settings page (Setup menu > Signaling & Media tab > Media folder > DSP Settings):

   ![AGC Parameters](image)

   **Figure 16-9: AGC Parameters**

2. Configure the following parameters:
   - 'Enable AGC' (EnableAGC) - Enables the AGC mechanism.
   - 'AGC Slope' (AGCGainSlope) - Determines the AGC convergence rate.
   - 'AGC Redirection' (AGCRedirection) - Determines the AGC direction.
   - 'AGC Target Energy' - Defines the signal energy value (dBm) that the AGC attempts to attain.
   - 'AGC Minimum Gain' (AGCMinGain) - Defines the minimum gain (in dB) by the AGC when activated.
   - 'AGC Maximum Gain' (AGCMaxGain) - Defines the maximum gain (in dB) by the AGC when activated.
   - 'AGC Disable Fast Adaptation' (AGCDisableFastAdaptation) - Enables the AGC Fast Adaptation mode.

3. When using AGC with the SBC application, the 'Transcoding Mode' (TranscodingMode) parameter must be set to Force. The parameter can either be the global parameter or per IP Profile.

4. Click Apply.

### 16.7 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 cryptographic key exchange mechanism to negotiate the keys. To negotiate the keys, the device supports the Session Description Protocol Security Descriptions (SDES) protocol (according to RFC 4568) or Datagram Transport Layer Security (DTLS) protocol for SBC calls. For more information on DTLS, see SRTP using DTLS Protocol on page 207. The key exchange is done by adding the ‘a=crypto’ attribute to the SDP. This attribute is used (by both sides) to declare the various supported cipher suites and to attach the encryption key. If negotiation of the encryption data is successful, the call is established.

SRTP supports the following cipher suites (all other suites are ignored):
- AES_CM_128_HMAC_SHA1_32
- AES_CM_128_HMAC_SHA1_80

When the device is the offering side (SDP offer), it can generate a Master Key Identifier (MKI). You can configure the MKI size globally (using the SRTPTxPacketMKISize...
parameter) or per SIP entity (using the IP Profile parameter, IpProfile_MKISize). The length of the MKI is limited to four bytes. If the remote side sends a longer MKI, the key is ignored.

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

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

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

- UNENCRYPTED_SRTP
- UNENCRYPTED_SRTCP
- UNAUTHENTICATED_SRTP

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

Below is an example of crypto attributes usage:

```
a=crypto:1 AES_CM_128_HMAC_SHA1_80
inline:PsKoMpHICg+b5X0YluSvNz1mEh/dAu
a=crypto:2 AES_CM_128_HMAC_SHA1_32
inline:IsPloGkBf9a+c6XzRuMqHldEiAd
```

The device also supports symmetric MKI negotiation, whereby it can forward the MKI size received in the SDP offer 'a=crypto' line in the SDP answer. You can enable symmetric MKI globally (using the EnableSymmetricMKI parameter) or per SIP entity (using the IP Profile parameter, IpProfile_EnableSymmetricMKI and IpProfile_SBCEnforceMKISize). For more information on symmetric MKI, see "Configuring IP Profiles" on page 417.

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

**Note:**

- For a detailed description of the SRTP parameters, see "Configuring IP Profiles" on page 417 and "SRTP Parameters" on page 869.
- When SRTP is used, the channel capacity may be reduced.

The procedure below describes how to configure SRTP through the Web interface.
To enable and configure SRTP:

1. Open the Media Security page (Setup menu > Signaling & Media tab > Media folder > Media Security).

2. From the 'Media Security' drop-down list (EnableMediaSecurity), select Enable to enable SRTP.

3. Configure the other SRTP parameters as required.

4. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.
16.7.1 SRTP using DTLS Protocol

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

DTLS support is important for deployments with WebRTC. WebRTC requires that media channels be encrypted through DTLS for SRTP key exchange. Negotiation of SRTP keys through DTLS is done during the DTLS handshake between WebRTC client and peer. For more information on WebRTC, see "WebRTC" on page 566.

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

\a=setup:actpass
\a=fingerprint: SHA-1

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

➢ To configure DTLS:

1. In the TLS Context table (see "Configuring TLS Certificate Contexts" on page 105), configure a TLS Context with the DTLS version (TLSContexts_DTLSVersion).
2. Open the IP Groups table (see "Configuring IP Groups" on page 345) and for the IP Group associated with the SIP entity, assign it the TLS Context for DTLS, using the 'DTLS Context' parameter (IPGroup_DTLSContext).
3. Open the IP Profiles table (see "Configuring IP Profiles" on page 417) and for the IP Profile associated with the SIP entity, configure the following:
   • Configure the 'SBC Media Security Mode' parameter (IPProfile_SBCMediaSecurityBehavior) to SRTP or Both.
   • Configure the 'Media Security Method' parameter (IPProfile_SBCMediaSecurityMethod) to DTLS.
• Configure the 'RTCP Mux' parameter (IpProfile_SBCRTCPMux) to **Supported**. Multiplexing is required as the DTLS handshake is done for the port used for RTP and thus, RTCP and RTP must be multiplexed onto the same port.

• Configure the ini file parameter, SbcDtlsMtu (or CLI command configure voip > sbc settings > sbc-dtls-mtu) to define the maximum transmission unit (MTU) size for the DTLS handshake.

**Note:**

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

This section describes configuration for various supported services.

17.1 DHCP Server Functionality

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

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

The DHCP server is activated when you configure a valid entry in the DHCP Servers table (see "Configuring the DHCP Server" on page 209) and associate it with an active IP network interface (listed in the IP Interfaces table). When an IP phone on the LAN requests an IP address, the DHCP server allocates one from the address pool. In scenarios of duplicated IP addresses on the LAN (i.e., an unauthorized network device using one of the IP addresses of the DHCP address pool), the DHCP server detects this condition using an Address Resolution Protocol (ARP) request and temporarily blacklists the duplicated address.

You can also configure the DHCP server to respond only to DHCPDiscover requests from DHCP clients that contain a specific value for Option 60 (Vendor Class Identification). For more information, see "Configuring the Vendor Class Identifier" on page 214.

17.1.1 Configuring the DHCP Server

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

<table>
<thead>
<tr>
<th>DHCP Option Code</th>
<th>DHCP Option Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Option 53</td>
<td>DHCP Message Type</td>
</tr>
<tr>
<td>Option 54</td>
<td>DHCP Server Identifier</td>
</tr>
<tr>
<td>Option 51</td>
<td>IP Address Lease Time</td>
</tr>
<tr>
<td>Option 1</td>
<td>Subnet Mask</td>
</tr>
<tr>
<td>Option 3</td>
<td>Router</td>
</tr>
<tr>
<td>Option 6</td>
<td>Domain Name Server</td>
</tr>
<tr>
<td>Option 44</td>
<td>NetBIOS Name Server</td>
</tr>
<tr>
<td>Option 46</td>
<td>NetBIOS Node Type</td>
</tr>
<tr>
<td>Option 42</td>
<td>Network Time Protocol Server</td>
</tr>
</tbody>
</table>
Once you have configured the DHCP server, you can configure the following:

- DHCP Vendor Class Identifier names (DHCP Option 60) - see "Configuring the Vendor Class Identifier" on page 214
- Additional DHCP Options - see "Configuring Additional DHCP Options" on page 215
- Static IP addresses for DHCP clients - see "Configuring Static IP Addresses for DHCP Clients" on page 217

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

To view and delete currently serviced DHCP clients, see "Viewing and Deleting DHCP Clients" on page 218.

The following procedure describes how to configure the DHCP server through the Web interface. You can also configure it through ini file (DhcpServer) or CLI (configure network > dhcp-server server <index>).

**To configure the device’s DHCP server:**

1. Open the DHCP Servers page (Setup menu > IP Network tab > Advanced folder > DHCP Servers).
2. Click **New**; the following dialog box appears:

*Figure 17-1: DHCP Servers Table - Add Dialog Box*

3. Configure a DHCP server according to the parameters described in the table below.

4. Click **Apply**.

*Table 17-2: DHCP Servers Table Parameter Descriptions*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Index | Defines an index number for the new table row. **Note:**  
- Each row must be configured with a unique index.  
- Currently, only one index row can be configured. |
| dhcp server <index> | |
| Interface Name | Associates an IP interface on which the DHCP server operates. The IP interfaces are configured in the IP Interfaces table (see "Configuring IP Network Interfaces" on page 137). By default, no value is defined.  
| network-if |  
| [DhcpServer_InterfaceName] | |
| Start IP Address | Defines the starting IP address (IPv4 address in dotted-decimal format) of the IP address pool range used by the DHCP server to allocate addresses. The default value is 192.168.0.100. **Note:** The IP address must belong to the same subnet as the associated interface’s IP address.  
<p>| start-address |<br />
| [DhcpServer_StartIPAddress] | |
| End IP Address | Defines the ending IP address (IPv4 address in dotted-decimal format) of the IP address pool range used by the DHCP server |
| end-address |<br />
| [DhcpServer_EndIPAddress] | |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DHCP Server End IP Address</td>
<td>To allocate addresses. The default value is 192.168.0.149. <strong>Note:</strong> The IP address must belong to the same subnet as the associated interface's IP address and must be &quot;greater or equal&quot; to the starting IP address defined in 'Start IP Address'.</td>
</tr>
<tr>
<td>Subnet Mask</td>
<td>Defines the subnet mask (for IPv4 addresses) for the DHCP client. The value is sent in DHCP Option 1 (Subnet Mask). The default value is 0.0.0.0. <strong>Note:</strong> The value must be &quot;narrower&quot; or equal to the subnet mask of the associated interface's IP address. If set to &quot;0.0.0.0&quot;, the subnet mask of the associated interface is used.</td>
</tr>
<tr>
<td>Lease Time</td>
<td>Defines the duration (in minutes) of the lease time to a DHCP client for using an assigned IP address. The client needs to request a new address before this time expires. The value is sent in DHCP Option 51 (IP Address Lease Time). The valid value range is 0 to 214,748,3647. The default is 1440. When set to 0, the lease time is infinite.</td>
</tr>
<tr>
<td>DNS</td>
<td></td>
</tr>
<tr>
<td>DNS Server 1</td>
<td>Defines the IP address (IPv4) of the primary DNS server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 6 (Domain Name Server). The default value is 0.0.0.0.</td>
</tr>
<tr>
<td>DNS Server 2</td>
<td>Defines the IP address (IPv4) of the secondary DNS server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 6 (Domain Name Server). The default value is 0.0.0.0.</td>
</tr>
<tr>
<td>NetBIOS</td>
<td></td>
</tr>
<tr>
<td>NetBIOS Name Server</td>
<td>Defines the IP address (IPv4) of the NetBIOS WINS server that is available to a Microsoft DHCP client. The value is sent in DHCP Option 44 (NetBIOS Name Server). The default value is 0.0.0.0.</td>
</tr>
<tr>
<td>NetBIOS Node Type</td>
<td>Defines the node type of the NetBIOS WINS server for a Microsoft DHCP client. The value is sent in DHCP Option 46 (NetBIOS Node Type).</td>
</tr>
<tr>
<td>Time and Date</td>
<td></td>
</tr>
<tr>
<td>NTP Server 1</td>
<td>Defines the IP address (IPv4) of the primary NTP server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 42 (Network Time Protocol Server). The default value is 0.0.0.0.</td>
</tr>
<tr>
<td>NTP Server 2</td>
<td>Defines the IP address (IPv4) of the secondary NTP server that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 42 (Network Time Protocol Server).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Time Offset</strong></td>
<td>Defines the Greenwich Mean Time (GMT) offset (in seconds) that the DHCP server assigns to the DHCP client. The value is sent in DHCP Option 2 (Time Offset). The valid range is -43200 to 43200. The default is 0.</td>
</tr>
<tr>
<td><strong>Boot File</strong></td>
<td></td>
</tr>
<tr>
<td><strong>TFTP Server Name</strong></td>
<td>Defines the IP address or name of the TFTP server that the DHCP server assigns to the DHCP client. The TFTP server typically stores the boot file image, defined in the 'Boot file name' parameter (see below). The value is sent in DHCP Option 66 (TFTP Server Name). The valid value is a string of up to 80 characters. By default, no value is defined.</td>
</tr>
</tbody>
</table>
| **Boot File Name**                    | Defines the name of the boot file image for the DHCP client. The boot file stores the boot image for the client. The boot image is typically the operating system the client uses to load (downloaded from a boot server). The value is sent in DHCP Option 67 (Bootfile Name). To define the server storing the file, use the 'TFTP Server' parameter (see above). The valid value is a string of up to 256 characters. By default, no value is defined. The name can also include the following case-sensitive placeholder strings that are replaced with actual values if the 'Expand Boot-file Name' parameter is set to Yes:  
  - `<MAC>`: Replaced by the MAC address of the client (e.g., `boot_<MAC>.ini`). The MAC address is obtained in the client's DHCP request.  
  - `<IP>`: Replaced by the IP address assigned by the DHCP server to the client. |
| **Expand Boot-File Name**             | Enables the use of the placeholders in the boot file name, defined in the 'Boot file name' parameter.  
  - [0] No  
  - [1] Yes (default) |
<p>| <strong>Override Router</strong>                   | Defines the IP address (IPv4 in dotted-decimal notation) of the default router that the DHCP server assigns the DHCP client. The value is sent in DHCP Option 3 (Router). The default value is 0.0.0.0. If not specified (empty or &quot;0.0.0.0&quot;), the IP address of the default gateway configured in the IP Interfaces table for the IP network interface that you associated with the DHCP server (see the 'Interface Name' parameter above) is used. |
| <strong>SIP</strong>                               | Defines the IP address or DNS name of the SIP server that the DHCP server assigns the DHCP client. The client uses this SIP server for its outbound SIP requests. The value is sent in DHCP Option 120 (SIP Server). After defining the parameter, use the 'SIP server type' parameter (see below) to define the type of |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address (FQDN or IP address).</td>
<td>The valid value is a string of up to 256 characters. The default is 0.0.0.0.</td>
</tr>
<tr>
<td>SIP Server Type</td>
<td>Defines the type of SIP server address. The actual address is defined in the 'SIP server' parameter (see above). Encoding is done per SIP Server Type, as defined in RFC 3361.</td>
</tr>
<tr>
<td>sip-server-type [DhcpServer_SipServerType]</td>
<td>• [0] DNS names = (Default) The 'SIP server' parameter is configured with an FQDN of the SIP server.</td>
</tr>
<tr>
<td></td>
<td>• [1] IP address = The 'SIP server' parameter configured with an IP address of the SIP server.</td>
</tr>
</tbody>
</table>

17.1.2 Configuring the Vendor Class Identifier

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

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

The following procedure describes how to configure the DHCP VCIs through the Web interface. You can also configure it through ini file (DhcpVendorClass) or CLI (configure network > dhcp-server vendor-class).

- To configure DHCP Vendor Class Identifiers:
  1. Open the DHCP Servers table (see "Configuring the DHCP Server" on page 209).
  2. Select the row of the desired DHCP server for which you want to configure VCIs, and then click the DHCP Vendor Class link located below the table; the DHCP Vendor Class table opens.
  3. Click New; the following dialog box appears:

```
Figure 17-2: DHCP Vendor Class Table - Add Dialog Box
```

4. Configure a VCI for the DHCP server according to the parameters described in the table below.
5. Click Apply.
Table 17-3: DHCP Vendor Class Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>dhcp vendor-class &lt;index&gt;</td>
<td>Associates the VCI table entry with a DHCP server that you configured in &quot;Configuring the DHCP Server&quot; on page 209. <strong>Note:</strong> Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0.</td>
</tr>
<tr>
<td>Vendor Class Identifier</td>
<td>Defines the value of the VCI DHCP Option 60. The valid value is a string of up to 80 characters. By default, no value is defined.</td>
</tr>
</tbody>
</table>

17.1.3 Configuring Additional DHCP Options

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

The following procedure describes how to configure DHCP Options through the Web interface. You can also configure it through ini file (DhcpOption) or CLI (configure network > dhcp-server option).

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

- **To configure DHCP Options:**
  1. Open the DHCP Servers table (see "Configuring the DHCP Server" on page 209).
  2. Select the row of the desired DHCP server for which you want to configure additional DHCP Options, and then click the **DHCP Option** link located below the table; the DHCP Option table opens.
3. Click **New**; the following dialog box appears:

   **Figure 17-3: DHCP Option Table - Add Dialog Box**

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

5. Click **Apply**.

**Table 17-4: DHCP Option Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index dhcp option</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>DHCP Server Index dhcp-server-number</td>
<td>Associates the DHCP Option table entry with a DHCP server that you configured in &quot;Configuring the DHCP Server&quot; on page 209. <strong>Note:</strong> Currently, only one DHCP server (Index 0) can be configured and therefore, the parameter is always set at 0.</td>
</tr>
<tr>
<td>Option option dhcpOption_Option</td>
<td>Defines the code of the DHCP Option. The valid value is 1 to 254. The default is 159. For example, for DHCP Option 150 (Cisco proprietary for defining multiple TFTP server IP addresses), enter the value 150.</td>
</tr>
<tr>
<td>Type type dhcpOption_Type</td>
<td>Defines the format (type) of the DHCP Option value that is configured in the 'Value' parameter (see below).</td>
</tr>
<tr>
<td>Value value dhcpOption_Value</td>
<td>Defines the value of the DHCP Option. For example, if you are using Option 66, the parameter is used for specifying the TFTP provisioning server (e.g., <a href="http://192.168.3.155:5000/provisioning/">http://192.168.3.155:5000/provisioning/</a>). The valid value is a string of up to 256 characters. By default, no value is defined. For IP addresses, the value can be one or more.</td>
</tr>
</tbody>
</table>
### 17.1.4 Configuring Static IP Addresses for DHCP Clients

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

The following procedure describes how to configure static IP addresses for DHCP clients through the Web interface. You can also configure it through ini file (DhcpStaticIP) or CLI (configure network > dhcp-server static-ip <index>).

1. To configure static IP addresses for DHCP clients:
   1. Open the DHCP Servers table (see "Configuring the DHCP Server" on page 209).
   2. Select the row of the desired DHCP server for which you want to configure static IP addresses for DHCP clients, and then click the **DHCP Static IP** link located below the table; the DHCP Static IP table opens.
3. Click **New**; the following dialog box appears:

   **Figure 17-4: DHCP Static IP Table - Add Dialog Box**

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

5. Click **Apply**.

   **Table 17-5: DHCP Static IP Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>dhcp static-ip &lt;index&gt;</td>
<td>Defines the &quot;reserved&quot;, static IP address (IPv4) to assign the DHCP client. The default is 0.0.0.0.</td>
</tr>
<tr>
<td>DHCP Server Index</td>
<td>Associates the DHCP Static IP table entry with a DHCP server that you configured in &quot;Configuring the DHCP Server&quot; on page 209.</td>
</tr>
<tr>
<td>IP Address</td>
<td>Defines the DHCP client by MAC address (in hexadecimal format). The valid value is a string of up to 20 characters. The format includes six groups of two hexadecimal digits, each separated by a colon. The default MAC address is 00:90:8f:00:00:00.</td>
</tr>
</tbody>
</table>

**17.1.5 Viewing and Deleting DHCP Clients**

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

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

- To view DHCP clients:
# show network dhcp clients

- To view DHCP clients according to IP address:
  # show network dhcp ip

- To view DHCP clients according to MAC address:
  # show network dhcp mac

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

To view or delete DHCP clients:

1. Open the DHCP Servers table (see "Configuring the DHCP Server" on page 209).
2. Select the row of the desired DHCP server for which you want to view DHCP clients, and then click the DHCP Clients link located below the table; the DHCP Clients table opens:

   ![Figure 17-5: DHCP Clients Table](image)

   The table displays the following per client:
   - **Index**: Table index number.
   - **DHCP Server Index**: The index number of the configured DHCP server scope in the DHCP Server table (see "Configuring the DHCP Server" on page 209) with which the client is associated.
   - **IP Address**: IP address assigned to the DHCP client by the DHCP server.
   - **MAC Address**: MAC address of the DHCP client.
   - **Lease Expiration**: Date on which the lease of the DHCP client's IP address obtained from the DHCP server expires.
3. To delete a client:
   a. Select the table row index of the DHCP client that you want to delete.
   b. Click the Action button, and then from the drop-down menu, choose Delete; a confirmation message appears.
   c. Click OK to confirm deletion.

### 17.2 SIP-based Media Recording

The device can record SIP-based media call sessions traversing it. The media recording support is in accordance with the Session Recording Protocol (siprec), which describes architectures for deploying session recording solutions and specifies requirements for extensions to SIP that will manage delivery of RTP media to a recording device. The siprec protocol is based on RFC 6341 (Use Cases and Requirements for SIP-Based Media Recording), Session Recording Protocol (draft-ietf-siprec-protocol-02), and Architecture (draft-ietf-siprec-architecture-03).
Warning for Deployments in France: The device supports SIP-based Media Recording (SIPREC) according to RFC 6341. As such, you must adhere to the Commission Nationale Informatique et Liberté’s (CNIL) directive (http://www.cnil.fr/english/data-protection/rights-and-obligations/) and be aware that article R226-15 applies penalties to the malicious interception, diversion, use or disclosure of correspondence sent, transmitted or received by means of telecommunication, or the setting up of a device designed to produce such interceptions.

Note:
- The SIP-based Media Recording feature is available only if the device is installed with a License Key that includes this feature. For installing a License Key, see "License Key" on page 669. The License Key also specifies the maximum number of supported SIP recording sessions.
- The device supports up to 12,000 concurrent SIPRec sessions. For Mediant VE, capacity depends on instance size. This capacity assumes that there are no other concurrent, regular (non-SIPRec) voice sessions.

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

The siprec protocol specifies the use of SIP, SDP, and RTP to establish a Recording Session (RS) from the Session Recording Client (SRC), which is on the path of the Communication Session (CS), to a Session Recording Server (SRS) at the recording equipment. The device functions as the SRC, sending recording sessions to a third-party SRS, as shown in the figure below.
The device can record calls between two IP Groups. The type of calls to record can be specified by source and/or destination prefix number or SIP Request-URI, as well as by call initiator. The side ("leg") on which the recording is done must be specified. Specifying the leg is important as it determines the various call media attributes of the recorded RTP (or SRTP) such as coder type.

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

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

The device can send recorded SBC calls to multiple SRSs. To achieve this, you can configure up to three groups of SRSs, where each group can contain one SRS (standalone), or two SRSs operating in an active-standby (1+1) mode for SRS redundancy. The device sends both SIP signaling and RTP to all standalone SRSs.

For SRS redundancy, the device sends SIP signaling to all SRSs (active and standby) in the SRS redundancy groups, but sends RTP only to the active SRSs. If during a recorded call session, the standby SRS detects that the active SRS has gone offline, the standby SRS sends a re-INVITE to the device and the device then sends the recorded RTP to the standby SRS instead (which now becomes the active SRS). For new calls, if the device receives no response or a reject response from the active SRS to its' sent INVITE message, the device sends the recorded call to the standby SRS.

**Figure 17-6: Multiple SRSs (Standalone and Redundancy)**

![Diagram showing recording client (SRC) communicating with AudioCodes Device, which then forwards recordings to multiple SRSs. SRSs are shown as both standalone and in redundancy groups.](image-url)
Note: SRS redundancy is a license-dependent feature and is available only if it is included in the License Key installed on the device. The license defines the maximum number of recorded calls that can be supported for SRS redundancy. For example, assume that your deployment has one standalone SRS and one SRS redundancy group and you want to record 100 call sessions. For the regular SIPRec license, you would need to order 200 SIPRec sessions (100 for the standalone SRS and 100 for the active SRS in the SRS redundancy group). For the SRS redundancy license, you would need to order 100 SIPRec Redundancy sessions to enable the SRS redundancy feature and support the number of recorded sessions. For more information and for ordering the feature, please contact your AudioCodes sales representative.

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

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

The SRS can respond with 'a=recvonly' for immediate recording or 'a=inactive' if recording is not yet needed, and send re-INVITE at any later time with the desired RTP/SRTP mode change. If a re-INVITE is received in the original call (e.g. when a call is on hold), the device sends another re-INVITE with two 'm=' lines to the SRS with the updated RTP/SRTP data. If the recorded leg uses SRTP, the device can send the media streams to the SRS as SRTP; otherwise, the media streams are sent as RTP to the SRS.

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

```
INVITE sip:VSRP@1.9.64.253 SIP/2.0
Via: SIP/2.0/UDP 192.168.241.44:5060;branch=z9hG4bKac505782914
Max-Forwards: 10
From: <sip:192.168.241.44>;tag=1c505764207
To: <sip:VSRP@1.9.64.253>
Call-ID: 5057630972412010111570192.168.241.44
CSeq: 1 INVITE
Contact: <sip:192.168.241.44:5060>;src
Supported: replaces,resource-priority
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
Require: siprec
User-Agent: Mediant /v.7.20A.000.038
```
Content-Type: multipart/mixed;boundary=boundary_ac1fffff85b
Content-Length: 1832

--boundary_ac1fffff85b
Content-Type: application/sdp
v=0
o=AudiocodesGW 921244928 921244893 IN IP4 10.33.8.70
s=SBC-Call
c=IN IP4 10.33.8.70
t=0 0
m=audio 6020 RTP/AVP 8 96
c=IN IP4 10.33.8.70
a=ptime:20
a=sendonly
a=label:1
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
m=audio 6030 RTP/AVP 8 96
c=IN IP4 10.33.8.70
a=ptime:20
a=sendonly
a=label:2
a=rtpmap:8 PCMA/8000
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15

--boundary_ac1fffff85b
Content-Type: application/rs-metadata
Content-Disposition: recording-session

<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
  <datamode>complete</datamode>
  <group id="00000000-0000-0000-0000-00003a36c4e3">
    <associate-time>2010-01-24T01:11:57Z</associate-time>
  </group>
  <session id="0000-0000-0000-0000-00000000d0d71a52">
    <group-ref>00000000-0000-0000-0000-00003a36c4e3</group-ref>
    <start-time>2010-01-24T01:11:57Z</start-time>
    <ac:AvayaUCID xmlns="urn:ietf:params:xml:ns:Avaya">FA080030C4E34B5B9E59</ac:AvayaUCID>
    <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>
  </session>
  <stream id="00000000-0000-0000-0000-00000000d0d71a52" session="0000-0000-0000-0000-00000000d0d71a52"/>
17.2.1 Enabling SIP-based Media Recording

The following procedure describes how to enable the SIP-based media Recording feature.

➢ To enable SIP-based media recording:

1. Open the SIP Recording Settings page (Setup menu > Signaling & Media tab > SIP Recording folder > SIP Recording Settings).

   Figure 17-7: Enabling SIPRec

   ![SIP Recording Settings](image)

2. From the 'SIP Recording Application' drop-down list, select **Enable**.
3. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.

   **Note:** SIPRec is a license-dependant feature and is available only if it is included in the License Key installed on the device. For more information, see SIP-based Media Recording on page 219.

17.2.2 Configuring SIP Recording Rules

The SIP Recording Rules table lets you configure up to 30 SIP-based media recording rules. A SIP Recording rule defines call routes that you want to record. For an overview of the feature, see "SIP-based Media Recording" on page 219.

**Note:** To view the total number of currently active SIPRec signaling sessions, use the CLI command **show voip calls statistics siprec**. For more information, refer to the **CLI Reference Guide**.

The following procedure describes how to configure SIP Recording rules through the Web interface. You can also configure it through ini file (SIPRecRouting) or CLI (configure voip > sip-definition sip-recording sip-rec-routing).

➢ To configure a SIP Recording Routing rule:

1. Open the SIP Recording Rules table (Setup menu > Signaling & Media tab > SIP Recording folder > SIP Recording Rules).
2. Click **New**; the following dialog box appears:

![Figure 17-8: SIP Recording Rules Table - Add Dialog Box](image)

The figure above shows a configuration example where the device records calls made by IP Group "ITSP" to IP Group "IP-PBX" that have the destination number prefix "1800". The device records the calls from the leg interfacing with IP Group "IP PBX" (peer) and sends the recorded media to IP Group "SRS-1". SRS redundancy has also been configured, where IP Group "SRS-1" is the active SRS and IP Group "SRS-2" the standby SRS.

3. Configure a SIP recording rule according to the parameters described in the table below.

4. Click **Apply**, and then save your settings to flash memory.

### Table 17-6: SIP Recording Rules Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index [SIPRecRouting_Index]</td>
<td>Defines an index number for the new table record.</td>
</tr>
</tbody>
</table>
| Recorded IP Group recorded-ip-group-name [SIPRecRouting_RecordedIPGroupName] | Defines the IP Group participating in the call and the recording is done on the leg interfacing with this IP Group. To configure IP Groups, see "Configuring IP Groups" on page 345. By default, all IP Groups are defined (Any). **Note:**  
  - The parameter is mandatory.  
  - For an SBC RTP-SRTP session, the recorded IP Group must be set to the RTP leg if recording is required to be RTP, or set to the SRTP leg if recording is required to be SRTP. |
| Recorded Source Prefix recorded-src-prefix [SIPRecRouting_RecordedSourcePrefix] | Defines calls to record based on source number or URI. By default, all source numbers or URIs are defined (*). |
| Recorded Destination Prefix recorded-dst-prefix [SIPRecRouting_RecordedDestinationPrefix] | Defines calls to record based on destination number or URI. By default, all destination numbers or URIs are defined (*). |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peer IP Group&lt;br&gt;peer-ip-group-name&lt;br&gt;[SIPRecRouting_PeerIPGroupName]</td>
<td>Defines the peer IP Group that is participating in the call. By default, all IP Groups are defined <em>(Any)</em>.</td>
</tr>
</tbody>
</table>
| Caller<br>caller<br>[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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Recording Server (SRS) IP Group<br>srs-ip-group-name<br>[SIPRecRouting_SRSIPGroupName] | Defines the IP Group of the recording server (SRS). By default, no value is defined.  
**Note:**  
- The parameter is mandatory.  
- The SIP Interface used for communicating with the SRS is according to the SRD assigned to the SRS IP Group (in the IP Groups table). |
| Redundant Recording Server (SRS) IP Group<br>srs-red-ip-group-name<br>[SIPRecRouting_SRSRedundantIPGroupName] | Defines the IP Group of the redundant SRS in the active-standby pair for SRS redundancy. By default, no value is defined.  
**Note:** The IP Group of this redundant SRS must be different to the IP Group of the main SRS (see 'Recording Server (SRS) IP Group' parameter). |

#### 17.2.3 Configuring SIP User Part for SRS

You can configure the SIP user part of the Request-URI for the recording server (SRS). The device inserts this user part in the SIP To header of the INVITE message sent to the SRS.

➢ **To configure the SIP user part for SRS:**

1. Open the SIP Recording Settings page (Setup menu > Signaling & Media tab > SIP Recording folder > SIP Recording Settings).

   ![Figure 17-9: Configuring User Part of To Header for SRS](image)

2. In the 'Recording Server (SRS) Destination Username' field, enter a user part value (string of up to 50 characters).
3. Click **Apply**.

#### 17.2.4 Interworking SIP-based Media Recording with Third-Party Vendors

The device can interwork the SIP-based Media Recording feature with third-party vendors, as described in the following subsections.
17.2.4.1 Genesys

The device's SIP-based media recording can interwork with Genesys' equipment. Genesys sends its proprietary X-Genesys-CallUUID header (which identifies the session) in the first SIP message, typically in the INVITE and the first 18x response. If the device receives a SIP message with Genesys SIP header, it adds the header's information to AudioCodes' proprietary tag in the XML metadata of the SIP INVITE that it sends to the recording server, as shown below:

```xml
<ac:GenesysUUID xmlns="urn:ietf:params:xml:ns:Genesys">4BOKLLA3VH66JF112M1CC9VHKS14F0KP</ac:GenesysUUID>
```

No configuration is required for this support.

17.2.4.2 Avaya UCID

The device's SIP-based media recording can interwork with Avaya equipment. The Universal Call Identifier (UCID) is Avaya's proprietary call identifier used to correlate call records between different systems and identifies sessions. Avaya generates this in outgoing calls. If the device receives a SIP INVITE from Avaya, it adds the UCID value, received in the User-to-User SIP header to AudioCodes' proprietary tag in the XML metadata of the SIP INVITE that it sends to the recording server. For example, if the received SIP header is:

```plaintext
User-to-User: 00FA080019001038F725B3;encoding=hex
```

the device includes the following in the XML metadata:

```xml
```

**Note:** For calls sent from the device to Avaya equipment, the device can generate the Avaya UCID, if required. To configure this support, use the following parameters:

- 'UUI Format' in the IP Groups table - enables Avaya support.
- 'Network Node ID' - defines the Network Node Identifier of the device for Avaya UCID.

17.3 RADIUS-based Services

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

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

17.3.1 Enabling RADIUS Services

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

1. Open the Authentication Server page (Setup menu > Administration tab > Web & CLI folder > Authentication Server).

2. Under the RADIUS group, from the ‘Enable RADIUS Access Control’ drop-down list, select Enable.

3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

17.3.2 Configuring RADIUS Servers

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

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

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

- Only one RADIUS server is configured and used for authorization and accounting purposes (no redundancy). Therefore, both the Authorization and Accounting ports are defined.

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

- Two RADIUS servers are configured and used for authorization and accounting purposes, providing redundancy. Therefore, both the Authorization and Accounting ports are defined.

The status of the RADIUS servers can be viewed through CLI:

```
# show system radius servers status
```
The example below shows the status of two RADIUS servers in redundancy mode for authorization and accounting:

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

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

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

**Note:**
- To enable and configure RADIUS-based accounting, see "Configuring RADIUS Accounting" on page 777.
- The device can send up to 201 concurrent RADIUS requests per RADIUS service type (Accounting or Authentication), per RADIUS server (up to three servers per service type), and per local port (up to 4 local ports). For example, 801 (201 * 4) concurrent RADIUS requests can be sent for Authentication and 801 (201 * 4) for Accounting.

**To configure a RADIUS server:**

1. Open the RADIUS Servers table (Setup menu > IP Network tab > RADIUS & LDAP folder > RADIUS Servers).
2. Click New; the following dialog box appears:

   ![Figure 17-11: RADIUS Servers Table - Add Dialog Box](image)

   **Figure 17-11: RADIUS Servers Table - Add Dialog Box**
3. Configure a RADIUS server according to the parameters described in the table below.

4. Click **Apply**.

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

### 17.3.3 Configuring Interface for RADIUS Communication

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

**Note:** If you configure the parameter to Control, make sure that only one Control interface is configured in the IP Interfaces table (see "Configuring IP Network Interfaces" on page 137); otherwise, RADIUS communication fails.

### 17.3.4 Configuring RADIUS Packet Retransmission

You can configure the device to resend packets to the RADIUS server if no response is received from the server. This functionality is applicable to RADIUS-based user authentication and RADIUS-based accounting.
To configure RADIUS packet retransmission:

1. Open the Authentication Server page (Setup menu > Administration tab > Web & CLI folder > Authentication Server).

   Figure 17-12: Configuring RADIUS Packet Retransmission

   | RADIUS Response Timeout [sec] | 2 |
   | RADIUS Packets Retransmission | 1 |

2. Under the RADIUS group, do the following:
   a. In the 'RADIUS Packets Retransmission' field (RADIUSRetransmission), enter the maximum number of RADIUS retransmissions that the device performs if no response is received from the RADIUS server.
   b. In the 'RADIUS Response Time Out' field (RadiusTO), enter the interval (in seconds) that the device waits for a response before sending a RADIUS retransmission.

3. Click Apply.

17.3.5 Configuring the RADIUS Vendor ID

The vendor-specific attribute (VSA) identifies the device to the RADIUS server using the Vendor ID (as registered with the Internet Assigned Numbers Authority or IANA). The device's default vendor ID is 5003 which can be changed, as described in the following procedure. For an example of using the Vendor ID, see "Setting Up a Third-Party RADIUS Server" on page 232. The procedure is applicable to both RADIUS-based user authentication and RADIUS-based accounting.

   Note: The Vendor ID must be the same as the Vendor ID set on the third-party RADIUS server. See the example for setting up a third-party RADIUS server in "Setting Up a Third-Party RADIUS Server" on page 232.

To configure the RADIUS Vendor ID:

1. Open the Authentication Server page (Setup menu > Administration tab > Web & CLI folder > Authentication Server).

   Figure 17-13: Configuring RADIUS Vendor ID

   | RADIUS VSA Vendor ID | 5003 |

2. Under the RADIUS group, in the 'RADIUS VSA Vendor ID' field, enter the same vendor ID number as set on the third-party RADIUS server.

3. Click Apply.

17.3.6 RADIUS-based Management User Authentication

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

When RADIUS authentication is not used, the user's login username and password are locally authenticated by the device using the Local Users table (see "Configuring..."
Management User Accounts" on page 73). However, you can configure the device to use the Local Users table as a fallback mechanism if the RADIUS server does not respond. 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. Communication between the device and the RADIUS server is done using a shared secret, which is not transmitted over the network.

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 232
- Configure the device as a RADIUS client for communication with the RADIUS server - see "Configuring RADIUS Authentication" on page 233

17.3.6.1 Setting Up a Third-Party RADIUS Server

The following procedure provides an example for setting up a third-party RADIUS server, FreeRADIUS which can be downloaded from 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 (configured on the device in "Configuring the RADIUS Vendor ID" on page 231)

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

   ```
   #
   ```
# clients.conf - client configuration directives
#
client 10.31.447 {
    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 Management User Accounts" on page 73.

# 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.

17.3.6.2 Configuring RADIUS-based User Authentication

The following procedure describes how to configure RADIUS-based login authentication. For a detailed description of the RADIUS parameters, see "RADIUS Parameters" on page 940.

➢ To configure RADIUS-based login authentication:

1. Open the Authentication Server page (Setup menu > Administration tab > Web & CLI folder > Authentication Server).

2. From the 'Use RADIUS for Web/Telnet Login' drop-down list, select Enable to enable RADIUS authentication for Web and Telnet login:

   Figure 17-15: Enabling RADIUS-based Login Authentication
3. When implementing Web user access levels, do one of the following:
   - **If the RADIUS server response includes the access level attribute:** In the 'RADIUS VSA Access Level Attribute' field, enter the code that indicates the access level attribute in the VSA section of the received RADIUS packet. For defining the RADIUS server with access levels, see "Setting Up a Third-Party RADIUS Server" on page 232.

   ![Figure 17-16: Authentication Settings Page - RADIUS Configuration](image)

   - **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.

   ![Figure 17-17: Configuring Default Access Level](image)

4. Configure RADIUS timeout handling:
   a. From the 'Behavior upon Authentication Server Timeout' drop-down list, select the option if the RADIUS server does not respond within five seconds:
      - **Deny Access:** device denies user login access.
      - **Verify Access Locally:** device checks the username and password configured locally for the user in the Local Users table (see "Configuring Management User Accounts" on page 73), 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.

   ![Figure 17-18: Configuring RADIUS Timeout](image)

5. Configure when the Local Users table must be used to authenticate login users. From the 'Use Local Users Database' drop-down list, select one of the following:
   - **When No Auth Server Defined (default):** When no RADIUS server is configured or if a server is configured but connectivity with the server is down (if the server is up, the device authenticates the user with the server).
   - **Always:** First attempts to authenticate the user using the Local Users table, but if not found, it authenticates the user with the RADIUS server.

   ![Figure 17-19: Local Users Table for Login Authentication](image)

6. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.
17.3.6.3 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 enable the device to use HTTPS, configure the 'Secured Web Connection (HTTPS)' parameter to HTTPS Only (see "Configuring Secured (HTTPS) Web" on page 80).

17.3.6.4 RADIUS-based User Authentication in URL

RADIUS authentication of the management user is typically done after the user accesses the Web interface by entering only the device's IP address in the Web browser's URL field (for example, http://10.13.4.12/) and then entering the username and password credentials in the Web interface's login screen. However, authentication with the RADIUS server can also be done immediately after the user enters the URL, if the URL also contains the login credentials. For example:

http://10.4.4.112/Forms/RadiusAuthentication?WSBackUserName=John&WSBackPassword=1234

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

17.3.7 RADIUS-based CDR Accounting

Once you have configured a RADIUS server(s) for accounting in "Configuring RADIUS Servers" on page 228, you need to enable and configure RADIUS-based CDR accounting (see "Configuring RADIUS Accounting" on page 777).

17.4 LDAP-based Management and SIP Services

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

You can use LDAP for the following LDAP services:

- **SIP-related (Control) LDAP Queries**: LDAP can be used for routing and 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 243). 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 "AD-based Routing for Microsoft Skype for Business" on page 257.

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 247.

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

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

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

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

The device then assigns the user the access level configured for that group (in "Configuring Access Level per Management Groups Attributes" on page 245). 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 243.

- Search filter, for example, (&(objectClass=person)(sAMAccountName=JohnD)), which filters the search in the subtree to include only the specific username. The search filter can be configured with the dollar ($) sign to represent the username, for example, (sAMAccountName=$). To configure the search filter, see "Configuring the LDAP Search Filter Attribute" on page 244.

- 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 Servers table (see "Configuring LDAP Servers" on page 240).

If the device finds a group, it assigns the user the corresponding access level and permits login; otherwise, login is denied. Once the LDAP response has been received (success or failure), the device ends the LDAP session.
For both of the previously discussed LDAP services, the following additional LDAP functionality is supported:

- Search method for searching DN object records between LDAP servers and within each LDAP server (see Configuring LDAP Search Methods).
- Default access level that is assigned to the user if the queried response does not contain an access level.
- Local Users 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 251.

17.4.1 Enabling the LDAP Service

Before you can configure LDAP support, you need to enable the LDAP service.

➢ To enable LDAP:
  1. Open the LDAP Settings page (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Settings).

      Figure 17-20: Enabling LDAP

      table

      LDAP Service

      Enable

  2. From the 'LDAP Service' drop-down list, select Enable.
  3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

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

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

➢ To enable LDAP-based login authentication:
  1. Open the Authentication Server page (Setup menu > Administration tab > Web & CLI folder > Authentication Server).

      Figure 17-21: Enabling LDAP-based Login Authentication

      table

      LDAP

      Use LDAP for Web/Telnet Login

      Disable

  2. Under the LDAP group, from the 'Use LDAP for Web/Telnet Login' drop-down list, select Enable.
  3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.
17.4.3 Configuring LDAP Server Groups

The LDAP Server Groups table lets you configure up to 600 LDAP Server Groups. An LDAP Server Group is a logical configuration entity that contains up to two LDAP servers. LDAP servers are assigned to LDAP Server Groups in the LDAP Servers table (see "Configuring LDAP Servers" on page 240). To use a configured LDAP server, you must assign it to an LDAP Server Group.

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

To use an LDAP server for management user login authentication and authorization, you must configure its' LDAP Server Group as "Management" type. Additional LDAP-based management parameters need to be configured, as described in "Enabling LDAP-based Web/CLI User Login Authentication and Authorization" on page 237 and "Configuring LDAP Servers" on page 240.

The following procedure describes how to configure an LDAP Server Group through the Web interface. You can also configure it through ini file (LDAPServerGroups) or CLI (configure system > ldap ldap-server-groups).

Note: The device provides a preconfigured LDAP Server Group ("DefaultCTRLServersGroupin") in the LDAP Server Groups table, which can be modified or deleted.

➢ To configure an LDAP Server Group:

1. Open the LDAP Server Groups table (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Server Groups).
2. Click New; the following dialog box appears:

   ![Figure 17-22: LDAP Server Groups Table - Add Dialog Box](image)

3. Configure an LDAP Server Group according to the parameters described in the table below.
4. Click Apply.
### Table 17-8: LDAP Server Groups Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Index                      | Defines an index number for the new table row.  
  Index [LdapServerGroups_Index]  
  **Note:** Each row must be configured with a unique index. |
| Name                      | Defines a descriptive name, which is used when associating the row in other tables.  
  Name [LdapServerGroups_Name]  
  The valid value is a string of up to 20 characters.  
  **Note:** Each row must be configured with a unique name. |
| Type                      | Defines whether the servers in the group are used for SIP-related LDAP queries (Control) or management login authentication-related LDAP queries (Management).  
  Type [LdapServerGroups_ServerType]  
  - [0] Control (Default)  
  - [1] Management  
  **Note:** Only one LDAP Server Group can be defined for management. |
| Server Search Method      | Defines the method for querying between the two LDAP servers in the group.  
  Server Search Method [LdapServerGroups_SearchMethod]  
  - [0] Parallel = (Default) The device queries the LDAP servers at the same time.  
  - [1] Sequential = The device first queries one of the LDAP servers and if the DN object is not found or the search fails, it queries the second LDAP server. |
| DN Search Method          | Defines the method for querying the Distinguished Name (DN) objects within each LDAP server.  
  DN Search Method [LdapServerGroups_SearchDnsMethod]  
  - [0] Sequential = (Default) The query is done in each DN object, one by one, until a result is returned. For example, a search for the DN object record "JohnD" is first run in DN object "Marketing" and if a result is not found, it searches in "Sales", and if not found, it searches in "Administration", and so on.  
  - [1] Parallel = The query is done in all DN objects at the same time. For example, a search for the DN object record "JohnD" is done at the same time in the "Marketing", "Sales" and "Administration" DN objects. |
| **Cache**                 |             |
| Cache Entry Timeout       | Defines the duration (in minutes) that an entry in the device's LDAP cache is valid. If the timeout expires, the cached entry is used only if there is no connectivity with the LDAP server.  
  Cache Entry Timeout [LdapServerGroups_CacheEntryTimeout]  
  The valid range is 0 to 35791. The default is 1200. If set to 0, the LDAP entry is always valid. |
| Cache Entry Removal Timeout| Defines the duration (in hours) after which the LDAP entry is deleted from the device's LDAP cache.  
  Cache Entry Removal Timeout [LdapServerGroups_CacheEntryRemovalTimeout]  
  The valid range is 0 to 596. The default is 0 (i.e., the entry is never deleted). |
17.4.4 Configuring LDAP Servers

The LDAP Servers table lets you configure up to 1,200 LDAP servers. The table defines the address and connectivity settings of the LDAP server. The LDAP server can be configured for SIP-related queries (e.g., routing and manipulation) or LDAP-based management user login authentication and authorization (username-password).

The following procedure describes how to configure an LDAP server through the Web interface. You can also configure it through ini file (LdapConfiguration) or CLI (configure system > ldap ldap-configuration).

Note: When you configure an LDAP server, you need to assign it an LDAP Server Group. Therefore, before you can configure an LDAP server in the table, you must first configure at least one LDAP Server Group in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 238).

➢ To configure an LDAP server:

1. Open the LDAP Servers table (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Servers).
2. Click New; the following dialog box appears:

   Figure 17-23: LDAP Servers Table - Add Dialog Box

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

Table 17-9: LDAP Servers Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[LdapConfiguration_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>LDAP Servers Group</td>
<td>Assigns the LDAP server to an LDAP Server Group, configured in</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>server-group [LdapConfiguration_Group]</td>
<td>the LDAP Server Groups table (see &quot;Configuring LDAP Server Groups&quot; on page 238).</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• The parameter is mandatory and must be set before configuring the other parameters in the table.</td>
</tr>
<tr>
<td></td>
<td>• Up to two LDAP servers can be assigned to the same LDAP Server Group.</td>
</tr>
<tr>
<td>LDAP Network Interface</td>
<td>Assigns one of the device's IP network interfaces through which communication with the LDAP server is done.</td>
</tr>
<tr>
<td>interface-type [LdapConfiguration_Interface]</td>
<td>By default, no value is defined and the device uses the OAMP network interface, configured in the IP Interfaces table.</td>
</tr>
<tr>
<td></td>
<td>To configure IP network interfaces, see &quot;Configuring IP Network Interfaces&quot; on page 137.</td>
</tr>
<tr>
<td>Use TLS use-tls [LdapConfiguration_useTLS]</td>
<td>Enables the device to encrypt the username and password (for Control and Management related queries) using TLS when sending them to the LDAP server.</td>
</tr>
<tr>
<td></td>
<td>• [0] No = (Default) Username and password are sent in clear-text format.</td>
</tr>
<tr>
<td></td>
<td>• [1] Yes</td>
</tr>
<tr>
<td>TLS Context tls-context</td>
<td>Assigns a TLS Context for the connection with the LDAP server.</td>
</tr>
<tr>
<td>[LdapConfiguration_ContextName]</td>
<td>By default, no value is defined and the device uses the default TLS Context (ID 0).</td>
</tr>
<tr>
<td></td>
<td>To configure TLS Contexts, see &quot;Configuring TLS Certificate Contexts&quot; on page 105.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The parameter is applicable only if the 'Use TLS' parameter is configured to Yes.</td>
</tr>
<tr>
<td>Connection</td>
<td></td>
</tr>
<tr>
<td></td>
<td>By default, no IP address is defined.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• The parameter is mandatory.</td>
</tr>
<tr>
<td></td>
<td>• If you want to use an FQDN for the LDAP server, leave the parameter undefined and configure the FQDN in the 'LDAP Server Domain Name' parameter (see below).</td>
</tr>
<tr>
<td>LDAP Server Port server-port [LdapConfiguration_LdapConfServerPort]</td>
<td>Defines the port number of the LDAP server.</td>
</tr>
<tr>
<td></td>
<td>The valid value range is 0 to 65535. The default port number is 389.</td>
</tr>
<tr>
<td>LDAP Server Max Respond Time max-respond-time [LdapConfiguration_LdapConfServerMaxRespondTime]</td>
<td>Defines the duration (in msec) that the device waits for LDAP server responses.</td>
</tr>
<tr>
<td></td>
<td>The valid value range is 0 to 86400. The default is 3000.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If the response time expires, you can configure the device to use the Local Users table for authenticating the user. For more information, see &quot;Configuring Local Database for Management&quot;.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>User Authentication</td>
<td>“User Authentication” on page 251.</td>
</tr>
<tr>
<td>LDAP Server Domain Name</td>
<td>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. <strong>Note:</strong> If the ‘LDAP Server IP’ parameter is configured, the ‘LDAP Server Domain Name’ parameter is ignored. Thus, if you want to use an FQDN, leave the ‘LDAP Server IP’ parameter undefined.</td>
</tr>
<tr>
<td>Verify Certificate</td>
<td>Enables certificate verification when the connection with the LDAP server uses TLS.</td>
</tr>
<tr>
<td>Connection Status</td>
<td>(Read-only) Displays the connection status with the LDAP server.</td>
</tr>
<tr>
<td>Query</td>
<td></td>
</tr>
<tr>
<td>LDAP Password</td>
<td>Defines the user password for accessing the LDAP server during connection and binding operations.</td>
</tr>
</tbody>
</table>

**Note:**
- The parameter is mandatory.
### Parameter | Description
--- | ---
- By default, the device sends the password in clear-text format. You can enable the device to encrypt the password using TLS (see the 'Use SSL' parameter below).

**LDAP Bind DN**

**bind-dn**

[LdapConfiguration_LdapConfBindDn]

- Defines the LDAP server's bind Distinguished Name (DN) or username.
- LDAP-based SIP queries: The DN is used as the username during connection and binding to the LDAP server. The DN is used to uniquely name an AD object. Below are example parameter settings:
  - cn=administrator,cn=Users,dc=domain,dc=com
  - administrator@domain.com
  - domain\administrator
- LDAP-based user login authentication: The parameter represents the login username entered by the user during a login attempt. You can use the $ (dollar) sign in this value to enable the device to automatically replace the $ sign with the user's login username in the search filter, which it sends to the LDAP server for authenticating the user's username-password combination. An example configuration for the parameter is $@sales.local, where the device replaces the $ with the entered username, for example, JohnD@sales.local. The username can also be configured with the domain name of the LDAP server.

**Note:** By default, the device sends the username in clear-text format. You can enable the device to encrypt the username using TLS (see the 'Use SSL' parameter below).

**Management Attribute**

**mgmt-attr**

[LdapConfiguration_MngmAuthAttr]

- 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 245.

**Note:**
- The parameter is applicable only to LDAP-based login authentication and authorization (i.e., the 'Type' parameter is set to Management).
- If this functionality is not used, the device assigns the user the configured default access level. For more information, see "Configuring Access Level per Management Groups Attributes" on page 245.

### 17.4.5 Configuring LDAP DNs (Base Paths) per LDAP Server

The LDAP Search DN table lets you configure LDAP base paths. The table is a "child" of the LDAP Servers table (see "Configuring LDAP Servers" on page 240) 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 through the Web interface. You can also configure it through ini file (LdapServersSearchDNs) or CLI (configure system > ldap ldap-servers-search-dns).

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

1. Open the LDAP Servers table (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Servers).
2. In the table, select the row of the LDAP server for which you want to configure DN base paths, and then click the LDAP Servers Search Based DNs link located below the table; the LDAP Server Search Base DN table opens.
3. Click New; the following dialog box appears:

   ![LDAP Search Base DN Table - Add Dialog Box](image)

   Figure 17-24: LDAP Search Base DN Table - Add Dialog Box

4. Configure an LDAP DN base path according to the parameters described in the table below.
5. Click Apply, and then save your settings to flash memory.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index set internal-index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[LdapServersSearchDNs_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Base DN set base-path</td>
<td>Defines the full path (DN) to the objects in the AD where the query is done.</td>
</tr>
<tr>
<td>[LdapServersSearchDNs_Base_Path]</td>
<td>The valid value is a string of up to 256 characters.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
</tbody>
</table>

### 17.4.6 Configuring the LDAP Search Filter Attribute

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

- **Search base object (distinguished name or DN, e.g., "ou=ABC,dc=corp,dc=abc,dc=com"):** The DN defines the location in the directory from which the LDAP search begins and is configured in "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 243.

- **Filter (e.g., 
  "(&objectClass=person)(sAMAccountName=johnd)"**): This filters the search in the subtree to include only the login username (and excludes others). This is
configured by the 'LDAP Authentication Filter' parameter, as described in the following procedure. You can use the dollar ($) sign to represent the username. For example, the filter can be configured as "(sAMAccountName=$)", where if the user attempts to log in with the username "SueM", the LDAP search is done only for the attribute sAMAccountName that equals "SueM".

- **Attribute (e.g., "memberOf") to return from objects that match the filter criteria:** The attribute is configured by the 'Management Attribute' parameter in the LDAP Servers table (see "Configuring LDAP Servers" on page 240).

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

**Note:**
- 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 (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Settings).
2. In the 'LDAP Authentication Filter' field, enter the LDAP search filter attribute for searching the login username for user authentication:

   ![Figure 17-25: Configuring LDAP Search Filter](image)

   **LDAP Authentication Filter**

3. Click Apply.

**17.4.7 Configuring Access Level per Management Groups Attributes**

The Management LDAP Groups table lets you configure LDAP group objects and their corresponding management user access level. The table is a "child" of the LDAP Servers table (see "Configuring LDAP Servers" on page 240) 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.
Note:

- 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 as used also for RADIUS (see "Configuring RADIUS-based User Authentication" on page 233). 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 240).

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 Management User Accounts" on page 73.

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 243.
- Filter (e.g., "(&(objectClass=person)(sAMAccountName=johnd))"), which filters the search in the subtree to include only the login username (and excludes others). For configuration, see "Configuring the LDAP Search Filter Attribute" on page 244.
- 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 Servers table.

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

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

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

The following procedure describes how to configure an access level per management groups through the Web interface. You can also configure it through ini file (MgmntLDAPGroups) or CLI (configure system > ldap mgmt-ldap-groups).

To configure management groups and corresponding access level:

1. Open the LDAP Servers table (Setup menu > IP Network tab > RADIUS & LDAP
folder > LDAP Servers).

2. In the table, select the row of the LDAP server for which you want to configure management groups with a corresponding access level, and then click the Management LDAP Groups link located below the table; the Management LDAP Groups table opens.

3. Click New; the following dialog box appears:

   **Figure 17-26: Management LDAP Groups Table - Add Dialog Box**

4. Configure a group name(s) with a corresponding access level according to the parameters described in the table below.

5. Click Apply, and then save your settings to flash memory.

   **Table 17-11: Management LDAP Groups Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[MgmntLDAPGroups_GroupIndex]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Level</td>
<td>Defines the access level of the group(s).</td>
</tr>
<tr>
<td>[MgmntLDAPGroups_Level]</td>
<td>• [0] Monitor (Default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Admin</td>
</tr>
<tr>
<td></td>
<td>• [2] Security Admin</td>
</tr>
<tr>
<td>Groups</td>
<td>Defines the attribute names of the groups in the LDAP server. Valid value</td>
</tr>
<tr>
<td>[MgmntLDAPGroups_Group]</td>
<td>is a string of up to 256 characters. To define multiple groups, separate</td>
</tr>
<tr>
<td></td>
<td>each group name with a semicolon (;).</td>
</tr>
</tbody>
</table>

### 17.4.8 Configuring the Device’s LDAP Cache

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

- Improves routing decision performance by using local cache for subsequent LDAP queries
- Reduces number of queries performed on an LDAP server and corresponding bandwidth consumption
- Provides partial survivability in case of intermittent LDAP server failure (or network isolation)
The handling of LDAP queries using the device's LDAP cache is shown in the flowchart below:

**Figure 17-27: LDAP Query Process with Local LDAP Cache**

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

**Table 17-12: Example of LDAP Query for Cached Attributes**

<table>
<thead>
<tr>
<th>Attributes Requested in New LDAP Query for Cached Key</th>
<th>Attributes Sent in LDAP Query to LDAP Server</th>
<th>Attributes Saved in Cache after LDAP Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>e, a, b, c, d</td>
<td>e, a, b, c, d</td>
</tr>
<tr>
<td>e, f</td>
<td>e, f, a, b, c, d</td>
<td>e, f, a, b, c, d</td>
</tr>
<tr>
<td>e, f, g, h, i</td>
<td>e, f, g, h, i, a</td>
<td>e, f, g, h, i, a</td>
</tr>
<tr>
<td>e, f, g, h, i, j</td>
<td>e, f, g, h, i, j</td>
<td>e, f, g, h, i, j</td>
</tr>
</tbody>
</table>
Note:
- The LDAP Cache feature is applicable only to LDAP-based SIP queries (Control).
- The maximum LDAP cache size is 20,000 entries.
- The device can save up to six LDAP Attributes in the cache per user (search LDAP key).
- The device also saves in the cache queried Attributes that do not have any values in the LDAP server.

The following procedure describes how to configure the device's LDAP cache through the Web interface. For a full description of the cache parameters, see "LDAP Parameters" on page 942.

➢ To enable and configure the LDAP cache:

1. Open the LDAP Settings page (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Settings).

   ![Figure 17-28: Enabling LDAP Cache](image)

   - LDAP Cache Service
   - LDAP Cache Entry Timeout
   - LDAP Cache Entry Removal Timeout

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

3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

17.4.8.1 Refreshing the LDAP Cache

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

**Figure 17-29: LDAP Cache Refresh Flowchart**

To refresh the LDAP cache per LDAP Server Group:

1. Open the LDAP Settings page (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Settings).

2. Under the Cache Actions group, do the following:
   a. From the 'LDAP Group Index' drop-down list, select the required LDAP Server Group (see "Configuring LDAP Server Groups" on page 238).
   b. In the 'LDAP Refresh Cache by Key' field, enter the LDAP search key that you want to refresh (e.g., telephoneNumber=1004).
   c. Click **Refresh**; if a request with the specified key exists in the cache, a request is sent to the LDAP server for the Attributes associated in the cache with the search key.
17.4.8.2 Clearing the LDAP Cache

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

➢ To clear the LDAP cache:
  1. Open the LDAP Settings page (Setup menu > IP Network tab > RADIUS & LDAP folder > LDAP Settings).
  2. Under the Cache Actions group, do the following:

         a. From the 'LDAP Group Index' drop-down list, select the required LDAP Server Group (see "Configuring LDAP Server Groups" on page 238).
         b. Click Clear Group.

17.4.9 Configuring Local Database for Management User Authentication

You can configure the device to use the Local Users table (local database) to authenticate management users based on username-password combination. You can configure the device to use the Local Users table (see "Configuring Management User Accounts" on page 73) upon the following scenarios:

- LDAP or RADIUS server is not configured (or broken connection) or always use the Local 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 Local Users table.

If user authentication using the Local 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 Local Users table.

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

➢ To use the Local Users table for authenticating management users:
  1. Open the Authentication Server page (Setup menu > Administration tab > Web & CLI folder > Authentication Server).

         Figure 17-31: Local Users Table for Login Authentication

         Use Local Users Database
         Behavior upon Authentication Server Timeout

  2. Under the General group, do the following:

         a. Configure when the Local Users table must be used to authenticate login users. From the 'Use Local Users Database' drop-down list, select one of the following:

             ♦ When No Auth Server Defined (default): When no LDAP/RADIUS server is configured or if a server is configured but connectivity with the server is down (if the server is up, the device authenticates the user with the server).
             ♦ Always: First attempts to authenticate the user using the Local Users table,
but if not found, it authenticates the user with the LDAP/RADIUS server.

b. Configure whether the Local 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 Local Users table.

3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

17.4.10 LDAP-based Login Authentication Example

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

The LDAP server's entry data structure schema in the example is as follows:
DN (base path): OU=testMgmt,OU=QA,DC=testqa,DC=local. The DN path to search for the username in the directory is shown below:

Figure 17-32: Base Path (DN) in LDAP Server
**Search Attribute Filter**: (sAMAccountName=$). The login username is found based on this attribute (where the attribute’s value equals the username):

*Figure 17-33: Username Found using sAMAccount Attribute Search Filter*

![Active Directory Explorer - Sysinternals: www.sysinternals.com][10.3.9.93 [testpq.testqa.local]]

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

*Figure 17-34: User's memberOf Attribute*

![Active Directory Explorer - Sysinternals: www.sysinternals.com][10.3.9.93 [testpq.testqa.local]]
**Management Group:** mySecAdmin. The group to which the user belongs, as listed under the memberOf attribute:

![Figure 17-35: User’s mySecAdmin Group in memberOf Management Attribute](image)

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

- LDAP-based login authentication (management) is enabled in the LDAP Server Groups table (see "Configuring LDAP Server Groups" on page 238):

![Figure 17-36: Configuring LDAP Server Group for Management](image)

- The DN is configured in the LDAP Server Search Base DN table (see "Configuring LDAP DNs (Base Paths) per LDAP Server" on page 243):

![Figure 17-37: Configuring DN](image)
The search attribute filter based on username is configured by the 'LDAP Authentication Filter' parameter (see "Configuring the LDAP Search Filter Attribute" on page 244):

**Figure 17-38: Configuring Search Attribute Filter**

![Configuration Panel](image)

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

**Figure 17-39: Configuring Management Attribute**

![Server Configuration Panel](image)

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 245):

**Figure 17-40: Configuring Management Group Attributes for Determining Access Level**

![Group Configuration Panel](image)
17.4.11 Enabling LDAP Searches for Numbers with Characters

Typically, the device performs LDAP searches in the AD for complete numbers where the digits are adjacent to one another (e.g., 5038234567). However, if the number is defined in the AD with characters (such as spaces, hyphens and periods) separating the digits (e.g., 503-823 4567), the LDAP query returns a failed result.

To enable the device to search the AD for numbers that may contain characters between its digits, you need to specify the Attribute (up to five) for which you want to apply this functionality, using the LDAPNumericAttributes parameter. For example, the telephoneNumber Attribute could be defined in AD with the telephone number "503-823-4567" (i.e., hyphens), "503.823.4567" (i.e., periods) or "503 823 4567" (i.e., spaces). If the device performs an LDAP search on this Attribute for the number 5038234567, the LDAP query will return results only if you configure the LDAPNumericAttributes parameter with the telephoneNumber Attribute (e.g., LDAPNumericAttributes= telephoneNumber). To search for the number with characters, the device inserts the asterisk (*) wildcard between all digits in the LDAP query (e.g., telephoneNumber = 5*0*3*8*2*3*4*5*6*7). As the AD server recognizes the * wildcard as representing any character, it returns all possible results to the device. Note that the wildcard represents only a character; a query result containing a digit in place of a wildcard is discarded and the device performs another query for the same Attribute. For example, it may return the numbers 533 -823-4567 (second digit "3" and hyphens) and 503- 823-4567. As the device discards query results where the wildcard results in a digit, it selects 503 -823-4567 as the result. The correct query result is cached by the device for subsequent queries and/or in case of LDAP server failure.

17.4.12 AD-based Routing for Microsoft Skype for Business

Typically, enterprises wishing to deploy the Microsoft® Skype for Business (formerly known as Lync) are faced with a complex, call routing dial plan when migrating users from their existing PBX or IP PBX to the Skype for Business 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:

- Skype for Business client - users connected to Skype for Business through the Mediation Server
- PBX or IP PBX - users not yet migrated to Skype for Business
- Mobile - mobile number
- Private - private telephone line for Skype for Business users (in addition to the primary telephone line)
17.4.12.1 Querying the AD and Routing Priority

The device queries the AD using the initial destination number (i.e., called number). The query can return up to four user phone numbers, each pertaining to one of the IP domains (i.e., private number, Skype for Business 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 17-13: Parameters for Configuring Query Attribute Key

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Queried User Domain (Attribute) in AD</th>
<th>Query or Query Result Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSLDAPPBXNumAttributeName</td>
<td>PBX or IP PBX number (e.g., &quot;telephoneNumber&quot; - default)</td>
<td>telephoneNumber= +3233554447</td>
</tr>
<tr>
<td>MSLDAPOCSNumAttributeName</td>
<td>Mediation Server / Skype for Business client number (e.g., &quot;msRTCSIP-Line&quot;)</td>
<td>msRTCSIP-Line=<a href="mailto:john.smith@company.com">john.smith@company.com</a></td>
</tr>
<tr>
<td>MSLDAPMobileNumAttributeName</td>
<td>Mobile number (e.g., &quot;mobile&quot;)</td>
<td>mobile=+3247647156</td>
</tr>
<tr>
<td>MSLDAPPPrivateNumAttributeName</td>
<td>Any attribute (e.g., &quot;msRTCSIP-PrivateLine&quot;) Note: Used only if set to same value as Primary or Secondary key.</td>
<td>msRTCSIP-PrivateLine= +3233554480</td>
</tr>
<tr>
<td>MSLDAPPrimaryKey</td>
<td>Primary Key query search instead of PBX key - can be any AD attribute</td>
<td>msRTCSIP-PrivateLine= +3233554480</td>
</tr>
<tr>
<td>MSLDAPSecondaryKey</td>
<td>Secondary Key query key search if Primary Key fails - can be any attribute</td>
<td>-</td>
</tr>
</tbody>
</table>

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 - Skype for Business, PBX / IP PBX, private (only if it equals Primary or Secondary key), and mobile.

6. The device adds unique prefix keywords to the query results in order to identify the query type (i.e., IP domain). These prefixes are used as the prefix destination number value in the Tel-to-IP Routing table to denote the IP domains:
   - "PRIVATE" (PRIVATE:<private_number>): used to match a routing rule based on query results of the private number (MSLDAPPrivateNumAttributeName)
   - "OCS" (OCS:<Skype for Business_number>): used to match a routing rule based on query results of the Skype for Business client number (MSLDAPOCSNumAttributeName)
   - "PBX" (PBX:<PBX_number>): used to match a routing rule based on query results of the PBX / IP PBX number (MSLDAPPBXNumAttributeName)
   - "MOBILE" (MOBILE:<mobile_number>): used to match a routing rule based on query results of the mobile number (MSLDAPMobileNumAttributeName)
   - "LDAP_ERR": used to match a routing rule based on a failed query result when no attribute is found in the AD

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

7. The device uses the Tel-to-IP Routing table to route the call based on the LDAP query result. The device routes the call according to the following priority:
   1. **Private line**: If the query is done for the private attribute and it's found, the device routes the call according to this attribute.
   2. **Mediation Server SIP address (Skype for Business)**: 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 Skype for Business client).
   3. **PBX / IP PBX**: If the Skype for Business client is not found in the AD, it routes the call to the PBX / IP PBX.
   4. **Mobile number**: If the Skype for Business client (or Mediation Server) is unavailable (e.g., SIP response 404 "Not Found" upon INVITE sent to Skype for Business client), and the PBX / IP PBX is also unavailable, the device routes the call to the user’s mobile number (if exists in the AD).
   5. **Alternative route**: If the call routing to all the above fails (e.g., due to unavailable destination - call busy), the device can route the call to an alternative destination if an alternative routing rule is configured.
   6. **"Redundant" route**: If the query failed (i.e., no attribute found in the AD), the device uses the routing rule matching the "LDAP_ERR" prefix destination number value.
The flowchart below summarizes the device's process for querying the AD and routing the call based on the query results:

**Figure 17-41: Querying AD in Skype for Business Environment**

- Queries AD using PBX Attribute Name if Primary Key not configured, otherwise, using Primary Key.
- Secondary Key Configured?
  - Yes: Routes Call to Original Dialed Number using LDAP_ERR Routing Rule.
  - No: Routes Call to Alternative Routing Rule, if Configured.

**Query Results**
Retrieves only Required Attributes (Skype for Business / 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:skype_number, PBX:pbx_number, MOBILE:mobile_number)

**Routes Call According to Priority**
1. Private if available
2. Mediation Server (Skype for Business / OCS) if available
3. PBX if Skype for Business / OCS does not exist
4. Mobile if Skype for Business / OCS or PBX were tried

**Call Failure**
(e.g., Busy Destination)

**Note:** If you are using the device's local LDAP cache, see "Configuring the Device's LDAP Cache" on page 247 for the LDAP query process.
17.4.12.2 Configuring AD-Based Routing Rules

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

➢ To configure LDAP-based IP routing for Skype for Business:

1. Configure the LDAP server parameters, as described in "Configuring LDAP Servers" on page 240.

2. Configure the AD attribute names used in the LDAP query:
   a. Open the SIP Definitions General Settings page (Setup menu > Signaling & Media tab > SIP Definitions folder > SIP Definitions General Settings).

   Figure 17-42: LDAP Parameters for Microsoft Skype for Business

<table>
<thead>
<tr>
<th>LDAP Numeric Attributes</th>
<th>LDAP OCS Number Attribute Name</th>
<th>msRTC-SIP-Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>MS LDAP PBX Number Attribute Name</td>
<td>telephoneNumber</td>
<td></td>
</tr>
<tr>
<td>LDAP MOBILE Number Attribute Name</td>
<td>mobile</td>
<td></td>
</tr>
<tr>
<td>LDAP DISPLAY Name Attribute Name</td>
<td>displayName</td>
<td></td>
</tr>
<tr>
<td>LDAP PRIVATE Number Attribute Name</td>
<td>msRTC-SIP-PrivateLine</td>
<td></td>
</tr>
<tr>
<td>LDAP Primary Key</td>
<td>telephoneNumber</td>
<td></td>
</tr>
<tr>
<td>LDAP Secondary Key</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   b. Configure the LDAP attribute names as desired.

3. Configure query-result routing rules for each IP domain (private, PBX / IP PBX, Skype for Business clients, and mobile), using the LDAP keywords (case-sensitive) for the prefix destination number:

4. Configure AD-based IP-to-IP routing rules:
   a. Open the IP-to-IP Routing table (see Configuring SBC IP-to-IP Routing Rules on page 501).

   b. Configure query-result routing rules for each IP domain (private, PBX / IP PBX, Skype for Business clients, and mobile), using the LDAP keywords (case-sensitive) in the Destination Username Prefix field:
      ♦ PRIVATE: Private number
      ♦ OCS: Skype for Business client number
      ♦ PBX: PBX / IP PBX number
      ♦ MOBILE: Mobile number
      ♦ LDAP_ERR: LDAP query failure

   c. Configure a routing rule for routing the initial call (LDAP query) to the LDAP server, by setting the 'Destination Type' field to LDAP for denoting the IP address of the LDAP server.

   d. For alternative routing, enable the alternative routing mechanism and configure corresponding SIP reasons for alternative routing. For this feature, alternative routing starts from the table row located under the LDAP query row.
The table below shows an example for configuring AD-based SBC routing rules in the IP-to-IP Routing Table:

**Table 17-14: AD-Based SBC IP-to-IP Routing Rule Configuration Examples**

<table>
<thead>
<tr>
<th>Index</th>
<th>Destination Username Prefix</th>
<th>Destination Type</th>
<th>Destination Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PRIVATE:</td>
<td>Dest Address</td>
<td>10.33.45.60</td>
</tr>
<tr>
<td>2</td>
<td>PBX:</td>
<td>Dest Address</td>
<td>10.33.45.65</td>
</tr>
<tr>
<td>3</td>
<td>OCS:</td>
<td>Dest Address</td>
<td>10.33.45.68</td>
</tr>
<tr>
<td>4</td>
<td>MOBILE:</td>
<td>Dest Address</td>
<td>10.33.45.100</td>
</tr>
<tr>
<td>5</td>
<td>LDAP_ERR</td>
<td>Dest Address</td>
<td>10.33.45.80</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
<td>LDAP</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>*</td>
<td>Dest Address</td>
<td>10.33.45.72</td>
</tr>
</tbody>
</table>

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 Skype for Business client (i.e., Mediation Server at 10.33.45.68) upon successful AD query result for the Skype for Business 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, Skype for Business, PBX, and mobile), and a relevant SBC Alternative Routing Reason (see Configuring SIP Response Codes for Alternative Routing Reasons on page 519) 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.
17.5 Least Cost Routing

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

17.5.1 Overview

The LCR feature enables the device to choose the outbound IP destination routing rule based on lowest call cost. This is useful in that it enables service providers to optimize routing costs for customers. For example, you may wish to define different call costs for local and international calls or different call costs for weekends and weekdays (specifying even the time of call). The device sends the calculated cost of the call to a Syslog server (as Information messages), thereby enabling billing by third-party vendors.

LCR is implemented by defining Cost Groups and assigning them to routing rules in the IP-to-IP Routing table. The device searches the routing table for matching routing rules and then selects the rule with the lowest call cost. If two routing rules have identical costs, the rule appearing higher up in the table is used (i.e., first-matched rule). If the selected route is unavailable, the device selects the next least-cost routing rule.

Even if a matched routing rule is not assigned a Cost Group, the device can select it as the preferred route over other matched rules that are assigned Cost Groups. This is determined according to the settings of the 'Default Call Cost' parameter configured for the Routing Policy (associated with the routing rule). To configure the Routing Policy, see Configuring SBC Routing Policy Rules on page 521.

The Cost Group defines a fixed connection cost (connection cost) and a charge per minute (minute cost). Cost Groups can also be configured with time segments (time bands), which define connection cost and minute cost based on specific days of the week and time of day (e.g., from Saturday through Sunday, between 6:00 and 18:00). If multiple time bands are configured per Cost Group and a call spans multiple time bands, the call cost is calculated using only the time band in which the call was initially established.

In addition to Cost Groups, the device can calculate the call cost using an optional, user-defined average call duration value. The logic in using this option is that a Cost Group may be cheap if the call duration is short, but due to its high minute cost, may prove very expensive if the duration is lengthy. Thus, together with Cost Groups, the device can use this option to determine least cost routing. The device calculates the Cost Group call cost as follows:

\[ \text{Total Call Cost} = \text{Connection Cost} + (\text{Minute Cost} \times \text{Average Call Duration}) \]

The below table shows an example of call cost when taking into consideration call duration. This example shows four defined Cost Groups and the total call cost if the average call duration is 10 minutes:

<table>
<thead>
<tr>
<th>Cost Group</th>
<th>Connection Cost</th>
<th>Minute Cost</th>
<th>1 Minute</th>
<th>10 Minutes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>6</td>
<td>7</td>
<td>61</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>10</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>C</td>
<td>0.3</td>
<td>8</td>
<td>8.3</td>
<td>80.3</td>
</tr>
<tr>
<td>D</td>
<td>6</td>
<td>1</td>
<td>7</td>
<td>16</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Cost Group</th>
<th>Connection Cost</th>
<th>Minute Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. &quot;Local Calls&quot;</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2. &quot;International Calls&quot;</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

The Cost Groups are assigned to routing rules for local and international calls:

<table>
<thead>
<tr>
<th>Routing Index</th>
<th>Dest Phone Prefix</th>
<th>Destination IP</th>
<th>Cost Group ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2000</td>
<td>x.x.x.x</td>
<td>1 &quot;Local Calls&quot;</td>
</tr>
<tr>
<td>2</td>
<td>00</td>
<td>x.x.x.x</td>
<td>2 &quot;International Calls&quot;</td>
</tr>
</tbody>
</table>

**Example 2:** This example shows how the device determines the cheapest routing rule in the Tel-to-IP Routing table:

The 'Default Call Cost' parameter in the Routing Policy rule is configured to **Lowest Cost**, meaning that if the device locates other matching routing rules (with Cost Groups assigned), the routing rule without a Cost Group is considered the lowest cost route.

- The following Cost Groups are configured:

<table>
<thead>
<tr>
<th>Cost Group</th>
<th>Connection Cost</th>
<th>Minute Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. &quot;A&quot;</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2. &quot;B&quot;</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>

- The Cost Groups are assigned to routing rules:

<table>
<thead>
<tr>
<th>Routing Index</th>
<th>Dest Phone Prefix</th>
<th>Destination IP</th>
<th>Cost Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>201</td>
<td>x.x.x.x</td>
<td>&quot;A&quot;</td>
</tr>
<tr>
<td>2</td>
<td>201</td>
<td>x.x.x.x</td>
<td>&quot;B&quot;</td>
</tr>
<tr>
<td>3</td>
<td>201</td>
<td>x.x.x.x</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>201</td>
<td>x.x.x.x</td>
<td>&quot;B&quot;</td>
</tr>
</tbody>
</table>

The device calculates the optimal route in the following index order: 3, 1, 2, and then 4, due to the following logic:

- Index 1 - Cost Group "A" has the lowest connection cost and minute cost
- Index 2 - Cost Group "B" takes precedence over Index 4 entry based on the first-matched method rule
- Index 3 - no Cost Group is assigned, but as the 'Default Call Cost' parameter is configured to **Lowest Cost**, it is selected as the cheapest route
- Index 4 - Cost Group "B" is only second-matched rule (Index 1 is the first)
Example 3: This example shows how the cost of a call is calculated if the call spans over multiple time bands.

Assume a Cost Group, "CG Local" is configured with two time bands, as shown below:

<table>
<thead>
<tr>
<th>Cost Group</th>
<th>Time Band</th>
<th>Start Time</th>
<th>End Time</th>
<th>Connection Cost</th>
<th>Minute Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG Local</td>
<td>TB1</td>
<td>16:00</td>
<td>17:00</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>TB2</td>
<td>17:00</td>
<td>18:00</td>
<td>7</td>
<td>2</td>
</tr>
</tbody>
</table>

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 17-43: 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

17.5.2 Configuring LCR

To configure LCR, perform the following main steps:
2. Configure Cost Groups - see "Configuring Cost Groups" on page 265.
3. Configure Time Bands for a Cost Group - see "Configuring Time Bands for Cost Groups" on page 266.

17.5.2.1 Configuring Cost Groups

The Cost Groups table lets you configure up to 10 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.

The following procedure describes how to configure Cost Groups through the Web interface. You can also configure it through ini file (CostGroupTable) or CLI (configure voip > sip-definition least-cost-routing cost-group).
To configure a Cost Group:

1. Open the Cost Groups table (Setup menu > Signaling & Media tab > SIP Definitions folder > Least Cost Routing > Cost Groups).
2. Click New; the following dialog box appears:

   ![Cost Groups dialog box]

3. Configure a Cost Group according to the parameters described in the table below.
4. Click Apply, and then save your settings to flash memory.

Table 17-16: Cost Groups Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index [CostGroupTable_Index]</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name cost-group-name [CostGroupTable_CostGroupName]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td></td>
<td>The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each Cost Group must have a unique name.</td>
</tr>
<tr>
<td>Default Connection Cost default-connection-cost [CostGroupTable_DefaultConnectionCost]</td>
<td>Defines the call connection cost (added as a fixed charge to the call) for a call outside the time bands.</td>
</tr>
<tr>
<td></td>
<td>The valid value range is 0-65533. The default is 0.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>Default Minute Cost default-minute-cost [CostGroupTable_DefaultMinuteCost]</td>
<td>Defines the call charge per minute for a call outside the time bands.</td>
</tr>
<tr>
<td></td>
<td>The valid value range is 0-65533. The default is 0.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>

17.5.2.1.1 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) and a 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.
- If a Time Band is not configured for a specific day and time range, the default connection cost and default minute cost configured for the Cost Group in the Cost Groups table is applied.

The following procedure describes how to configure Time Bands per Cost Group through the Web interface. You can also configure it through ini file (CostGroupTimebands) or CLI (configure voip > sip-definition least-cost-routing cost-group-time-bands).

To configure a Time Band per Cost Group:

1. Open the Cost Groups table (Setup menu > Signaling & Media tab > SIP Definitions folder > Least Cost Routing > Cost Groups).
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 New; the following dialog box appears:

   ![Time Band Table](image)

4. Configure a Time Band according to the parameters described in the table below.
5. Click Apply, and then save your settings to flash memory.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>start-time</td>
<td>Defines the day and time of day from when this time band is applicable.</td>
</tr>
<tr>
<td>Connection Cost</td>
<td>Defines the minute cost.</td>
</tr>
<tr>
<td>Minute Cost</td>
<td></td>
</tr>
</tbody>
</table>

Table 17-17: Time Band Table Description

- **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).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>End Time</td>
<td>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.</td>
</tr>
<tr>
<td>[CostGroupTimebands_EndTime]</td>
<td></td>
</tr>
<tr>
<td>Connection Cost</td>
<td>Defines the call connection cost during the 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).</td>
</tr>
<tr>
<td>[CostGroupTimebands_ConnectionCost]</td>
<td></td>
</tr>
<tr>
<td>Minute Cost</td>
<td>Defines the call cost per minute charge during the time band. 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).</td>
</tr>
<tr>
<td>[CostGroupTimebands_MinuteCost]</td>
<td></td>
</tr>
</tbody>
</table>

17.5.2.2 Assigning Cost Groups to Routing Rules

To use your configured Cost Groups, you need to assign them to routing rules:
- IP-to-IP Routing table - see Configuring SBC IP-to-IP Routing Rules on page 501

17.6 Remote Web Services

This section describes configuration for remote Web services.

⚠️ Note: To debug remote Web services, see Debugging Web Services on page 805.

17.6.1 Configuring Remote Web Services

The Remote Web Services table lets you configure up to seven Web-based (HTTP/S) services provided by third-party, remote HTTP/S hosts. The following types of services can be offered by the remote host:
- **Routing**: Call routing service, whereby the remote host (e.g., routing server) determines the next hop of an incoming call on the path to the final destination. For more information on employing a third-party, remote routing server, see "Centralized Third-Party Routing Server" on page 275.
- **Call Status**: Call status of calls processed by the device. The call status is provided to the remote host through CDRs sent by the device.
- **Topology Status**: Status of device configuration (add, edit and delete). The device sends topology status to the HTTP host, using the REST TopologyStatus API command. To enable the functionality, see "Enabling Topology Status Services" on page 275.
Topology status includes the following:

- **IP Group Connectivity**: Status is reported when the keep-alive mechanism, enabled for the associated Proxy Set, detects that the IP Group is unavailable, or when CAC thresholds (configured in the Admission Control table) associated with the IP Group are crossed.

- **Configuration Status**: Status is reported when IP Groups or SIP Interfaces that are configured to be used by remote Web-based services (i.e., the UsedByRoutingServer parameter is set to 1 - Used) are created or deleted. If you subsequently change the settings of the UsedByRoutingServer parameter or the ‘Name’ parameter, the device reports the change as a creation or deletion of the corresponding configuration entity.

- **Capture**: Recording of signaling and RTP packets, and Syslog. The remote host can be, for example, a Syslog server or AudioCodes SEM.

**Note:**

- You can configure only **one** Remote Web Service for Routing, for Call Status, and for Topology. However, you can configure up to four Remote Web Services for Capture.

- The Routing service also includes the Call Status and Topology Status services.

- Currently, the Capture service is not supported.

- The device supports HTTP redirect responses (3xx) only during connection establishment with the host. Upon receipt of a redirect response, the device attempts to open a new socket with the host and if this is successful, closes the current connection.

The following procedure describes how to configure Remote Web Services through the Web interface. You can also configure it through ini file (HTTPRemoteServices) or CLI (configure system > http-services > http-remote-services).

➢ **To configure a remote Web service:**

1. Open the Remote Web Services table (**Setup** menu > **IP Network** tab > **Web Services** folder > **Remote Web Services**).
2. Click New; the following dialog box appears:

Figure 17-44: Remote Web Services Table - Add Dialog Box

3. Configure a remote Web service according to the parameters described in the table below.

4. Click Apply, and then save your settings to flash memory.

Table 17-18: Remote Web Services Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index [HTTPRemoteServices_Index]</td>
<td>Defines an index number for the new table row. <strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• Each row must be configured with a unique index.</td>
</tr>
<tr>
<td></td>
<td>• The parameter is mandatory.</td>
</tr>
<tr>
<td>Name rest-name [HTTPRemoteServices_Name]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. <strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• Each row must be configured with a unique name.</td>
</tr>
<tr>
<td></td>
<td>• The parameter is mandatory.</td>
</tr>
<tr>
<td>Type rest-message-type [HTTPRemoteServices_HTTPType]</td>
<td>Defines the type of service provided by the HTTP remote host:</td>
</tr>
<tr>
<td></td>
<td>• [0] Routing (default) = Routing service (also includes Call Status and Topology Status).</td>
</tr>
<tr>
<td></td>
<td>• [1] Call Status = Call status service.</td>
</tr>
<tr>
<td></td>
<td>• [2] Topology Status = Topology status service (e.g., change in configuration).</td>
</tr>
<tr>
<td></td>
<td>• [3] Capture = Recording of signaling and RTP packets, which can be sent to a remote host, for example, to a Syslog server or AudioCodes SEM.</td>
</tr>
<tr>
<td></td>
<td>• [5] QoS = QoS-based call routing. For more information,</td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
| **Path** | see Configuring QoS-Based Routing by Routing Server on page 278.  
**Note:**  
- You can configure only one remote Web service for each of the following service types: Routing, Call Status, Topology Status, and QoS.  
- For the Topology Status option to be functional, you must enable the functionality (see "Enabling Topology Status Services" on page 275).  
- If you do no configure the parameter to QoS, the device sends QoS reports to the Topology server.  
- The Routing option also includes the Call Status and Topology Status services.  
- Currently, the Capture option is not supported.  

| **rest-path** | Defines the path (prefix) to the REST APIs. The valid value is a string of up to 80 characters. The default is "api". |
| **http-service-state** | (Read-only) Displays the status of the host associated with the Web service.  
- "Connected": At least one of the hosts is connected.  
- "Disconnected": All hosts are disconnected.  
- "Not In Service": Configuration of the service is invalid. |

### Connection

| **Policy** | Defines the mode of operation when you have configured multiple remote hosts (in the HTTP Remote Hosts table) for a specific remote Web service.  
- [0] Round Robin = (Default) Load balancing of traffic across all configured hosts. Every consecutive message is sent to the next available host.  
- [1] Sticky Primary = Device always attempts to send traffic to the first (primary) host. If the host does not respond, the device sends the traffic to the next available host. If the primary host becomes available again, the device sends the traffic to the primary host.  
- [2] Sticky Next = Similar to Sticky Primary, but if the primary host does not respond, the device sends the traffic to the next available host and continues sending traffic to this host even if the primary host becomes available again. |
| **http-policy** | Defines whether the HTTP connection with the host remains open or is only opened per request.  
- [0] Disable = Connection is not persistent and closes when the device detects inactivity. The device uses HTTP keep-alive messages to detect inactivity.  
- [1] Enable = (Default) Connection remains open (persistent) even during inactivity. The device uses HTTP keep-alive / HTTP persistent connection messages to keep the connection open. |

<p>| <strong>http-persistent-connection</strong> | Defines how many sockets (connection) are established per |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>http-num-sockets</strong>&lt;br&gt;[HTTPRemoteServices_NumOfSockets]</td>
<td>remote host. The valid value is 1 to 10. The default is 1.</td>
</tr>
<tr>
<td><strong>Login</strong></td>
<td>Enables the use of proprietary REST API Login and Logout commands for connecting to the remote host. The commands verify specific information (e.g., software version) before allowing connectivity with the device.</td>
</tr>
</tbody>
</table>
| | • [0] Disable = Commands are not used.  
• [1] Enable (default) | |
| **Username**<br>rest-user-name<br>[HTTPRemoteServices_AuthUserName] | Defines the username for HTTP authentication. The valid value is a string of up to 80 characters. The default is "user". |
| **Password**<br>rest-password<br>[HTTPRemoteServices_AuthPassword] | Defines the password for HTTP authentication. The valid value is a string of up to 80 characters. The default is "password". |
| **Security** | Assigns a TLS Context for connection with the remote host. By default, no value is defined. |
| | To configure TLS Contexts, see "Configuring TLS Certificate Contexts" on page 105.  
**Note:** The parameter is applicable only if the connection is HTTPS. | |
| **Verify Certificate**<br>rest-verify-certificates<br>[HTTPRemoteServices_VerifyCertificate] | Enables certificate verification when connection with the host is based on HTTPS. |
| | • [0] Disable = (Default) No certificate verification is done.  
• [1] Enable = The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the associated TLS Context.  
**Note:** The parameter is applicable only if the connection is HTTPS. | |
| **Timeouts** | Defines the TCP response timeout (in seconds) from the remote host. If one of the remote hosts does not respond to a request within the specified timeout, the device closes the corresponding socket and attempts to connect to the next |
17.6.1.1 Configuring Remote HTTP Hosts

The HTTP Remote Hosts table lets you configure up to 10 remote HTTP hosts per Remote Web Service. The HTTP Remote Hosts table is a “child” of the Remote Web Services table (configured in “Configuring Remote Web Services” on page 268).

The following procedure describes how to configure HTTP Remote hosts through the Web interface. You can also configure it through ini file (HTTPRemoteServices) or CLI (configure system > http-services > http-remote-hosts).

To configure a remote HTTP host:

1. Open the Remote Web Services table (Setup menu > IP Network tab > Web Services folder > Remote Web Services).
2. In the table, select the required remote Web service index row, and then click the HTTP Remote Hosts link located below the table; the HTTP Remote Hosts table appears.
3. Click New; the following dialog box appears:

   Figure 17-45: HTTP Remote Hosts Table - Add Dialog Box

4. Configure an HTTP remote host according to the parameters described in the table.
5. Click **Apply**, and then save your settings to flash memory.

**Table 17-19: HTTP Remote Hosts Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Index                      | Defines an index number for the new table row. **Note:**  
  - Each row must be configured with a unique index.  
  - The parameter is mandatory.                                                                                                                                                                                                                                               |
| rest-servers               |                                                                                                                                                                                                                                                                                                                                            |
| [HTTPRemoteHosts_RemoteHostIndex] |                                                                                                                                                                                                                                                                                                                                            |
| Name                       | Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. **Note:**  
  - Each row must be configured with a unique name.  
  - The parameter is mandatory.                                                                                                                                                                                                                                               |
| [HTTPRemoteHosts_Name]     |                                                                                                                                                                                                                                                                                                                                            |
| Address                    | Defines the address (IP address or FQDN) of the remote host. The valid value is a string of up to 80 characters. **Note:**  
  - An IPv6 address can only be configured if the interface is a CONTROL type.  
  - If the address is an FQDN and the DNS resolution results in multiple IP addresses, the device attempts to establish multiple connections (sessions) for each IP address. Only the first 10 resolved IP addresses are used regardless of the number of hosts.  
  - FQDN resolution is also performed (immediately) when connection is subsequently "closed" (by timeout or by the remote host) and connections are updated accordingly. In addition, the device periodically (every 15 minutes) performs DNS name resolution to ensure that the list of resolved IP addresses has not changed. If a change is detected, the device updates its’ list of IP addresses and re-establishes connections accordingly.  
  - In addition to multiple HTTP sessions, the device establishes multiple (TCP) connections per session, thereby enhancing data exchange capabilities with the host. |
| rest-address               |                                                                                                                                                                                                                                                                                                                                            |
| [HTTPRemoteHosts_Address]  |                                                                                                                                                                                                                                                                                                                                            |
| Port                       | Defines the port of the host. The valid value is 0 to 65535. The default is 80.                                                                                                                                                                                                                                                        |
| rest-port                  |                                                                                                                                                                                                                                                                                                                                            |
| [HTTPRemoteHosts_Port]     |                                                                                                                                                                                                                                                                                                                                            |
| Interface                  | Assigns one of the device's IP network interfaces through which communication with the remote host is done. By default, no value is defined and the OAMP interface is used.                                                                                                                                                     |
| rest-interface             |                                                                                                                                                                                                                                                                                                                                            |
| [HTTPRemoteHosts_Interface] |                                                                                                                                                                                                                                                                                                                                           |
| Transport Type             | Defines the protocol for communicating with the remote host.                                                                                                                                                                                                                                                                             |
|                           |                                                                                                                                                                                                                                                                                                                                            |
### 17. Services

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rest-transport-type</td>
<td>host:</td>
</tr>
<tr>
<td>[HTTPRemoteHosts_HTTPTransportType]</td>
<td>• [0] HTTP (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] HTTPS</td>
</tr>
<tr>
<td>Status</td>
<td>(Read-only) Displays the status of the connection with the remote host.</td>
</tr>
<tr>
<td>http-host-state</td>
<td>• &quot;Connected&quot;: The host is connected.</td>
</tr>
<tr>
<td></td>
<td>• &quot;Disconnected&quot;: The host is disconnected.</td>
</tr>
<tr>
<td></td>
<td>• &quot;Not In Service&quot;: Configuration of the host is invalid.</td>
</tr>
</tbody>
</table>

#### 17.6.2 Enabling Topology Status Services

The following procedure describes how to enable Topology Status for Web-based services. For more information on Topology Status services, see "Configuring Remote Web Services" on page 268.

➢ To enable Topology Status services:

1. Open the Web Service Settings page (Setup menu > IP Network tab > Web Services folder > Web Service Settings).
2. From the 'Topology Status' drop-down list (RoutingServerGroupStatus), select **Enable**.

![Figure 17-46: Enabling Topology Status Web-based Service](image)

3. Click **Apply**.

#### 17.6.3 Centralized Third-Party Routing Server

You can employ a remote, third-party Routing server to handle call routing decisions in deployments consisting of multiple AudioCodes devices. Employing a Routing server replaces the need for the device's routing tables (IP-to-IP Routing table) to determine call destination.

When the device receives an incoming call (SIP INVITE, NOTIFY or MESSAGE), it searches the IP-to-IP Routing table for a matching routing rule that is also configured to use a Routing server. If found, the device requests the Routing server for an appropriate destination. The request is sent to the Routing server using an HTTP Get Route message. The request contains information about the call (SIP message).

The Routing server uses its own algorithms and logic in determining the best route path. The Routing server manages the call route between devices in "hops", which may be spread over different geographical locations. The destination to each hop (device) can be by IP address (with port) or IP Group. If the destination is an IP address, even though the destination type (in the IP-to-IP Routing table) is an IP Group, the device only uses the IP Group for profiling (i.e., associated IP Profile etc.). If multiple devices exist in the call routing path, the Routing server sends the IP address only to the last device ("node") in the path.

Once the device receives the resultant destination hop from the Routing server, it sends the call to that destination. The Routing server can provide the device with an appropriate
route or reject the call. However, if for the initial request (first sent Get Route request for the call) the Routing server cannot find an appropriate route for the call or it does not respond, for example, due to connectivity loss (i.e., the Routing server sends an HTTP 404 "Not Found" message), the device routes the call using its routing tables. If the Get Route request is not the first one sent for the call (e.g., in call forwarding or alternative routing) and the Routing server responds with an HTTP 404 "Not Found" message, the device rejects the call.

This HTTP request-response transaction for the routing path occurs between Routing server and each device in the route path (hops) as the call traverses the devices to its final destination. Each device in the call path connects to the Routing server, which responds with the next hop in the route path. Each device considers the call an incoming call from an IP Group. The session ID (SID) is generated by the first device in the path and then passed unchanged down the route path, enabling the Routing server to uniquely identify requests belonging to the same call session.

Communication between the device and the Routing server is through the device's embedded Representational State Transfer (RESTful) API. The RESTful API is used to manage the routing-related information exchanged between the Routing server (RESTful server) and the device (RESTful client). When you have configured the device with connection settings of the Routing server and the device starts-up, it connects to the Routing server and activates the RESTful API, which triggers the routing-related API commands.

The following figure provides an example of information exchange between devices and a Routing server for routing calls:

**Figure 17-47: Example of Call Routing Information Exchange between Devices and Routing Server**

The Routing server can also manipulate call data such as calling name, if required. It can also create new IP Groups and associated configuration entities, if necessary for routing. Multiple Routing servers can also be employed, whereby each device in the chain path can use a specific Routing server. Alternatively, a single Routing server can be employed and used for all devices ("stateful" Routing server).

The device automatically updates (sends) the Routing server with its' configuration topology regarding SIP routing-related entities (SRDs, SIP Interfaces, and IP Groups) that
have been configured for use by the Routing server. For example, if you add a new IP Group and enable it for use by the Routing server, the device sends this information to the Routing server. Routing of calls associated with routing-related entities that are disabled for use by the Routing server (default) are handled only by the device (not the Routing server).

In addition to regular routing, the Routing server also supports the following:

- **Alternative Routing:** If a call fails to be established, the device "closest" to the failure and configured to send "additional" routing requests (through REST API - "additionalRoute" attribute in HTTP Get Route request) to the Routing server, sends a new routing request to the Routing server. The Routing server may respond with a new route destination, thereby implementing alternative routing. Alternatively, it may enable the device to return a failure response to the previous device in the route path chain and respond with an alternative route to this device. Therefore, alternative routing can be implemented at any point in the route path. If the Routing server sends an HTTP 404 "Not Found" message for an alternative route request, the device rejects the call. If the Routing server is configured to handle alternative routing, the device does not make any alternative routing decisions based on its alternative routing tables.

- **Call Status:** The device can report call status to the Routing server to indicate whether a call has successfully been established and/or failed (disconnected). The device can also report when an IP Group (Proxy Set) is unavailable, detected by the keep-alive mechanism, or when the CAC thresholds permitted per IP Group have been crossed.

- **Credentials for Authentication:** The Routing Server can provide user (e.g., IP Phone caller) credentials (username-password) in the Get Route response, which can be used by the device to authenticate outbound SIP requests if challenged by the outbound peer, for example, Microsoft Skype for Business (per RFC 2617 and RFC 3261). If multiple devices exist in the call routing path, the Routing server sends the credentials only to the last device ("node") in the path.

- **QoS:** The device can report QoS metrics per IP Group to the Routing server, which the Routing server can use to determine the best route (i.e., QoS-based routing). For more information, see Configuring QoS-Based Routing by Routing Server on page 278.

- **Call Preemption for Emergency Calls:** If you enable call preemption for emergency calls (e.g., 911) on the device, the routing server determines whether or not the incoming call is an emergency call and if so, handles the routing decision accordingly (i.e., preempts a non-emergency call if the maximum call capacity of the device is reached in order to allow the emergency call to be routed). To enable call preemption for emergency calls, use the parameter SBCPreemptionMode for SBC calls and CallPriorityMode for Gateway calls.

- To configure routing based on Routing server:

1. For each configuration entity (e.g., IP Group) that you want routing done by the Routing server, configure the entity's 'Used By Routing Server' parameter to **Used**.

   **Figure 17-48: Configuring Entity to Use Routing Server**

<table>
<thead>
<tr>
<th>Used By Routing Server</th>
<th>Used</th>
</tr>
</thead>
</table>

2. Configure an additional Security Administrator user account in the Local Users table (see "Configuring Management User Accounts" on page 73), which is used by the Routing server (REST client) to log in to the device's management interface.
3. Configure the address and connection settings of the Routing server, referred to as a Remote Web Service and HTTP remote host (see "Configuring Remote Web Services" on page 268). You must configure the 'Type' parameter of the Remote Web Service to Routing, as shown in the following example:

**Figure 17-49: Configuring Remote Web Service for Routing Server**

```
<table>
<thead>
<tr>
<th>Index</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Routing Server</td>
<td>Routing</td>
</tr>
</tbody>
</table>
```

4. In the IP-to-IP Routing table, configure the 'Destination Type' parameter of the routing rule to Routing Server (see Configuring SBC IP-to-IP Routing Rules on page 501), as shown below:

**Figure 17-50: Configuring Routing Rule to use Routing Server**

```
<table>
<thead>
<tr>
<th>Destination Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Server</td>
</tr>
</tbody>
</table>
```

**17.6.4 Configuring QoS-Based Routing by Routing Server**

You can configure the device to allow the routing server to route calls based on QoS metrics (media and signaling). The device collects QoS metrics per IP Group that you have configured to operate with the routing server ('Used by Routing Server' parameter configured to Used in the IP Groups table). The metrics include the following:

- **Signaling**: ASR, NER, and ACD
- **Media**: packet loss (Rx/Tx), packet delay (local/remote), jitter (local/remote), MOS (local/remote), audio bandwidth (Rx/Tx), video bandwidth (Rx/Tx), and total bandwidth (Rx/Tx)

The device collects QoS metrics for both incoming call traffic and outgoing traffic from the remote endpoint. It sends the QoS reports to the routing server, where each report can contain the status of up to 100 IP Groups. If more than 100 IP Groups exist, the device sends multiple QoS reports (sequentially) to the routing server. The device sends the reports every user-defined period. The routing logic of where to route calls based on QoS ("good", "fair", and "bad") is configured on the routing server.

- To configure the device for QoS-based routing by routing server:
  1. Open the Web Service Settings page (Setup menu > IP Network tab > Web Services folder > Web Service Settings), and then do the following:
     a. From the 'Quality Status' drop-down list (RoutingServerQualityStatus), select **Enable** to enable QoS-based routing.
b. In the 'Quality Status Rate' field (RoutingServerQualityStatusRate), enter the rate (in sec) at which the device sends QoS reports.

Figure 17-51: Enabling QoS-Based Routing via Routing Server

<table>
<thead>
<tr>
<th>Quality Status</th>
<th>Enable</th>
<th>Rate (sec)</th>
</tr>
</thead>
</table>

C. Click **Apply**.

2. Open the Remote Web Services table (see Configuring Remote Web Services on page 278), and then for the Remote Web Service entry that you configured for the routing server, do the following:

a. From the 'Type' drop-down list (HTTPRemoteServices_HTTPType), select **QoS**.

b. Click **Apply**.

3. Enable voice quality monitoring and RTCP XR, using the 'Enable RTCP XR' (VQMonEnable) parameter (see Configuring RTCP XR on page 757).

**Note:** For media metrics calculations, the device's License Key must include voice quality monitoring and RTCP XR.

### 17.7 HTTP-based Proxy Services

The device supports the following HTTP-based proxy services:

**HTTP Reverse Proxy for Managing Equipment behind NAT:**

You can configure the device to function as a reverse HTTP proxy server. This functionality is required to enable administrators to manage communication equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and the administrator is located in a public domain (e.g., in the WAN). Thus, this functionality resolves NAT issues, enabling the administrator to access the IP Phone's management interface (e.g., embedded Web server).

To support the functionality, the following configuration is required:

1. Enable the HTTP Proxy application (see "Enabling the HTTP Proxy Application" on page 280).

2. Define a local, listening HTTP interface for the leg interfacing with the administrator (see "Configuring HTTP Interfaces" on page 281).

**Note:** It is recommended **not** to use port 80 as this is the default port used by IP Phones for their Web-based management interface.

3. Define each HTTP-based managed equipment:

   a. Define the URL prefix for accessing the equipment's management interface (see "Configuring HTTP Proxy Services" on page 283). To access the equipment's management interface, the administrator needs to enter the following URL in a Web browser:

   http://<device's WAN IP address:port>/url prefix/
b. Define the IP address of the managed equipment (see "Configuring HTTP Proxy Hosts" on page 284).

Note: For this feature, no special configuration is required on the managed equipment.

**HTTP-based EMS Services for AudioCodes Equipment behind NAT:**

You can configure the device to act as an HTTP Proxy that enables AudioCodes EMS to manage AudioCodes equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and EMS is located in a public domain (e.g., in the WAN). Thus, the feature resolves NAT traversal issues. The IP Phones register with the device in order to allow communication between the IP Phones and the EMS.

To support the functionality, the following configuration is required:

1. Enable the HTTP Proxy application (see "Enabling the HTTP Proxy Application" on page 280).
2. Configure two local, listening HTTP interfaces - one for the EMS and one for the IP Phones (see "Configuring HTTP Interfaces" on page 281).
3. Configure the address of the EMS server (see "Configuring an HTTP-based EMS Service" on page 286).

### 17.7.1 Enabling the HTTP Proxy Application

Before you can configure HTTP-based proxy services, you must enable the HTTP Proxy application, as described in the following procedure. Once enabled, the Web interface displays menus in the Navigation pane that are relevant to the HTTP Proxy application.

Note: The HTTP Proxy application is a license-dependent feature and is available only if it is included in the License Key installed on the device. For ordering the feature, please contact your AudioCodes sales representative. For installing a new License Key, see License Key on page 669.

To enable the HTTP Proxy application:

1. Open the General Settings page (Setup menu > IP Network tab > HTTP Proxy folder > General Settings).

   ![Figure 17-52: Enabling HTTP Proxy Application](image)

2. From the 'HTTP Proxy Application' drop-down list, select Enable.
3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.
17.7.2 Debugging Remote HTTP Services

You can enable the device to generate debug messages for remote Web (HTTP) services and send them to a Syslog server.

➢ To enable debugging of HTTP services:
1. Open the General Settings page (Setup menu > IP Network tab > HTTP Proxy folder > General Settings):

   ![HTTP Proxy Debug Level](image)

2. From the 'HTTP Proxy Debug Level' drop-down list, select one of the following:
   - **No Debug**: Disables debugging.
   - **Basic**: Enables debugging and sends basic information in the Syslog.
   - **Detailed**: Enables debugging and sends detailed information in the Syslog.
3. Click Apply.

17.7.3 Configuring HTTP Interfaces

The HTTP Interfaces table lets you configure up to 10 HTTP Interfaces. An HTTP Interface represents a local, listening interface for receiving HTTP/S requests from HTTP-based (Web) clients such as managed equipment (e.g., IP Phones) and/or AudioCodes EMS management tool for HTTP/S-based services.

The following procedure describes how to configure HTTP Interfaces through the Web interface. You can also configure it through ini file (HTTPInterface) or CLI (configure network > http-proxy http-interface).

➢ To configure an HTTP Interface:
1. Open the HTTP Interfaces table (Setup menu > IP Network tab > HTTP Proxy folder > HTTP Interfaces).
2. Click **New**; the following dialog box appears:

   ![HTTP Interfaces Table - Add Dialog Box](image)

3. Configure an HTTP Interface according to the parameters described in the table below.
4. Click **Apply**, and then save your settings to flash memory.

**Table 17-20: HTTP Interfaces Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
</tbody>
</table>
| [HTTPInterface_Index]      | **Note:**  
  - Each row must be configured with a unique index.  
  - The parameter is mandatory.                                                                                                                     |
| Name                       | Defines a descriptive name, which is used when associating the row in other tables.                                                                                                                        |
| interface-name             | **Note:**  
  - Each row must be configured with a unique name.  
  - The parameter is mandatory.                                                                                                                     |
| Network Interface          | Assigns a local, network interface to the HTTP interface.                                                                                                                                                     |
| network-interface          | By default, no value is defined.                                                                                                                                                                            |
| [HTTPInterface_NetworkInterface] | To configure network interfaces, see "Configuring IP Network Interfaces" on page 137.                                                                                                                          |
| **Protocol**               |                                                                                                                                                                                                             |
| protocol                   | Defines the protocol type.                                                                                                                                                                                  |
| [HTTPInterface_Protocol]   |  
  - [0] HTTP (default)  
  - [1] HTTPS                                                                                                                                                |
| HTTP Port                  | Defines the local, listening HTTP port.                                                                                                                                                                      |
| http-port                  | The valid value is 0 to 65534. The default is 0.                                                                                                                                                            |
| [HTTPInterface_Port]       | **Note:** The parameter is mandatory.                                                                                                                                                                       |
| **Security**               |                                                                                                                                                                                                             |
| TLS Context                | Assigns a TLS Context for the connection with the HTTP Proxy service.                                                                                                                                          |
| tls-context                | By default, the default TLS Context (Index 0) is assigned.                                                                                                                                                   |
| [HTTPInterface_TLSContext] | To configure TLS Contexts, see "Configuring TLS Certificate Contexts" on page 105.                                                                                                                               |
| **Verify Certificate**     | Enables TLS certificate verification when the connection with the proxy service is based on HTTPS.                                                                                                          |
| verify-cert                 |  
  - [0] No = (Default) No certificate verification is done.  
  - [1] Yes = The device verifies the authentication of the certificate received from the HTTPS peer. The device authenticates the certificate against the trusted root certificate store associated with the associated TLS Context (see 'TLS Context' parameter above) and if ok, allows communication with the HTTPS peer. If authentication fails, the device denies communication (i.e., handshake fails). The device can also authenticate the certificate by querying with an Online Certificate Status Protocol (OCSP) server whether the certificate has been revoked. This is also configured for the |
| [HTTPInterface_VerifyCert] |                                                                                                                                                                                                             |
17. Services

17.7.4 Configuring HTTP Proxy Services

The HTTP Proxy Services table lets you configure up to 10 HTTP Proxy Services. The following procedure describes how to configure HTTP Proxy Services through the Web interface. You can also configure it through ini file (HTTPProxyService) or CLI (configure network > http-proxy http-proxy-serv).

➢ To configure an HTTP Proxy Service:

1. Open the HTTP Proxy Services table (Setup menu > IP Network tab > HTTP Proxy folder > HTTP Proxy Services).
2. Click New; the following dialog box appears:

```
Figure 17-54: HTTP Proxy Services Table - Add Dialog Box
```

3. Configure an HTTP Proxy service according to the parameters described in the table below.
4. Click Apply, and then save your settings to flash memory.

Table 17-21: HTTP Proxy Services Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[HTTPProxyService_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td></td>
<td>• The parameter is mandatory.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td>service-name</td>
<td>The valid value is a string of up to 40 characters. By default, no value is defined.</td>
</tr>
<tr>
<td>[HTTPProxyService_ServiceName]</td>
<td>Note:</td>
</tr>
</tbody>
</table>
### Parameter Description

- **Listening Interface**
  - **listening-int**
  - Assigns an HTTP Interface to the HTTP Proxy service. To configure HTTP Interfaces, see "Configuring HTTP Interfaces" on page 281.
  - **Note:** The parameter is mandatory.

- **URL Prefix**
  - **url-prefix**
  - Defines the URL prefix that is used to access the managed equipment's embedded Web server. The URL prefix is matched against the target of the HTTP requests sent by the client (such as GET and POST). If a match is located in the table, the device removes the prefix from the request and then forwards the HTTP request to the managed equipment without the prefix. For example, for the URL of `GET /home/index.html HTTP/1.1` (which is part of the URL `http://10.20.30.40/home/index.html`), a URL prefix of `"/home"` can be configured. To match all URLs, configure the parameter to `"/"` (default).

- **Keep-Alive Mode**
  - **keep-alive-mode**
  - Enables a keep-alive mechanism with the managed equipment:
    - [0] Disable
    - [1] Options = (Default) Enables keep-alive by sending HTTP OPTIONS messages. If no response is received from the HTTP host, the device stops forwarding HTTP requests to the host and raises an SNMP alarm (achHTTPProxyServiceAlarm). If you configured the address of the host as an FQDN (see "Configuring HTTP Proxy Hosts" on page 284) and the DNS resolution results in multiple IP addresses, when no response is received from the keep-alive, the device checks connectivity with the next resolved IP address and so on, until a response is received.

### 17.7.4.1 Configuring HTTP Proxy Hosts

The HTTP Proxy Hosts table lets you configure HTTP Proxy hosts for HTTP Proxy services. An HTTP Proxy Host represents the HTTP-based managed equipment (e.g., IP Phone). The table is a "child" of the HTTP Proxy Services table (see "Configuring HTTP Proxy Services" on page 283). You can configure up to 50 HTTP Proxy hosts; up to 5 HTTP Proxy hosts per HTTP Proxy Service.

The following procedure describes how to configure HTTP Remote hosts through the Web interface. You can also configure it through ini file (HTTPProxyHost) or CLI (configure network > http-proxy http-proxy-host).

- **To configure an HTTP Proxy Host:**
  1. Open the HTTP Proxy Services table (Setup menu > IP Network tab > HTTP Proxy folder > HTTP Proxy Services).
  2. In the table, select the required HTTP Proxy Service index row, and then click the HTTP Proxy Hosts link located below the table; the HTTP Proxy Hosts table appears.
3. Click **New**; the following dialog box appears:

![HTTP Proxy Hosts Table - Add Dialog Box](image)

**Figure 17-55: HTTP Proxy Hosts Table - Add Dialog Box**

4. Configure an HTTP Proxy Host according to the parameters described in the table below.

5. Click **Apply**, and then save your settings to flash memory.

**Table 17-22: HTTP Proxy Hosts Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>Note:</td>
<td>• Each row must be configured with a unique index.</td>
</tr>
<tr>
<td></td>
<td>• The parameter is mandatory.</td>
</tr>
<tr>
<td>Network Interface</td>
<td>Assigns a local, network interface to the HTTP Proxy Host.</td>
</tr>
<tr>
<td>network-interface</td>
<td>By default, no value is defined.</td>
</tr>
<tr>
<td>[HTTPProxyHost_NetworkInterface]</td>
<td>To configure network interfaces, see &quot;Configuring IP Network Interfaces&quot; on page 137.</td>
</tr>
<tr>
<td>Proxy Address</td>
<td>Defines the address of the managed equipment (host).</td>
</tr>
<tr>
<td>proxy-address</td>
<td>The valid value is an IP address in dotted-decimal notation or an FQDN (up to 100 characters).</td>
</tr>
<tr>
<td>[HTTPProxyHost_IpAddress]</td>
<td>If the address is an FQDN, the device uses DNS to resolve it into an IP address. If the DNS</td>
</tr>
<tr>
<td></td>
<td>resolution results in multiple IP addresses, the device uses the first available address (i.e.,</td>
</tr>
<tr>
<td></td>
<td>that responds to the keep-alive).</td>
</tr>
<tr>
<td>Protocol</td>
<td>Defines the protocol type.</td>
</tr>
<tr>
<td>protocol</td>
<td>• [0] HTTP (default)</td>
</tr>
<tr>
<td>HTTP Port</td>
<td>Defines the port of the managed equipment.</td>
</tr>
<tr>
<td>http-port</td>
<td>The default is 0.</td>
</tr>
<tr>
<td>[HTTPProxyHost_Port]</td>
<td><strong>Note</strong>: The parameter is mandatory.</td>
</tr>
<tr>
<td><strong>Security</strong></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TLS Context</td>
<td>Assigns a TLS Context for the TLS connection with the HTTP Proxy host. By default, the default TLS Context (Index 0) is assigned. To configure TLS Contexts, see &quot;Configuring TLS Certificate Contexts&quot; on page 105.</td>
</tr>
<tr>
<td>verify-cert</td>
<td>Enables TLS certificate verification when the connection with the host is based on HTTPS.</td>
</tr>
</tbody>
</table>

**Note:** The parameter is applicable only if the connection protocol is HTTPS (defined using the 'Protocol' parameter, above).

### 17.7.5 Configuring an HTTP-based EMS Service

The EMS Services table lets you configure a single HTTP-based EMS service. For more information on the EMS service, see "HTTP-based Proxy Services" on page 279.

The following procedure describes how to configure an EMS Service through the Web interface. You can also configure it through ini file (EMSService) or CLI (configure network > http-proxy ems-serv).

1. To configure an EMS Service:
   1. Open the EMS Services table (Setup menu > IP Network tab > HTTP Proxy folder > EMS Services).
2. Click **New**; the following dialog box appears:

   ![Figure 17-56: EMS Services Table - Add Dialog Box](image)

3. Configure an EMS Service according to the parameters described in the table below.
4. Click **Apply**, and then save your settings to flash memory.

   **Table 17-23: EMS Services Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index [EMSService_Index]</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>🔄 Each row must be configured with a unique index.</td>
</tr>
<tr>
<td></td>
<td>🔄 The parameter is mandatory.</td>
</tr>
<tr>
<td>Name service-name [EMSService_ServiceName]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>🔄 Each row must be configured with a unique name.</td>
</tr>
<tr>
<td></td>
<td>🔄 The parameter is mandatory.</td>
</tr>
<tr>
<td>EMS Primary Server primary-server [EMSService_PrimaryServer]</td>
<td>Defines the address of the primary EMS server.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The parameter is mandatory.</td>
</tr>
<tr>
<td>Listening Interface to devices dev-login-int [EMSService_DeviceLoginInterface]</td>
<td>Assigns an HTTP Interface (local, listening HTTP interface:port) for communication with the client. To configure HTTP Interfaces, see &quot;Configuring HTTP Interfaces&quot; on page 281.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> By default, no value is defined.</td>
</tr>
<tr>
<td></td>
<td>🔄 The parameter is mandatory.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Listening to EMS Interface</td>
<td>Assigns an HTTP Interface (local, listening HTTP interface:port) for communication with the EMS. To configure HTTP Interfaces, see &quot;Configuring HTTP Interfaces&quot; on page 281. By default, no value is defined. Note: The parameter is mandatory.</td>
</tr>
</tbody>
</table>

### 17.8 E9-1-1 Support for Microsoft Skype for Business

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 Skype for Business and AudioCodes' device's ELIN interworking capabilities, which provides the SIP Trunk to the E9-1-1 emergency service provider. This section also describes the configuration of the device for interoperating between the Skype for Business environment and the E9-1-1 emergency provider.

**Note:** The ELIN feature for E9-1-1 is a license-dependent feature and is available only if it is included in the License Key installed on the device. For ordering the feature, please contact your AudioCodes sales representative. For installing a new License Key, see "License Key" on page 669.

### 17.8.1 About E9-1-1 Services

E9-1-1 is a national emergency service for many countries, enabling E9-1-1 operators to automatically identify the geographical location and phone number of a 911 caller. In E9-1-1, the 911 caller is routed to the nearest E9-1-1 operator, termed public safety answering point (PSAP) based on the location of the caller. Automatic identification of the caller's location and phone number reduces the time spent on requesting this information from the 911 caller. Therefore, the E9-1-1 service enables the PSAP to quickly dispatch the relevant emergency services (for example, fire department or police) to the caller's location. Even if the call prematurely disconnects, the operator has sufficient information to call back the 911 caller.

The figure below illustrates the routing of an E9-1-1 call to the PSAP:
1. The VoIP user dials 9-1-1.
2. AudioCodes’ ELIN device sends the call to the emergency service provider over the SIP Trunk (PSAP server).
3. The emergency service provider identifies the call is an emergency call and sends it to an E9-1-1 Selective Router in the Emergency Services provider’s network.
4. The E9-1-1 Selective Router determines the geographical location of the caller by requesting this information from an Automatic Location Identification (ALI) database based on the phone number or Automatic Number Identifier (ANI) of the 911 caller. Exact location information is also supplied by the Master Street Address Guide (MSAG) database, which is a companion database to the ALI database. Phone companies and public safety agencies collaborate beforehand to create master maps that match phone numbers, addresses and cross streets to their corresponding PSAP. This MSAG is the official record of valid streets (with exact spelling), street number ranges, and other address elements with which the service providers are required to update their ALI databases.
5. The E9-1-1 Selective Router sends the call to the appropriate PSAP based on the retrieved location information from the ALI.
6. The PSAP operator dispatches the relevant emergency services to the E9-1-1 caller.

17.8.2 Microsoft Skype for Business and E9-1-1

Microsoft Skype for Business 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, Skype for Business offers an innovative solution to solving Enterprises E9-1-1 location problems.

17.8.2.1 Gathering Location Information of Skype for Business Clients for 911 Calls

When a Microsoft® Skype for Business 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 Skype for
Business client registration process or when the operating system detects a network connection change, each Skype for Business 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 Skype for Business client dials 9-1-1, this location information is then included as part of the emergency call and used by the emergency service provider to route the call to the correct PSAP.

The gathering of location information in the Skype for Business network is illustrated in the figure below:

**Figure 17-57: Microsoft Skype for Business Client Acquiring Location Information**

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 291.

2. The Administrator validates addresses with the emergency service provider’s MSAG – a companion database to the ALI database. This ensures that the civic address is valid as an official address (e.g., correct address spelling).

3. The Skype for Business client initiates a location request to the LIS under the following circumstances:
   - Immediately after startup and registering the user with Skype for Business
   - Approximately every four hours after initial registration
   - Whenever a network connection change is detected (such as roaming to a new WAP)

The Skype for Business 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 Skype for Business client that moves inside the corporate network such as a soft phone on a laptop that connects wirelessly to the corporate network, Skype for Business 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 Skype for Business 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.

17.8.2.2 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>
<thead>
<tr>
<th>Network Element</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wireless access point</td>
<td><code>&lt;BSSID&gt;</code>,<code>&lt;Description&gt;</code>,<code>&lt;Location&gt;</code>,<code>&lt;CompanyName&gt;</code>,<code>&lt;HouseNumber&gt;</code>,<code>&lt;HouseNumberSuffix&gt;</code>,<code>&lt;PreDirectional&gt;</code>,...<code>&lt;StreetName&gt;</code>,<code>&lt;StreetSuffix&gt;</code>,<code>&lt;PostDirectional&gt;</code>,<code>&lt;City&gt;</code>,<code>&lt;State&gt;</code>,<code>&lt;PostalCode&gt;</code>,<code>&lt;Country&gt;</code></td>
</tr>
<tr>
<td>Subnet</td>
<td><code>&lt;Subnet&gt;</code>,<code>&lt;Description&gt;</code>,<code>&lt;Location&gt;</code>,<code>&lt;CompanyName&gt;</code>,<code>&lt;HouseNumber&gt;</code>,<code>&lt;HouseNumberSuffix&gt;</code>,<code>&lt;PreDirectional&gt;</code>,...<code>&lt;StreetName&gt;</code>,<code>&lt;StreetSuffix&gt;</code>,<code>&lt;PostDirectional&gt;</code>,<code>&lt;City&gt;</code>,<code>&lt;State&gt;</code>,<code>&lt;PostalCode&gt;</code>,<code>&lt;Country&gt;</code></td>
</tr>
<tr>
<td>Port</td>
<td><code>&lt;ChassisID&gt;</code>,<code>&lt;PortIDSubType&gt;</code>,<code>&lt;PortID&gt;</code>,<code>&lt;Description&gt;</code>,<code>&lt;Location&gt;</code>,<code>&lt;CompanyName&gt;</code>,<code>&lt;HouseNumber&gt;</code>,<code>&lt;HouseNumberSuffix&gt;</code>,<code>&lt;PreDirectional&gt;</code>,...<code>&lt;StreetName&gt;</code>,<code>&lt;StreetSuffix&gt;</code>,<code>&lt;PostDirectional&gt;</code>,<code>&lt;City&gt;</code>,<code>&lt;State&gt;</code>,<code>&lt;PostalCode&gt;</code>,<code>&lt;Country&gt;</code></td>
</tr>
</tbody>
</table>
For the ELIN number to be included in the SIP INVITE (XML-based PIDF-LO message) sent by the Mediation Server to the ELIN device, the administrator must add the ELIN number to the `<CompanyName>` column (shown in the table above in **bold** typeface). As the ELIN device supports up to five ELINs per PIDF-LO, the `<CompanyName>` column can be populated with up to this number of ELINs, each separated by a semicolon. The digits of each ELIN can be separated by hyphens (xxx-xxx-xxx) or they can be adjacent (xxxxxxxxxx).

When the ELIN device receives the SIP INVITE, it extracts the ELINs from the NAM field in the PIDF-LO (e.g., `<ca:NAM>1111-222-333; 1234567890 </ca:NAM>`), which corresponds to the `<CompanyName>` column of the LIS.

If you do not populate the location database, and the Skype for Business location policy, Location Required is set to **Yes** or **Disclaimer**, the user will be prompted to enter a location manually.

### 17.8.2.3 Passing Location Information to the PSTN Emergency Provider

When a Skype for Business client, enabled for E9-1-1 emergency services, dials 9-1-1, the location data and callback information stored on the client is sent with the call through the Mediation Server to a SIP Trunk-based emergency service provider. The emergency service provider then routes the call to the nearest and most appropriate PSAP based on the location information contained within the call.

Skype for Business passes the location information of the Skype for Business client in an IETF-standard format - Presence Information Data Format - Location Object (PIDF-LO)–in a SIP INVITE message. However, this content cannot be sent on the SIP Trunks since they do not support such a content. To overcome this, Enterprises deploying the device can divide their office space into Emergency Response Locations (ERLs) and assign a dedicated Emergency Location Identification Number (ELIN) to each ERL (or zone). When Skype for Business sends a SIP INVITE message with the PIDF-LO to the device, it can parse the content and translate the calling number to an appropriate ELIN. The device then sends the call to the SIP Trunk with the ELIN number as the calling number. The ELIN number is sent to the emergency service provider, which sends it on to the appropriate PSAP according to the ELIN address match in the ALI database lookup.

The ERL defines a specific location at a street address, for example, the floor number of the building at that address. The geographical size of an ERL is according to local or national regulations (for example, less than 7000 square feet per ERL). Typically, you would have an ERL for each floor of the building. The ELIN is used as the phone number for 911 callers within this ERL.
The figure below illustrates the use of ERLs and ELINs, with an E9-1-1 call from floor 2 at the branch office:

**Figure 17-58: ERLs and ELINs for E9-1-1 in Skype for Business**

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 17-25: Designating ERLs and Assigning to ELINs**

<table>
<thead>
<tr>
<th>ERL Number</th>
<th>Physical Area</th>
<th>IP Address</th>
<th>ELIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Floor 1</td>
<td>10.13.124.xxx</td>
<td>503 972-4410</td>
</tr>
<tr>
<td>2</td>
<td>Floor 2</td>
<td>10.15.xxx.xxx</td>
<td>503 972-4411</td>
</tr>
<tr>
<td>3</td>
<td>Floor 3</td>
<td>10.18.xxx.xxx</td>
<td>503 972-4412</td>
</tr>
</tbody>
</table>

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.

**17.8.3 AudioCodes ELIN Device for Skype for Business E9-1-1 Calls to PSTN**

Microsoft Mediation Server sends the location information of the E9-1-1 caller in the XML-based PIDF-LO body contained in the SIP INVITE message. However, this content cannot be sent on the SIP Trunk since they do not support such content. To solve this issue, Skype for Business requires a device (ELIN SBC) to send the E9-1-1 call to the SIP Trunk. When Skype for Business sends the PIDF-LO to the device, it parses the content and translates the calling number to an appropriate ELIN. This ensures that the call is routed to...
an appropriate PSAP, based on ELIN-address match lookup in the emergency service provider's ALI database.

The figure below illustrates an AudioCodes ELIN device deployed in the Skype for Business environment for handling E9-1-1 calls between the Enterprise and the emergency service provider.

**Figure 17-59: ELIN SBC for E9-1-1 in Skype for Business Environment**

17.8.3.1 Detecting and Handling E9-1-1 Calls

The ELIN device identifies E9-1-1 calls and translates their incoming E9-1-1 calling numbers into ELIN numbers, sent toward the PSAP. The device handles the received E9-1-1 calls as follows:

1. The device identifies E9-1-1 calls if the incoming SIP INVITE message contains a PIDF-LO XML message body. This is indicated in the SIP Content-Type header, as shown below:

   ```
   Content-Type: application/pidf+xml
   ```

2. The device extracts the ELIN number(s) from the "NAM" field in the XML message. The "NAM" field corresponds to the <CompanyName> column in the Location Information Server (LIS). The device supports up to five ELIN numbers per XML message. The ELINs are separated by a semicolon. The digits of the ELIN number can be separated by hyphens (xxx-xxx-xxx) or they can be adjacent (xxxxxxxxx), as shown below:

   ```
   <ca:NAM>1111-222-333; 1234567890</ca:NAM>
   ```

3. The device saves the From header value of the SIP INVITE message in its ELIN database table (Call From column). The ELIN table is used for PSAP callback, as discussed later in "PSAP Callback to Skype for Business Clients for Dropped E9-1-1 Calls" on page 296. The ELIN table also stores the following information:
   - **ELIN**: ELIN number
   - **Time**: Time at which the original E9-1-1 call was terminated with the PSAP
   - **Count**: Number of E9-1-1 calls currently using the ELIN

An example of the ELIN database table is shown below:

<table>
<thead>
<tr>
<th>ELIN</th>
<th>Time</th>
<th>Count</th>
<th>Index</th>
<th>Call From</th>
</tr>
</thead>
<tbody>
<tr>
<td>4257275678</td>
<td>22:11:52</td>
<td>0</td>
<td>2</td>
<td>4258359333</td>
</tr>
</tbody>
</table>
The ELIN table stores this information for a user-defined period (see "Configuring the E9-1-1 Callback Timeout" on page 298), starting from when the E9-1-1 call, established with the PSAP, terminates. After this time expires, the table entry with its ELIN is disregarded and no longer used (for PSAP callback). Therefore, table entries of only the most recently terminated E9-1-1 callers are considered in the ELIN table. The maximum entries in the ELIN table is 300.

4. The device uses the ELIN number as the E9-1-1 calling number and sends it in the SIP INVITE message (as an ANI / Calling Party Number) to the SIP Trunk.

An example of a SIP INVITE message received from an E9-1-1 caller is shown below. The SIP Content-Type header indicating the PiDF-LO, and the NAM field listing the ELINs are shown in bold typeface.

```
INVITE sip:911;phone-context=Redmond@192.168.1.12;user=phone
SIP/2.0
From: "voip_911_user1"<sip:voip_911_user1@contoso.com>;epid=1D19090AED;tag=d04d65d924
To: <sip:911;phone-context=Redmond@192.168.1.12;user=phone>
CSeq: 8 INVITE
Call-ID: e6828be1-1cdd-4fb0-bdda-cda7faaf46df4
VIA: SIP/2.0/TLS 192.168.0.244:57918;branch=z9hG4bK528b7ad7
CONTACT: <sip:voip_911_user1@contoso.com;opaque=user:epid:R4bCDAUj51a06PUBkraS0QAA;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
```
17.8.3.2 Pre-empting Existing Calls for E9-1-1 Calls

If the ELIN device receives an E9-1-1 call from the IP network and there are unavailable channels (for example, all busy), the device immediately terminates one of the non-E9-1-1 calls (arbitrary) and accepts the E9-1-1 call on the freed channel:

- The preemption is done only on a call pertaining to the same source IP Group from which the E9-1-1 call is received, or the same destination IP Group (i.e., PSAP Server).

This feature is initiated only if the received SIP INVITE message contains a *Priority* header set to "emergency", as shown below:

```
PRIORITY: emergency
```

17.8.3.3 PSAP Callback to Skype for Business Clients for Dropped E9-1-1 Calls

As the E9-1-1 service automatically provides all the contact information of the E9-1-1 caller to the PSAP, the PSAP operator can call back the E9-1-1 caller. This is especially useful in cases where the caller disconnects prematurely. However, as the Enterprise sends ELINs to the PSAP for E9-1-1 calls, a callback can only reach the original E9-1-1 caller using the device to translate the ELIN number back into the E9-1-1 caller's extension number.

In the ELIN table of the device, the temporarily stored *From* header value of the SIP INVITE message originally received from the E9-1-1 caller is used for PSAP callback. When the PSAP makes a callback to the E9-1-1 caller, the device translates the called number (i.e., ELIN) received from the PSAP to the corresponding E9-1-1 caller's extension number as matched in the ELIN table.

The handling of PSAP callbacks by the device is as follows:

1. When the device receives a call from the emergency service provider, it searches the ELIN table for an ELIN that corresponds to the received called party number in the incoming message.
2. If a match is found in the ELIN table, it routes the call to the Mediation 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 device updates the Time in the ELIN table. (The Count is not affected).

The PSAP callback can be done only within a user-defined period (see "Configuring the E9-1-1 Callback Timeout" on page 298), started from after the original E9-1-1 call established with the PSAP is terminated. After this time expires, the table entry with its ELIN is disregarded and no longer used (for PSAP callback). Therefore, table entries of only the most recently terminated E9-1-1 callers are considered in the ELIN table. If the PSAP callback is done after this timeout expires, the device is unable to route the call to the E9-1-1 caller and instead, either sends it as a regular call or most likely, rejects it if there are no matching routing rules. However, if another E9-1-1 caller has subsequently been processed with the same ELIN number, the PSAP callback is routed to this new E9-1-1 caller.

In scenarios where the same ELIN number is used by multiple E9-1-1 callers, upon receipt of a PSAP callback, the device sends the call to the most recent E9-1-1 caller. For example, if the ELIN number "4257275678" is being used by three E9-1-1 callers, as shown in the table below, then when a PSAP callback is received, the device sends it to the E9-1-1 caller with phone number "4258359555".

<table>
<thead>
<tr>
<th>ELIN</th>
<th>Time</th>
<th>Call From</th>
</tr>
</thead>
<tbody>
<tr>
<td>4257275678</td>
<td>11:00</td>
<td>4258359333</td>
</tr>
<tr>
<td>4257275678</td>
<td>11:01</td>
<td>4258359444</td>
</tr>
<tr>
<td>4257275678</td>
<td>11:03</td>
<td>4258359555</td>
</tr>
</tbody>
</table>

### 17.8.3.4 Selecting ELIN for Multiple Calls within Same ERL

The device supports the receipt of up to five ELIN numbers in the XML message of each incoming SIP INVITE message. As discussed in the preceding sections, the device sends the ELIN number as the E9-1-1 calling number to the emergency service provider. If the XML message contains more than one ELIN number, the device chooses the ELIN according to the following logic:

- If the first ELIN in the list is not being used by other active calls, it chooses this ELIN.
- If the first ELIN in the list is being used by another active call, the device skips to the next ELIN in the list, and so on until it finds an ELIN that is not being used and sends this ELIN.
- If all the ELINs in the list are in use by active calls, the device selects the ELIN number as follows:
  1. The ELIN with the lowest count (i.e., lowest number of active calls currently using this ELIN).
  2. If the count between ELINs is identical, the device selects the ELIN with the greatest amount of time passed since the original E9-1-1 call using this ELIN was terminated with the PSAP. For example, if E9-1-1 caller using ELIN 4257275678 was terminated at 11:01 and E9-1-1 caller using ELIN 4257275670 was terminated at 11:03, then the device selects ELIN 4257275678.

In this scenario, multiple E9-1-1 calls are sent with the same ELIN.
17.8.4 Configuring AudioCodes ELIN Device

This section describes E9-1-1 configuration of the AudioCodes ELIN Gateway deployed in the Skype for Business environment.

17.8.4.1 Enabling the E9-1-1 Feature

By default, the ELIN device feature for E9-1-1 emergency call handling in a Skype for Business environment is disabled.

➢ To enable ELIN feature:
  - For the IP Group through which you want to communicate with the public-safety answering point (PSAP), configure the 'SBC PSAP Mode' parameter to Enable. For more information, see Configuring IP Groups on page 345.

17.8.4.2 Configuring the E9-1-1 Callback Timeout

The PSAP can use the ELIN to call back the E9-1-1 caller within a user-defined time interval (in minutes) from when the initial call established with the PSAP has been terminated. By default, an ELIN can be used for PSAP callback within 30 minutes after the call is terminated. You can change this to any value between 0 and 60:

➢ To configure the E9-1-1 callback timeout
  1. Open the Priority & Emergency page (Setup menu > Signaling & Media tab > SIP Definitions folder > Priority and Emergency).
  2. In the 'E911 Callback Timeout' field (E911CallbackTimeout), enter the required callback timeout.

Figure 17-60: Configuring E9-1-1 Callback Timeout

![Figure 17-60: Configuring E9-1-1 Callback Timeout](image)

3. Click Apply.

17.8.4.3 Configuring SBC IP-to-IP Routing Rule for E9-1-1

To route incoming E9-1-1 calls to the emergency service provider's PSAP server, you need to configure routing rules in the IP-to-IP Routing table for routing between the emergency callers' IP Group and the PSAP server's IP Group. The only special configuration is to define the emergency number (e.g., 911) in the 'Destination Username Prefix' parameter of the IP Group belonging to the E9-1-1 callers. The following example shows IP-to-IP routing rules for E9-1-1 in a Skype for Business environment:

![Figure 17-61: Example of IP-to-IP Routing Rules for Skype for Business E9-1-1](image)
17.9 Microsoft Skype for Business Presence of Third-Party Endpoints

Microsoft presence capability allows Skype for Business users to know the status (e.g., "Available" or "Do Not Disturb") of their contacts. Presence status of contacts is displayed on the user's Skype for Business endpoint. Presence information of Skype for Business endpoints (such as Skype for Business desktop client) is handled solely by the Skype for Business Server, without any intervention of the device. However, when third-party (non-Skype for Business) endpoint devices (e.g., mobile phone or PBX phone) are used by the Skype for Business users, presence status information can only be reported to the Skype for Business Server by the device. For example, if John and Alice are Skype for Business users and John makes or receives a call on a mobile device, Alice is able to see that John is in a call, even though the call is not on a native Skype for Business endpoint. Once the device reports the presence status, the Skype for Business Server sends this status change to the Skype for Business users in the network.

Note:

- Currently, the device reports the following presence status:
  - "On the Phone" - user is busy (in a call or doesn't want to be disturbed)
  - "Clear" - cancels the "On the Phone" status (returning the user's presence to its previous state)
- The feature supports Skype for Business Server 2015 and Lync Server 2013 version 5.0.8308.866 and later.

The device notifies the Skype for Business Server of a user's presence status, by using SIP PUBLISH messages. The message transactions between the device and Skype for Business Server is as follows:

1. The device routes a call between two Skype for Business users and when connected, sends a PUBLISH message with the Event header set to "presence", Expires header set to "600", Content-Type header set to "application/pidf+xml", and where the XML body's "activity" is set to "on-the-phone", as shown in the following example for user John Doe:

   PUBLISH sip:john.doe@sfb.example SIP/2.0
   From: <sip:john.doe@sfb.example>;tag=1c537837102
To: <sip:john.doe@sfb.example>
CSeq: 1 PUBLISH
Event: presence
Expires: 600
Content-Type: application/pidf+xml
Content-Length: 489
<?xml version="1.0" encoding="utf-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:ci="urn:ietf:params:xml:ns:pidf:cid"
entity="sip:john.doe@sfb.example">
<tuple id="0">
<status>
<basic>open</basic>
<ep:activities>
<ep:activity>on-the-phone</ep:activity>
</ep:activities>
</status>
</tuple>
<ci:display-name>John Doe</ci:display-name>
</presence>

2. The Skype for Business Server responds to the device with a SIP 200 OK. The message is sent with a SIP-ETag header which identifies the entity (and Expires header set to 600 seconds), as shown in the following example:

SIP/2.0 200 OK
From: "John Doe"<sip:john.doe@sfb.example>;tag=1c537837102
To: <sip:john.doe@sfb.example>;tag=0E4324A4B27040E4A167108D4FAD27E3
Call-ID: 1284896643279201635736@10.33.221.57
CSeq: 1 PUBLISH
Via: SIP/2.0/TLS
10.33.221.57;alias;...received=10.33.221.57;ms-received
port=48093;ma-received-cid=4900
SIP-ETag: 2545777538-1-1
Expires: 600
Content-Length: 0

3. If the call lasts longer than 600 seconds, the device sends another PUBLISH message with the same SIP-ETag value and with an Expires header value of 600 seconds. The Skype for Business Server responds with another 200 OK, but with a new SIP-ETag value (and Expires header set to 600 seconds). This scenario occurs for each 600-second call interval.

4. When the call ends, the device sends a PUBLISH message to cancel the user's online presence status (and the user's previous presence state is restored). The message is sent with a SIP-If-Match header set to the matching entity tag (SIP-ETag) value (i.e., SIP-ETag value of last 200 OK) and Expires header value set to "0", as shown in the following example:
PUBLISH sip:john.doe@sfb.example SIP/2.0
From: <sip:john.doe@sfb.example>;tag=lc1654434948
To: <sip:john.doe@sfb.example>
CSeq: 1 PUBLISH
Contact: <sip:john.doe@10.33.221.57;transport=tls>
Event: presence
Expires: 0
User-Agent: sur1-vg1.ecarecenters.net/v.7.20A.001.080
SIP-If-Match: 2545777538-1-1
Content-Length: 0
The following figure shows a basic illustration of the device’s integration into Microsoft Skype for Business Presence feature for third-party endpoints.
17.9.1 Configuring Skype for Business Server for Presence

On the Skype for Business Server side, you need to define the device in the Skype for Business Topology as a Trusted Application.

**Note:**
- Detailed configuration of Skype for Business Server is beyond the scope of this document.
- Before performing the below procedure, make sure that you have defined the device in the PSTN Gateway node of the Skype for Business Server Topology (using the Topology Builder).

Using the Skype for Business Server Management Shell, perform the following steps:

1. **Obtain the Site ID**
   Run the following cmdlet to retrieve the SiteId property of the site:
   ```powershell
   Get-CsSite
   ```

2. **Create a Trusted Application Pool**
   Run the following cmdlet to create a new pool to host the presence application:
   ```powershell
   New-CsTrustedApplicationPool -Identity <Pool FQDN> -Registrar <Registrar FQDN> -Site <Site Id>
   ```
   where:
   - **Identity** is the FQDN of the device, which sends the SIP PUBLISH messages with the presence status to Skype for Business Server
   - **Registrar** is the FQDN of the Registrar service for the pool
   - **Site** is the Site Id
   For example:
   ```powershell
   New-CsTrustedApplicationPool -Identity audcsbcbgw.example.com -Registrar skypepool.example.com -Site Portland
   ```

3. **Add the Trusted Application (Presence) to the Pool**
   Run the following cmdlet:
   ```powershell
   New-CsTrustedApplication-ApplicationId <String> -TrustedApplicationPoolFqdn <String> -Port <Port Number>
   ```
   where:
   - **ApplicationId** is the name of the application
   - **TrustedApplicationPoolFqdn** is the FQDN of the trusted application pool
   - **Port** is the port number on which the application will run (5061)
   For example:
   ```powershell
   New-CsTrustedApplication -ApplicationId MSpresence -TrustedApplicationPoolFqdn audcsbcbgw.example.com -Port 5061
   ```
   Make sure the port number matches the port number configured on the device.

4. **Enable and Publish the Skype for Business Server 2015 Topology**
   Run the following cmdlet to publish and enable your new topology:
   ```powershell
   Enable-CsTopology
   ```
17.9.2 Configuring the Device for Skype for Business Presence

The following procedure describes how to configure the device for notifying Skype for Business Server of presence status of Skype for Business users when making and receiving calls using third-party, endpoint devices.

To help you understand the configuration, the following lists in chronological order the main processing steps:

1. The device receives an incoming call.
2. The device uses a Call Setup Rule to perform LDAP queries on the Microsoft Active Directory to retrieve Skype for Business usernames (Request URIs) for the corresponding calling (source) and/or called (destination) number. For SBC, the Call Setup Rule is associated with the classified source IP Group (in the IP Groups table).
3. The device routes the call to the required destination, according to the normal routing rules.
4. When the call is connected, the device sends a SIP PUBLISH message to Skype for Business Server, indicating that the users' presence status is now "On-the-Phone".
5. When the call ends, the device sends another SIP PUBLISH message to the Skype for Business Server, clearing the users' "On-the-Phone" status (the presence status changes to what it was before the call was connected).

➢ To configure the device for Skype for Business presence:

1. Enable the Microsoft presence feature: open the SIP Definitions General Settings page (Setup menu > Signaling & Media tab > SIP Definitions folder > SIP Definitions General Settings), and then from the 'Enable MsPresence message' drop-down list, select Enable:

   Figure 17-62: Enabling Microsoft Presence Integration

   ![Enable MsPresence message]

2. Configure a TLS Context (TLS certificate) for secured communication (mutual authentication) between the device and the Skype for Business Server (see Configuring TLS Certificate Contexts on page 105).
3. Configure a Proxy Set to define the address of the Skype for Business Server (see Configuring Proxy Sets on page 359). Make sure you configure the following:
   - 'TLS Context Name': Assign the TLS Context that you configured in Step 2 (above).
   - 'Proxy Address': Configure the address (FQDN or IP address).
   - 'Transport Type': TLS
4. Configure an IP Group to represent the Skype for Business Server (see Configuring IP Groups on page 345). Make sure that you assign it with the Proxy Set that you configured in Step 3 (above).
5. Assign the IP Group of the Skype for Business Server as the destination (presence gateway) to where the device must send the PUBLISH messages: open the SIP Definitions General Settings page (Setup menu > Signaling & Media tab > SIP Definitions folder > SIP Definitions General Settings), and then in the 'Presence Publish IP Group ID' field, enter the IP Group ID of the Skype for Business Server that you configured in Step 4 (above):

   Figure 17-63: Assigning IP Group of Presence Gateway

   ![Presence Publish IP Group ID]
6. Configure the Skype for Business LDAP server (Active Directory) to query for the Skype for Business users’ SIP URIs (see Configuring LDAP Servers on page 240).

7. Configure Call Setup Rules to perform LDAP queries in the Microsoft Active Directory for the SIP URI of the caller (source) and called (destination) parties (see Configuring Call Setup Rules on page 384). The device first needs to search the AD for the caller or called number of the third-party endpoint device. For example, to search for a called mobile number, the searched LDAP Attribute would be "mobile" set to the value of the destination number (e.g., ‘mobile=+’ + param.call.dst.user). If the entry exists, the query searches for the Attribute (e.g., ipPhone) where the SIP URI is defined for the corresponding mobile user. If found, the query returns the Attribute's value (i.e., URI) to the device (instructed using the special 'Condition' string "presence.dst" or "presence.src"). This is the URI that the device uses as the Request-URI in the PUBLISH message that it sends to the Skype for Business Server. The configuration of the example used in this step is shown below:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Rule 1</th>
<th>Rule 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query Type</td>
<td>LDAP</td>
<td>LDAP</td>
</tr>
<tr>
<td>Search Key</td>
<td>'mobile=+’ + param.call.dst.user</td>
<td>'mobile=+’ + param.call.src.user</td>
</tr>
<tr>
<td>Attributes To Get</td>
<td>ipPhone</td>
<td>ipPhone</td>
</tr>
<tr>
<td>Condition</td>
<td>ldap.attr.ipPhone exists</td>
<td>ldap.attr.ipPhone exists</td>
</tr>
<tr>
<td>Action Subject</td>
<td>presence.dst</td>
<td>presence.src</td>
</tr>
<tr>
<td>Action Type</td>
<td>Add</td>
<td>Add</td>
</tr>
<tr>
<td>Action Value</td>
<td>ldap.attr.ipPhone</td>
<td>ldap.attr.ipPhone</td>
</tr>
</tbody>
</table>

8. Configure routing rules to route the calls in the network.

9. Configure IP Groups to represent your call party entities, and assign them the group of Call Setup Rules (Set ID) that you configured in Step 7 (above).
18 Quality of Experience

This chapter describes how to configure the Quality of Experience feature.

18.1 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.

18.1.1 Configuring the SEM Server

The device can report QoE voice metrics to a single SEM server or to two SEM servers deployed in a Geographic Redundancy, High-Availability (HA) mode. Geographic Redundancy is when each SEM/EMS server is located in a different network subnet and has its own IP address. For the device to report QoE to both servers, you need to configure the IP address of each server. For normal HA mode, when both SEM/EMS servers are located in the same subnet, a single SEM/EMS server (global, virtual) IP address is used for all network components (EMS clients and managed devices). Thus, in such a setup, you need to configure only this IP address.

You can also configure the device to use a TLS connection with the SEM server. Before you can do this, configure a TLS Context (certificates) in the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 105). If no TLS Context is specified, the device uses the default TLS Context (ID 0).

You can also configure at what stage of the call the device must send the report to the SEM server. The report can be sent during the call or only at the end of the call. Reporting at the end of the call may be beneficial when network congestion occurs, as this reduces bandwidth usage over time.

Note: If a QoE traffic overflow is experienced between SEM and the device, the device sends the QoE data only at the end of the call, regardless of your settings.

For a detailed description of the SEM parameters, see "Quality of Experience Parameters" on page 876.
To configure the SEM server address and other related features:

1. Open the Session Experience Manager page (Setup menu > Signaling & Media tab > Media folder > Quality of Experience > Session Experience Manager).

Figure 18-1: Configuring Session Experience Manager

2. Configure the address of the SEM server:
   a. In the 'Server IP' field, enter the primary SEM server's IP address.
   b. If Geographical-Redundancy HA mode exists, in the 'Redundant Server IP' field, enter the secondary SEM server's IP address.
   c. In the 'Interface Name' field, enter the device's IP network interface from which the device sends the reports to the SEM server.

3. From the 'QoE Report Mode' drop-down list, select when you want the device to send reports of a call to the SEM (during or at the end of the call).

4. (Optional) Configure a TLS connection with the SEM server:
   a. From the 'QoE Connection by TLS' drop-down list, select Enable.
   b. From the 'QoE TLS Context Name' drop-down list, select the TLS Context which defines the TLS settings (e.g., certificates).

5. Click Apply.

18.1.2 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.

18.1.3 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 757. To configure what to report to the SEM, see "Configuring Quality of Experience Profiles" on page 307.
18.2 Configuring Quality of Experience Profiles

Quality of Experience Profiles enable you to effectively monitor the quality of voice calls traversing the device in your network. Quality of Experience Profiles define severity thresholds for voice metrics monitored by the device, which if crossed can result in various actions (discussed later in the section).

Quality of Experience is configured using two tables with parent-child type relationship. The Quality of Experience Profile table is the parent, which defines the name of the Quality of Experience Profile. The Quality of Experience Color Rules table is the child, which defines severity thresholds per voice metric for the specific Quality of Experience Profile. You can configure up to 256 Quality of Experience Profiles and up to 256 Quality of Experience Color Rules.

Once configured, you can apply the Quality of Experience Profiles to specific calls (network links), by assigning them to any of the following configuration entities:

- IP Groups (see "Configuring IP Groups" on page 345)
- Media Realms (see "Configuring Media Realms" on page 319)
- Remote Media Subnets (see "Configuring Remote Media Subnets" on page 322)

The Quality of Experience Profile allows you to configure thresholds for the following monitored voice metrics:

- **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 (as displayed by SEM):

- **Green:** Indicates good call quality
- **Yellow:** Indicates fair call quality
- **Red:** Indicates poor call quality

When the threshold of a voice metric is crossed, the device changes the alarm severity and corresponding color-coded quality state of the call:

- **Minor Threshold (Yellow):** Lower threshold that indicates changes from Green or Red to Yellow.
- **Major Threshold (Red):** Higher threshold that indicates changes from Green or Yellow to Red.

The device also uses hysteresis to determine whether the threshold has indeed being crossed. Hysteresis defines the amount of fluctuation from the 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. Hysteresis is used only for threshold crossings toward a lesser severity (i.e., from Red to Yellow, Red to Green, or Yellow to Green).
The following example is used to explain how the device considers threshold crossings. The example is based on the MOS of a call, where the Major threshold is configured to 2, the Minor threshold to 4 and the hysteresis for both thresholds to 0.1:

**Figure 18-2: Threshold Crossings and Hysteresis**

![Threshold Crossings and Hysteresis](image)

**Table 18-1: Threshold Crossings based on Threshold and Hysteresis**

<table>
<thead>
<tr>
<th>Threshold Crossing</th>
<th>Calculation</th>
<th>Threshold based on Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green to Yellow (Minor alarm)</td>
<td>The change occurs if the measured metric crosses the configured Minor threshold only (i.e., hysteresis is not used).</td>
<td>4</td>
</tr>
<tr>
<td>Green to Red (Major alarm)</td>
<td>The change occurs if the measured metric crosses the configured Major threshold only (i.e., hysteresis is not used).</td>
<td>2</td>
</tr>
<tr>
<td>Yellow to Red (Major alarm)</td>
<td>The change occurs if the measured metric crosses the configured Major threshold only (i.e., hysteresis is not used).</td>
<td>2</td>
</tr>
<tr>
<td>Red to Yellow (Minor alarm)</td>
<td>The change occurs if the measured metric crosses the configured Major threshold with hysteresis configured for the Major threshold.</td>
<td>2.1 (i.e., 2 + 0.1)</td>
</tr>
<tr>
<td>Red to Green (alarm cleared)</td>
<td>The change occurs if the measured metric crosses the configured Minor threshold with hysteresis configured for the Minor threshold.</td>
<td>4.1 (i.e., 4 + 0.1)</td>
</tr>
<tr>
<td>Yellow to Green (alarm cleared)</td>
<td>The change occurs if the measured metric crosses the configured Minor threshold with hysteresis configured for the Minor threshold.</td>
<td>4.1 (i.e., 4 + 0.1)</td>
</tr>
</tbody>
</table>

Each time a 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). To configure the SEM server's address, see "Configuring the SEM Server" on page 305.
- Depending on the crossed threshold type, you can configure the device to reject calls to the destination IP Group or use an alternative IP Profile for the IP Group. For more information, see "Configuring Quality of Service Rules" on page 316.
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 519).

**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, which is used if you do not configure a Quality of Experience Profile.

The following procedure describes how to configure Quality of Experience Profiles through the Web interface. You can also configure it through other management platforms:

- **Quality of Experience Profile table:** ini file (QoEProfile) or CLI (configure voip > qoe qoe-profile)
- **Quality of Experience Color Rules table:** ini file (QOEColorRules) or CLI (configure voip > qoe qoe-color-rules)

➢ **To configure a QoE Profile:**

1. Open the Quality of Experience Profile table (Setup menu > Signaling & Media tab > Media folder > Quality of Experience > Quality of Experience Profile).
2. Click **New**; the following dialog box appears:

![Figure 18-3: Quality of Experience Profile Table - Dialog Box](image)

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

**Table 18-2: Quality of Experience Profile Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Profile Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 20 characters.</td>
</tr>
<tr>
<td>Sensitivity Level</td>
<td>Defines the pre-configured threshold profile to use. <strong>[0]</strong> User Defined = Need to define thresholds per monitored parameter in the Quality of Experience Color Rules table.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| [QOEProfile_SensitivityLevel] | ▪ [1] Low = Pre-configured low sensitivity thresholds.  
| | Reporting is done for small fluctuations in parameter values. |

5. In the Quality of Experience Profile table, select the 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 table appears.

6. Click New; the following dialog box appears:

   ![Figure 18-4: Quality of Experience Color Rules Table - Dialog Box]

7. Configure a rule according to the parameters described in the table below.

8. Click New, and then save your settings to flash memory.

Table 18-3: Quality of Experience Color Rules Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td></td>
</tr>
</tbody>
</table>
| Index | Defines an index number for the new table row.  
| [QOECOLORRULES_COLORRULEINDEX] | Note: Each row must be configured with a unique index. |
| Monitored Parameter | Defines the parameter to monitor and report.  
| [QOECOLORRULES_MONITOREDPARAM] | ▪ [0] MOS (default)  
| | ▪ [1] Delay  
| | ▪ [2] Packet Loss  
| | ▪ [4] RERL [Echo] |
| Direction | Defines the monitoring direction.  
| [QOECOLORRULES_DIRECTION] | ▪ [0] Device Side (default)  
| | ▪ [1] Remote Side |
| Sensitivity Level | Defines the sensitivity level of the thresholds.  
| [QOECOLORRULES_PROFILE] | ▪ [0] User Defined = Need to define the thresholds in the parameters described below.  
| | ▪ [1] Low = Pre-configured low sensitivity threshold values. |
### 18. Quality of Experience

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Thus, reporting is done only if changes in parameters' values are significant.</td>
</tr>
<tr>
<td></td>
<td>• [3] High = Pre-configured high sensitivity threshold values. Thus, reporting is done for small fluctuations in parameter values.</td>
</tr>
</tbody>
</table>

#### Thresholds

**Minor Threshold (Yellow)**

Defines the Minor threshold value, which is the lower threshold located between the Yellow and Green states. To consider a threshold crossing:

- **Increase in severity (i.e., Green to Yellow):** Only this value is used.
- **Decrease in severity (Red to Green, or Yellow to Green):** This value is used with the hysteresis, configured by the 'Minor Hysteresis (Yellow)' parameter (see below).

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.

**Minor Hysteresis (Yellow)**

Defines the amount of fluctuation (hysteresis) from the Minor threshold, configured by the 'Minor Threshold (Yellow)' parameter in order for the threshold to be considered as crossed. The hysteresis is used only to determine threshold crossings to Green (i.e., from Yellow to Green, or Red to Green). In other words, the device considers a threshold crossing to Green only if the measured voice metric crosses the Minor threshold and the hysteresis.

For example, if you configure the 'Minor Threshold (Yellow)' parameter to 4 and the 'Minor Hysteresis (Yellow)' parameter to 0.1 (for MOS), the device considers a threshold crossing to Green only if the MOS crosses 4.1 (i.e., 4 + 0.1).

**Major Threshold (Red)**

Defines the Major threshold value, which is the upper threshold located between the Yellow and Red states. To consider a threshold crossing:

- **Increase in severity (i.e., Yellow to Red):** Only this value is used.
- **Decrease in severity (Red to Yellow):** This value is used with the hysteresis, configured by the 'Major Hysteresis (Red)' parameter (see below).

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 (%).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jitter is in msec.</td>
<td></td>
</tr>
<tr>
<td>Echo measures the Residual Echo Return Loss (RERL) in dB.</td>
<td></td>
</tr>
<tr>
<td>Major Hysteresis (Red) major-hysteresis-red [QOEColorRules_MajorHysteresis]</td>
<td>Defines the amount of fluctuation (hysteresis) from the Major threshold, configured by the 'Major Threshold (Red)' parameter in order for the threshold to be considered as crossed. The hysteresis is used only to determine threshold crossings from Red to Yellow. In other words, the device considers a threshold crossing to Yellow only if the measured voice metric crosses the Major threshold and the hysteresis. For example, if you configure the 'Major Threshold (Red)' parameter to 2 and the 'Major Hysteresis (Red)' parameter to 0.1 (for MOS), the device considers a threshold crossing to Yellow only if the MOS crosses 2.1 (i.e., 2 + 0.1).</td>
</tr>
</tbody>
</table>

### 18.3 Configuring Bandwidth Profiles

The Bandwidth Profile table lets you configure up to 1,884 Bandwidth Profiles. A Bandwidth Profile defines bandwidth utilization thresholds for audio and/or video traffic (incoming and outgoing), which if crossed can result in various actions (discussed later in the section). Bandwidth Profiles enhance the device's monitoring of bandwidth utilization.

Once configured, you can apply Bandwidth Profiles to specific calls, by assigning them to any of the following configuration entities:

- IP Groups (see "Configuring IP Groups" on page 345)
- Media Realms (see "Configuring Media Realms" on page 319)
- Remote Media Subnets (see "Configuring Remote Media Subnets" on page 322)

Each time a configured bandwidth threshold is crossed, the device can do the following, depending on configuration:

- Reject calls destined to the IP Group or use an alternative IP Profile for the IP Group. For more information, see "Configuring Quality of Service Rules" on page 316.
- Use an alternative routing rule for alternative routing. If a call is rejected due to a crossed threshold, the device generates a SIP 806 response. You can configure the SIP response code as a reason for alternative routing (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 519).
- Send an SNMP alarm (acMediaRealmBWThresholdAlarm). The device clears the alarm when bandwidth utilization returns to normal (Green).

The SEM displays bandwidth utilization using color-coded states:

- **Green**: Indicates bandwidth utilization is within normal range.
- **Yellow**: Indicates bandwidth utilization is encroaching on "total" bandwidth, serving as a warning (or it could also mean that bandwidth utilization has dropped below the red state).
- **Red**: Indicates that bandwidth utilization has exceeded total bandwidth.

Bandwidth Profiles let you configure bandwidth thresholds, which when crossed changes the color-coded state for bandwidth utilization:

- **Green-Yellow (Minor) Threshold**: Lower threshold configured as a percentage of the configured major (total) bandwidth threshold. When bandwidth goes over the threshold, the device considers it a Yellow state (Minor alarm severity); when it goes below the threshold, it considers it a Green state (cleared alarm).
Yellow-Red (Major) Threshold: Upper threshold configured by the major (total) bandwidth threshold. When bandwidth goes over the threshold, the device considers it a Red state (Major alarm severity); when it goes below the threshold, it considers it a Yellow state (Minor alarm severity).

The device also uses hysteresis to determine whether the threshold has indeed being crossed. Hysteresis defines the amount of fluctuation from the 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. Hysteresis is used only for threshold crossings toward a lesser severity (i.e., from Red to Yellow, Red to Green, or Yellow to Green). Hysteresis is configured as a percentage of the configured major (total) bandwidth threshold.

The following example is used to explain how the device considers threshold crossings. The example is based on a setup where the Major (total) bandwidth threshold is configured to 64,000 Kbps, the Minor threshold to 50% (of the total) and the hysteresis to 10% (of the total):

![Figure 18-5: Bandwidth Threshold Crossings](image)

<table>
<thead>
<tr>
<th>Threshold Crossing</th>
<th>Calculation</th>
<th>Threshold based on Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green to Yellow (Minor alarm)</td>
<td>The change occurs if the current bandwidth crosses the configured Minor threshold only (i.e., hysteresis is not used).</td>
<td>32,000 Kbps</td>
</tr>
<tr>
<td>Green to Red (Major alarm)</td>
<td>The change occurs if the current bandwidth crosses the configured Major threshold only (i.e., hysteresis is not used).</td>
<td>64,000 Kbps</td>
</tr>
<tr>
<td>Yellow to Red (Major alarm)</td>
<td>The change occurs if the current bandwidth crosses the configured Major threshold only (i.e., hysteresis is not used).</td>
<td>64,000 Kbps</td>
</tr>
<tr>
<td>Red to Yellow (Minor alarm)</td>
<td>The change occurs if the current bandwidth crosses the configured Major threshold with hysteresis.</td>
<td>57,600 Kbps [64,000 - (10% x 64,000)]</td>
</tr>
<tr>
<td>Yellow to Green (alarm cleared)</td>
<td>The change occurs if the current bandwidth crosses the configured Minor threshold with hysteresis.</td>
<td>25,600 Kbps [32,000 - (10% x 64,000)]</td>
</tr>
<tr>
<td>Red to Green (alarm cleared)</td>
<td>The change occurs if the current bandwidth crosses the configured Minor threshold with hysteresis.</td>
<td>25,600 Kbps [32,000 - (10% x 64,000)]</td>
</tr>
</tbody>
</table>
The following procedure describes how to configure Bandwidth Profiles through the Web interface. You can also configure it through ini file (BWProfile) or CLI (configure voip > qoe bw-profile).

**To configure a Bandwidth Profile:**

1. Open the Bandwidth Profile table (Setup menu > Signaling & Media tab > Media folder > Quality of Experience > Bandwidth Profile).
2. Click New; the following dialog box appears:

   ![Figure 18-6: Bandwidth Profile Table - Dialog Box](image)

3. Configure a rule according to the parameters described in the table below.
4. Click Apply, and then reset the device with a save to flash memory.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index [BWProfile_Index]</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name [BWProfile_Name]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 20 characters.</td>
</tr>
<tr>
<td>Egress Audio Bandwidth egress-audio-bandwidth [BWProfile_EgressAudioBandwidth]</td>
<td>Defines the major (total) threshold for outgoing audio traffic (in Kbps).</td>
</tr>
<tr>
<td>Ingress Audio Bandwidth ingress-audio-bandwidth [BWProfile_IngressAudioBandwidth]</td>
<td>Defines the major (total) threshold for incoming audio traffic (in Kbps).</td>
</tr>
<tr>
<td>Egress Video Bandwidth</td>
<td>Defines the major (total) threshold for outgoing video traffic (in Kbps).</td>
</tr>
</tbody>
</table>
### 18. Quality of Experience

#### Parameter | Description
--- | ---
egress-video-bandwidth [BWProfile_EgressVideoBandwidth] | Defines the major (total) threshold for incoming video traffic (in Kbps).

Ingress Video Bandwidth

*ingress-video-bandwidth [BWProfile_IngressVideoBandwidth]* | Defines the major (total) threshold for incoming video traffic (in Kbps).

Total Egress Bandwidth

total-egress-bandwidth [BWProfile_TotalEgressBandwidth] | Defines the major (total) threshold for video and audio outgoing bandwidth (in Kbps).

Total Ingress Bandwidth

total-ingress-bandwidth [BWProfile_TotalIngressBandwidth] | Defines the major (total) threshold for video and audio incoming bandwidth (in Kbps).

#### Thresholds

**Minor Threshold**

*minor-threshold [BWProfile_MinorThreshold]* | Defines the Minor threshold value, which is the lower threshold located between the Yellow and Green states. The parameter is configured as a percentage of the major (total) bandwidth threshold (configured by the above bandwidth parameters). For example, if you configure the parameter to 50 and the 'Egress Audio Bandwidth' parameter to 64,000, the Minor threshold for outgoing audio bandwidth is 32,000 (i.e., 50% of 64,000).

To consider a threshold crossing:

- **Increase in severity (i.e., Green to Yellow):** Only this value is used.
- **Decrease in severity (Red to Yellow, or Yellow to Green):** This value is used with the hysteresis, configured by the 'Hysteresis' parameter (see below).

**Note:** The parameter applies to all your configured bandwidths.

**Hysteresis**

*hysteresis [BWProfile_Hysteresis]* | Defines the amount of fluctuation (hysteresis) from the configured bandwidth threshold in order for the threshold to be considered as crossed (i.e., avoids false reports of threshold crossings). The hysteresis is used only to determine threshold crossings when severity is reduced (i.e., from Red to Yellow, Yellow to Green, or Red to Green). The parameter is configured as a percentage of the Major (total) bandwidth threshold.

For example, if you configure the parameter to 10 and the 'Egress Audio Bandwidth' parameter to 64,000, the hysteresis is 6,400 (10% of 64,000) and threshold crossings are considered at the following bandwidths:

- Red-to-Yellow (Yellow-Minor alarm severity): 57,600 Kbps [64,000 - (10% x 64,000)]
- Yellow-to-Green (Green-alarm cleared): 25,600 Kbps [32,000 - (10% x 64,000)]

**Generate Alarm**

*generate-alarms [BWProfile_GenerateAlarms]* | Enables the device to send an SNMP alarm if a bandwidth threshold is crossed.

- [0] Disable (default)
- [1] Enable
18.4 Configuring Quality of Service Rules

The Quality of Service Rules table lets you configure up to 7,500 Quality of Service rules. A Quality of Service rule defines an action to perform when the threshold (major or minor) of a specific performance monitoring call metric is crossed for a specific IP Group. The call metric can be voice quality (i.e., MOS), bandwidth, Answer-seizure ratio (ASR), Network Effectiveness Ratio (NER), or Average Call Duration (ACD).

Depending on the call metric, you can configure the following actions to be performed if the threshold is crossed:

- Reject calls to the IP Group for a user-defined duration.
  
  Rejection of calls can also trigger alternative routing. When the device rejects a call due to an ASR, NER or ACD threshold crossing, it generates the SIP response code, 850 (Signaling Limits Exceeded). When the device rejects a call due to Voice Quality and Bandwidth threshold crossing, it generates the SIP response code, 806 (Media Limits Exceeded). If you configure these SIP response codes in the Alternative Routing Reasons table (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 519) and the device rejects a call, it searches in the IP-to-IP Routing table for an alternative routing rule.

  When the device rejects calls to an IP Group based on a Quality of Service rule, it raises an SNMP alarm (aclpGroupNoRouteAlarm). The alarm is also raised upon a keep-alive failure with the IP Group. For more information, refer to the SNMP Reference Guide.

- Use a different IP Profile for the IP Group or current call. This action can be useful, for example, when poor quality occurs due to packet loss and the device can then switch to an IP Profile configured with a higher RTP redundancy level or lower bit-rate coder.

To learn more about which actions are supported per call metric, see the description of the 'Rule Action' parameter below.

To configure thresholds, see the following sections:

- Voice Quality (MOS) - "Configuring Quality of Experience Profiles" on page 307
- Bandwidth - "Configuring Bandwidth Profiles" on page 312
- ASR, ACD and NER - "Configuring Performance Profiles" on page 743

The following procedure describes how to configure Quality of Service rules through the Web interface. You can also configure it through ini file (QualityOfServiceRules) or CLI (configure voip > qoe quality-of-service-rules).

1. **To configure a Quality of Service rule:**
   
   Open the Quality of Service Rules table (Setup menu > Signaling & Media tab > Media folder > Quality of Service Rules).
2. Click **New**; the following dialog box appears:

![Figure 18-7: Quality of Service Rules Table - Dialog Box](image)

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

**Table 18-6: Quality of Service Rules Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Match</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>IP Group</td>
<td>Assigns an IP Group. The rule applies to all calls belonging to the IP Group.</td>
</tr>
<tr>
<td>Rule Metric</td>
<td>Defines the performance monitoring call metric to which the rule applies if the metric’s threshold is crossed.</td>
</tr>
<tr>
<td>Severity</td>
<td>Defines the alarm severity level. When the configured severity occurs, the device performs the action of the rule.</td>
</tr>
<tr>
<td><strong>0</strong> Major (Default)</td>
<td></td>
</tr>
<tr>
<td><strong>1</strong> Minor</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If you configure the 'Rule Metric' parameter to **ACD**, **ASR** or **NER**...
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NER</td>
<td>you must configure the parameter to Major. For all other 'Rule Metric' parameter values, you can configure the parameter to any value.</td>
</tr>
</tbody>
</table>

**Action**

<table>
<thead>
<tr>
<th>Rule Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rule-action</td>
<td>Defines the action to be done if the rule is matched.</td>
</tr>
<tr>
<td>[QualityOfServiceRules_RuleAction]</td>
<td>- [0] Reject Calls = (Default) New calls destined to the specified IP Group are rejected for a user-defined duration. To configure the duration, use the 'Calls Reject Duration' parameter (see below).</td>
</tr>
<tr>
<td></td>
<td>- [1] Alternative IP Profile = A different IP Profile is used for the IP Group or call (depending on the 'Rule Metric' parameter). To specify the IP Profile, use the 'Alternative IP Profile Name' parameter (see below).</td>
</tr>
</tbody>
</table>

**Note:**
- If you configure the 'Rule Metric' parameter to ACD, ASR or NER, you must configure the parameter to Reject Calls.
- If you configure the 'Rule Metric' parameter to Voice Quality or Bandwidth:
  - If you configure the 'Severity' parameter to Minor, you must configure the parameter to Alternative IP Profile.
  - If you configure the 'Severity' parameter to Major, you can configure the parameter to any option.

When configured to Alternative IP Profile and the threshold is crossed, the device changes the IP Profile for the entire IP Group for all new calls.
- If you configure the 'Rule Metric' parameter to Poor InVoice Quality, you must configure the parameter to Alternative IP Profile. If the threshold is crossed (i.e., poor call quality), the device changes the IP Profile for the specific call only (during the call).

<table>
<thead>
<tr>
<th>Calls Reject Duration</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>calls-reject-duration</td>
<td>Defines the duration (in minutes) for which the device rejects calls to the IP Group if the rule is matched.</td>
</tr>
<tr>
<td>[QualityOfServiceRules_CallsRejectDuration]</td>
<td>The default is 5.</td>
</tr>
</tbody>
</table>

**Note:** The parameter is applicable only if the 'Rule Action' parameter is configured to Reject Calls.

<table>
<thead>
<tr>
<th>Alternative IP Profile Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alt-ip-profile-name</td>
<td>Assigns a different IP Profile to the IP Group or call (depending on the 'Rule Metric' parameter) if the rule is matched.</td>
</tr>
<tr>
<td>[QualityOfServiceRules_AltIPProfileName]</td>
<td>By default, no value is defined.</td>
</tr>
</tbody>
</table>

**Note:** The parameter is applicable only if the 'Rule Action' parameter is configured to Alternative IP Profile.
19 Control Network

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

19.1 Configuring Media Realms

The Media Realms table lets you configure a pool of up to 1,024 SIP media interfaces, termed Media Realms. Media Realms lets you divide a Media-type interface (configured in the IP Interfaces table) into several media realms, where each realm is specified by a UDP port range. Media Realms also define the maximum number of permitted media sessions.

Once configured, to apply Media Realms to specific calls, you need to assign them to any of the following configuration entities:

- IP Groups (see "Configuring IP Groups" on page 345)
- SIP Interfaces (see "Configuring SIP Interfaces" on page 337)

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. To configure Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 307.
- 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. To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 312.

The Media Realms table provides sub-tables ("child" tables) that let you configure the following:

- Remote Media Subnets: Defines remote destination subnets per Media Realm and assigns each subnet a Quality of Experience Profile and Bandwidth Profile. For more information, see "Configuring Remote Media Subnets" on page 322.
- Media Realm Extensions: Defines port ranges for multiple Media-type interfaces per Media Realm. For more information, see "Configuring Media Realm Extensions" on page 325.

Note:

- The Media Realm assigned to an IP Group overrides any other Media Realm assigned to any other configuration entity associated with the call.
- If you modify a Media Realm that is currently being used by a call, the device does not perform Quality of Experience for the call.
- If you delete a Media Realm that is currently being used by a call, the device maintains the call until the call parties end the call.
- The device provides a preconfigured Media Realm ("DefaultRealm") in the Media Realms table, which can be modified or deleted.

The following procedure describes how to configure Media Realms through the Web interface. You can also configure it through ini file (CpMediaRealm) or CLI (configure voip > realm).

➢ To configure a Media Realm:

1. Open the Media Realms table (Setup menu > Signaling & Media tab > Core Entities folder > Media Realms).
2. Click **New**; the following dialog box appears:

**Figure 19-1: Media Realms Table - Add Dialog Box**

3. Configure the Media Realm according to the parameters described in the table below.

4. Click **Apply**.

**Table 19-1: Media Realms table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[CpMediaRealm_Index]</td>
<td><strong>Note</strong>: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td>[CpMediaRealm_MediaRealmName]</td>
<td>The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>- The parameter is mandatory.</td>
</tr>
<tr>
<td></td>
<td>- Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>Topology Location</td>
<td>Defines the display location of the Media Realm in the Topology view.</td>
</tr>
<tr>
<td>topology-location</td>
<td>[CpMediaRealm_TopologyLocation]</td>
</tr>
<tr>
<td></td>
<td>- [0] Down = (Default) The Media Realm element is displayed on the lower border of the view.</td>
</tr>
<tr>
<td></td>
<td>- [1] Up = The Media Realm element is displayed on the upper border of the view.</td>
</tr>
<tr>
<td></td>
<td>For more information on the Topology view, see &quot;Building and Viewing SIP Entities in Topology View&quot; on page 369.</td>
</tr>
<tr>
<td>IPv4 Interface Name</td>
<td>Assigns an IPv4 network interface to the Media Realm.</td>
</tr>
<tr>
<td>ipv4</td>
<td>[CpMediaRealm_IPv4IF]</td>
</tr>
<tr>
<td></td>
<td>By default, no value is defined.</td>
</tr>
<tr>
<td></td>
<td>To configure IP network interfaces, see &quot;Configuring IP Network Interfaces&quot; on page 137.</td>
</tr>
</tbody>
</table>
### 19. Control Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IPv6 Interface Name</strong></td>
<td>Assigns an IPv6 network interface to the Media Realm. By default, no value is defined. To configure IP network interfaces, see Configuring IP Network Interfaces on page 137.</td>
</tr>
<tr>
<td>ipv6if</td>
<td></td>
</tr>
<tr>
<td>[CpMediaRealm_IPv6IF]</td>
<td></td>
</tr>
<tr>
<td><strong>Port Range Start</strong></td>
<td>Defines the starting port for the range of media interface UDP ports. By default, no value is defined.</td>
</tr>
<tr>
<td>port-range-start</td>
<td></td>
</tr>
<tr>
<td>[CpMediaRealm_PortRangeStart]</td>
<td></td>
</tr>
<tr>
<td><strong>Number of Media Session Legs</strong></td>
<td>Defines the number of media sessions for the configured port range. By default, no value is defined.</td>
</tr>
<tr>
<td>session-leg</td>
<td></td>
</tr>
<tr>
<td>[CpMediaRealm_MediaSessionLeg]</td>
<td></td>
</tr>
<tr>
<td><strong>Port Range End</strong></td>
<td>(Read-only field) Displays the ending port for the range of media interface UDP ports. The device automatically populates the parameter with a value, calculated by the summation of the 'Port Range Start' parameter and 'Number of Media Session Legs' parameter (multiplied by the port chunk size) minus 1: start port + (sessions * port spacing) - 1</td>
</tr>
<tr>
<td>port-range-end</td>
<td></td>
</tr>
<tr>
<td>[CpMediaRealm_PortRangeEnd]</td>
<td></td>
</tr>
<tr>
<td><strong>Default Media Realm</strong></td>
<td>Defines the Media Realm as the default Media Realm. The</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• You must either configure all your Media Realms with port ranges or all without; not some with and some without.</td>
</tr>
<tr>
<td></td>
<td>• The available UDP port range is according to the BaseUDPport parameter. For more information, see &quot;Configuring RTP Base UDP Port&quot; on page 195.</td>
</tr>
<tr>
<td></td>
<td>• The port must be different from ports configured for SIP traffic (i.e., ports configured for SIP Interfaces). For example, if the RTP port range is 6000 to 6999, the SIP port can be less than 6000 or greater than 6999.</td>
</tr>
<tr>
<td></td>
<td>• Media Realms must not have overlapping port ranges.</td>
</tr>
</tbody>
</table>

For example, a port starting at 6,000, 5 sessions and 10 port spacing:

\[
6,000 + (5 \times 10) - 1 = 6,000 + (50) - 1 = 6,049
\]

The device allocates the UDP ports for RTP, RTCP and T.38 traffic per leg in "jumps" (spacing) of 4, 5 or 10, configured by the UdpPortSpacing parameter. For example, if the port range starts at 6000 and the UDP port spacing is 10, the available ports include 6000, 6010, 6020, 6030, and so on (depending on number of media sessions).

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 configuring the T38UseRTPPort parameter to 1.

For more information on local UDP port range, see "Configuring RTP Base UDP Port" on page 195.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| is-default [CpMediaRealm_IsDefault] | default Media Realm is used for SIP Interfaces and IP Groups for which you have not assigned a Media Realm.  
▪ [0] No (default)  
▪ [1] Yes  
Note:  
▪ You can configure the parameter to **Yes** for only **one** Media Realm; all the other Media Realms must be configured to **No**.  
▪ If you do not configure the parameter (i.e., the parameter is **No** for all Media Realms), the device uses the first Media Realm in the table as the default.  
▪ If the table is not configured, the default Media Realm includes all configured media interfaces. |

### Quality of Experience

| QoE Profile | Assigns a QoE Profile to the Media Realm.  
| qoe-profile [CpMediaRealm_QoeProfile] | By default, no value is defined.  
To configure QoE Profiles, see "Configuring Quality of Experience Profiles" on page 307. |
| BW Profile | Assigns a Bandwidth Profile to the Media Realm.  
| bw-profile [CpMediaRealm_BWProfile] | By default, no value is defined.  
To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 312. |

### 19.1.1 Configuring Remote Media Subnets

Remote Media Subnets define destination subnets for media (RTP/SRTP) traffic on a specific Media Realm. Each Remote Media Subnet can be assigned different call quality (Quality of Experience Profile) and bandwidth utilization (Bandwidth Profile) profiles. These profiles are configured in "Configuring Quality of Experience Profiles" on page 307 and "Configuring Bandwidth Profiles" on page 312, 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 19-2: Remote Media Subnets Example**

The following procedure describes how to configure Remote Media Subnets through the Web interface. You can also configure it through ini file (RemoteMediaSubnet) or CLI (configure voip > remote-media-subnet).

➢ **To configure a Remote Media Subnet:***

1. Open the Media Realms table (see "Configuring Media Realms" on page 319).
2. Select the Media Realm row for which you want to add Remote Media Subnets, and then click the **Remote Media Subnet** link located below the table; the Remote Media Subnet table appears.
3. Click **New**; the following dialog box appears:

   **Figure 19-3: Remote Media Subnet Table - Add Dialog Box**

4. Configure the Remote Media Subnet according to the parameters described in the table below.

5. Click **Apply**.

   **Table 19-2: Remote Media Subnet Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td>Defines an index number for the new table row. &lt;br&gt;<strong>Note</strong>: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Defines a descriptive name, which is used when associating the row in other tables. &lt;br&gt;The valid value is a string of up to 20 characters. &lt;br&gt;<strong>Note</strong>: Each row must be configured with a unique name.</td>
</tr>
<tr>
<td><strong>Prefix Length</strong></td>
<td>Defines the subnet mask in Classless Inter-Domain Routing (CIDR) notation. For example, 16 denotes 255.255.0.0. &lt;br&gt;The default is 16.</td>
</tr>
<tr>
<td><strong>Address Family</strong></td>
<td>Defines the IP address protocol. &lt;br&gt;• [2] IPv4 (default) &lt;br&gt;• [10] IPv6</td>
</tr>
<tr>
<td><strong>Destination IP</strong></td>
<td>Defines the IP address of the destination. &lt;br&gt;The default is 0.0.0.0.</td>
</tr>
<tr>
<td><strong>QoE Profile</strong></td>
<td>Assigns a Quality of Experience Profile to the Remote Media Subnet. &lt;br&gt;By default, no value is defined. &lt;br&gt;To configure QoE Profiles, see &quot;Configuring Quality of Experience Profiles&quot; on page 307.</td>
</tr>
</tbody>
</table>
19. Control Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BW Profile</td>
<td>Assigns a Bandwidth Profile to the Remote Media Subnet. By default, no value is defined.</td>
</tr>
<tr>
<td>bw-profile</td>
<td>To configure Bandwidth Profiles, see &quot;Configuring Bandwidth Profiles&quot; on page 312.</td>
</tr>
<tr>
<td>[RemoteMediaSubnet_BWProfileName]</td>
<td></td>
</tr>
</tbody>
</table>

19.1.2 Configuring Media Realm Extensions

The Media Realm Extension table lets you configure 5 Media Realm Extensions. A Media Realm Extension is associated with a specific Media Realm and defines a port range and the number of media sessions for a specific Media-type network interface (configured in the IP Interfaces table). Therefore, a Media Realm Extension enhances a Media Realm by allowing you to define different port ranges, media sessions, and network interface than is defined by the associated Media Realm (i.e., the Media Realm is distributed across multiple interfaces).

Media Realm Extensions can be useful, for example, to overcome limitations of the maximum number of media ports supported per interface. Instead of configuring only a single Media Realm in the Media Realms table (see "Configuring Media Realms" on page 319), you can also configure additional "Media Realms" in the Media Realm Extensions table associated with the single Media Realm. An IP Group that is associated with a Media Realm configured with Media Realm Extensions, allocates its media sessions / ports between the different interfaces, as configured by the Media Real and its associated Media Realm Extensions. For example, two Media Realm Extensions could be configured, whereby one allocates 25 media sessions on interface "LAN-1" and another, 10 sessions on interface "LAN-2". The Media Realm associated with these Media Realm Extensions would be assigned to the relevant IP Group.

Figure 19-4: Example of Implementation of Media Realm Extensions

The following procedure describes how to configure Media Realm Extensions through the Web interface. You can also configure it through ini file (MediaRealmExtension) or CLI (configure voip > voip-network realm-extension).

➢ To configure a Media Realm Extension:

1. Open the Media Realms table (see "Configuring Media Realms" on page 319).
2. Select the Media Realm for which you want to add Remote Media Extensions, and then click the Media Realm Extension link located below the table; the Media Realm Extension table appears.
3. Click **New**; the following dialog box appears:

![Figure 19-5: Media Realm Extension Table - Add Dialog Box](image)

4. Configure the Media Realm Extension according to the parameters described in the table below.

5. Click **Apply**.

**Table 19-3: Media Realm Extension Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
</tbody>
</table>
| IPv4 Interface Name | Assigns an IPv4 network interface (configured in the IP Interfaces table) to the Media Realm Extension. By default, no value is defined. To configure IP network interfaces, see "Configuring IP Network Interfaces" on page 137. **Note:**  
- The parameter is mandatory.  
- You must configure the Media Realm Extension with an IP network interface that has the same IP version(s) as the Media Realm to which the Media Realm Extension is associated. If the associated Media Realm is assigned both an IPv4 and IPv6 network interface, you also need to assign the Media Realm Extension with both an IPv4 and IPv6 network interface. For example, if the associated Media Realm is assigned only an IPv4 network interface, you also need to assign the Media Realm Extension with an IPv4 network interface. |
| IPv6 Interface Name | Assigns an IPv6 network interface (configured in the IP Interfaces table) to the Media Realm Extension. By default, no value is defined. **Note:**  
- The parameter is mandatory.  
- You must configure the Media Realm Extension with an IP network interface that has the same IP version(s) as the Media Realm to which the Media Realm Extension is associated. If the associated Media Realm is assigned both an IPv4 and IPv6 network interface, you also need to assign the Media Realm Extension with both an IPv4 and IPv6 network interface. For example, if the associated Media Realm is assigned only an IPv4 network interface, you also need to assign the Media Realm Extension with an IPv4 network interface. |
### 19. Control Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>version(s) as the Media Realm to which the Media Realm Extension is associated. If the associated Media Realm is assigned both an IPv4 and IPv6 network interface, you also need to assign the Media Realm Extension with both an IPv4 and IPv6 network interface. For example, if the associated Media Realm is assigned only an IPv6 network interface, you also need to assign the Media Realm Extension with an IPv6 network interface.</td>
</tr>
<tr>
<td>Port Range Start</td>
<td>Defines the first (lower) port in the range of media UDP ports for the Media Realm Extension.</td>
</tr>
<tr>
<td>[MediaRealmExtension_PortRangeStart]</td>
<td>By default, no value is defined.</td>
</tr>
<tr>
<td>Notes:</td>
<td>• You must either configure all your Media Realms with port ranges or all without; not some with and some without.</td>
</tr>
<tr>
<td></td>
<td>• The available UDP port range is according to the BaseUDPport parameter. For more information, see &quot;Configuring RTP Base UDP Port&quot; on page 195.</td>
</tr>
<tr>
<td></td>
<td>• The port range must not overlap with any other media port range that you have configured in the table or in the Media Realms table.</td>
</tr>
<tr>
<td>Port Range End</td>
<td>Defines the last (upper) port in the range of media UDP ports for the Media Realm Extension.</td>
</tr>
<tr>
<td>[MediaRealmExtension_PortRangeEnd]</td>
<td>Note: It is unnecessary to configure the parameter. The device automatically populates the parameter with a value, calculated by the summation of the 'Number of Media Session Legs' parameter (multiplied by the port chunk size) and the 'Port Range Start' parameter. After you have added the Media Realm Extension row to the table, the parameter is displayed with the calculated value.</td>
</tr>
<tr>
<td>Number Of Media Session Legs</td>
<td>Defines the number of media sessions for the port range. For example, 100 ports correspond to 10 media sessions, since ports are allocated in chunks of 10.</td>
</tr>
<tr>
<td>[MediaRealmExtension_MediaSessionLeg]</td>
<td>By default, no value is defined.</td>
</tr>
<tr>
<td>Note:</td>
<td>The parameter is mandatory.</td>
</tr>
</tbody>
</table>

### 19.2 Configuring SRDs

The SRDs table lets you configure up to 600 signaling routing domains (SRD). The SRD is a logical representation of an entire SIP-based VoIP network (Layer 5) consisting of groups of SIP users and servers. The SRD is associated with all the configuration entities (e.g., SIP Interfaces and IP Groups) required for routing calls within the network. Typically, only a single SRD is required (recommended) for most deployments. Multiple SRDs are only required for multi-tenant deployments, where the physical device is "split" into multiple logical devices. For more information on multi-tenant architecture, see "Multiple SRDs for Multi-tenant Deployments" on page 333.

As the device is shipped with a default SRD ("DefaultSRD" at Index 0), if your deployment requires only one SRD, you can use the default SRD instead of creating a new one. When
only one SRD is employed and you create other related configuration entities (e.g., SIP Interfaces), the default SRD is automatically assigned to the new configuration entity. Therefore, when employing a single-SRD configuration topology, there is no need to handle SRD configuration (i.e., transparent).

SRDs are associated with the following configuration entities:

- **SIP Interface** (mandatory) - see "Configuring SIP Interfaces" on page 337
- **IP Group** (mandatory) - see "Configuring IP Groups" on page 345
- **Proxy Set** (mandatory) - see "Configuring Proxy Sets" on page 359
- **Admission Control rule** - see Configuring Admission Control Table on page 489
- **Classification rule** - see Configuring Classification Rules on page 493

As mentioned previously, if you use only a single SRD, the device automatically assigns it to the above-listed configuration entities.

As each SIP Interface defines a different Layer-3 network (see "Configuring SIP Interfaces" on page 337 for more information) on which to route or receive calls and as you can assign multiple SIP Interfaces to the same SRD, for most deployment scenarios (even for multiple Layer-3 network environments), you only need to employ a single SRD to represent your VoIP network (Layer 5). For example, if your VoIP deployment consists of an Enterprise IP PBX (LAN), a SIP Trunk (WAN), and far-end users (WAN), you would only need a single SRD. The single SRD would be assigned to three different SIP Interfaces, where each SIP Interface would represent a specific Layer-3 network (IP PBX, SIP Trunk, or far-end users) in your environment. The following figure provides an example of such a deployment:

**Figure 19-6: Deployment using a Single SRD**
Note:

- It is recommended to use a single-SRD configuration topology, unless you are deploying the device in a multi-tenant environment, in which case multiple SRDs are required.
- Each SIP Interface, Proxy Set, and IP Group can be associated with only one SRD.
- If you have upgraded your device to Version 7.0 and your device was configured with multiple SRDs but not operating in a multi-tenant environment, it is recommended to gradually change your configuration to a single SRD topology.
- If you upgrade the device from an earlier release to Version 7.0, your previous SRD configuration is fully preserved regarding functionality. The same number of SRDs is maintained, but the configuration elements are changed to reflect the configuration topology of Version 7.0. Below are the main changes in configuration topology when upgrading to Version 7.0:
  - The SIP Interface replaces the associated SRD in several tables (due to support for multiple SIP Interfaces per SRD).
  - Some fields in the SRDs table were duplicated or moved to the SIP Interfaces table.
  - Indices used for associating configuration entities in tables are changed to row pointers (using the entity's name).
  - Some tables are now associated (mandatory) with an SRD (SIP Interface, IP Group, Proxy Set, and Classification).
  - Some fields used for associating configuration entities in tables now have a value of Any to distinguish between Any and None (deleted entity or not associated).

The following procedure describes how to configure SRDs through the Web interface. You can also configure it through ini file (SRD) or CLI (configure voip > srd).

To configure an SRD:

1. Open the SRDs table (Setup menu > Signaling & Media tab > Core Entities folder > SRDs).
2. Click New; the following dialog box appears:

![Figure 19-7: SRDs Table - Add Dialog Box]

3. Configure an SRD according to the parameters described in the table below.
4. Click **Apply**.

### Table 19-4: SRDs table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index for the new table row.</td>
</tr>
<tr>
<td>[SRD_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td>name [SRD_Name]</td>
<td>The valid value can be a string of up to 40 characters.</td>
</tr>
<tr>
<td><strong>Sharing Policy</strong></td>
<td></td>
</tr>
<tr>
<td>type [SRD_SharingPolicy]</td>
<td>Defines the sharing policy of the SRD, which determines whether the SRD shares its SIP resources (SIP Interfaces, Proxy Sets, and IP Groups) with all other SRDs (Shared and Isolated).</td>
</tr>
<tr>
<td></td>
<td>• [0] Shared = (Default) SRD shares its resources with other SRDs (Isolated and Shared) and calls can thus be routed between the SRD and other SRDs.</td>
</tr>
<tr>
<td></td>
<td>• [1] Isolated = SRD does not share its resources with other SRDs and calls cannot be routed between the SRD and other Isolated SRDs. However, calls can be routed between the SRD and other Shared SRDs.</td>
</tr>
<tr>
<td></td>
<td>For more information on SRD Sharing Policy, see Multiple SRDs for Multi-tenant Deployments on page 333.</td>
</tr>
<tr>
<td><strong>SBC Operation Mode</strong></td>
<td></td>
</tr>
<tr>
<td>sbc-operation-mode [SRD_SBCOperationMode]</td>
<td>Defines the device's operational mode for the SRD.</td>
</tr>
<tr>
<td></td>
<td>• [0] B2BUA = (Default) Device operates as a back-to-back user agent (B2BUA), changing the call identifiers and headers between the inbound and outbound legs.</td>
</tr>
<tr>
<td></td>
<td>• [1] Call Stateful Proxy = Device operates as a Stateful Proxy, passing the SIP message transparently between inbound and outbound legs. In other words, the same SIP dialog identifiers (tags, Call-Id and CSeq) occur on both legs (as long as no other configuration disrupts the CSeq compatibility).</td>
</tr>
<tr>
<td></td>
<td>• [2] Microsoft Server = Operating mode for the One-Voice Resiliency feature, whereby the device is deployed together with Skype for Business-compatible IP Phones at small remote branch offices in a Microsoft® Skype for Business™ environment.</td>
</tr>
<tr>
<td></td>
<td>For more information on B2BUA and Stateful Proxy modes, see B2BUA and Stateful Proxy Operating Modes on page 454.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• The settings of the parameter also determines the default behavior of related parameters in the IP Profiles table (SBCRemoteRepresentationMode, SBCKeepVIAHeaders, SBCKeepUserAgentHeader, SBCKeepRoutingHeaders, SBCRemoteMultipleEarlyDialogs).</td>
</tr>
<tr>
<td></td>
<td>• If the 'SBC Operation Mode' parameter is configured in the IP Groups table, the 'SBC Operation Mode' parameter in the SRDs table is ignored.</td>
</tr>
<tr>
<td><strong>SBC Routing Policy</strong></td>
<td>Assigns a Routing Policy to the SRD.</td>
</tr>
<tr>
<td></td>
<td>By default, no value is defined if you have configured multiple...</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>sbc-routing-policy-name</code>&lt;br&gt; [SRD_SBCRoutingPolicyName]</td>
<td>Routing Policies. If you have configured only one Routing Policy, the device assigns it to the SRD by default. For more information on Routing Policies, see Configuring SBC Routing Policy Rules on page 521. <strong>Note:</strong>&lt;br&gt; - If you have assigned a Routing Policy to a Classification rule that is associated with the SRD, the Routing Policy assigned to the SRD is ignored.&lt;br&gt; - You can assign the same Routing Policy to multiple SRDs.</td>
</tr>
<tr>
<td><code>Used By Routing Server</code>&lt;br&gt; <code>used-by-routing-server</code>&lt;br&gt; [SRD_UsedByRoutingServer]</td>
<td>Enables the SRD to be used by a third-party routing server for call routing decisions.&lt;br&gt; - [0] Not Used (default)&lt;br&gt; - [1] Used&lt;br&gt; For more information on the third-party routing server feature, see &quot;Centralized Third-Party Routing Server&quot; on page 275.</td>
</tr>
<tr>
<td><code>Dial Plan</code>&lt;br&gt; <code>sbc-dial-plan-name</code>&lt;br&gt; [SRD_SBCDialPlanName]</td>
<td>Assigns a Dial Plan to the SRD. The device searches the Dial Plan for a dial plan rule that matches the source number and if not found, for a rule that matches the destination number. If a matching dial plan rule is found, the rule’s tag is used in the routing and/or manipulation processes as source and/or destination tags.&lt;br&gt; To configure Dial Plans, see Configuring Dial Plans on page 541.</td>
</tr>
<tr>
<td><strong>Registration</strong></td>
<td></td>
</tr>
<tr>
<td><code>Max. Number of Registered Users</code>&lt;br&gt; <code>max-reg-users</code>&lt;br&gt; [SRD_MaxNumOfRegUsers]</td>
<td>Defines the maximum number of users belonging to the SRD that can register with the device.&lt;br&gt; The default is -1, which means that the number of allowed user registrations is unlimited.</td>
</tr>
<tr>
<td><code>User Security Mode</code>&lt;br&gt; <code>block-un-reg-users</code>&lt;br&gt; [SRD_BlockUnRegUsers]</td>
<td>Defines the blocking (reject) policy for incoming SIP dialog-initiating requests (e.g., INVITE messages) from registered and unregistered users belonging to the SRD.&lt;br&gt; - [0] Accept All (Default) Accepts requests from registered and unregistered users.&lt;br&gt; - [1] Accept Registered Users = Accepts requests only from users registered with the device. Requests from users not registered are rejected.&lt;br&gt; - [2] Accept Registered Users from Same Source = Accepts requests only from registered users whose source address is the same as that registered with the device (during the REGISTER message process). All other requests are rejected. The device verifies whether the IP address and port are different only if the transport protocol is UDP; otherwise, the device verifies only the IP address. The verification is performed before any of the device’s call handling processes (i.e., Classification, Manipulation and Routing).&lt;br&gt; <strong>Note:</strong>&lt;br&gt; - The parameter is applicable only to calls belonging to User-type IP Groups.&lt;br&gt; - The feature is not applicable to REGISTER requests.&lt;br&gt; - The option, Accept Registered Users from Same Source [2] does not apply to registration refreshes. These requests are accepted.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Enable Un-Authenticated Registrations</td>
<td>Enables the device to accept REGISTER requests and register them in its registration database from new users that have not been authenticated by a proxy/registrar server (due to proxy down) and thus, re-routed to a User-type IP Group.</td>
</tr>
<tr>
<td>enable-un-auth-registrs [SRD_EnableUnAuthenticated Registrations]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>In normal operation scenarios in which the proxy server is available, the device forwards the REGISTER request to the proxy and if authenticated by the proxy (i.e., device receives a success response), the device adds the user to its registration database. The routing to the proxy is according to the SBC IP-to-IP Routing table where the destination is the proxy's IP Group. However, when the proxy is unavailable (e.g., due to network connectivity loss), the device can accept REGISTER requests from new users if a matching alternative routing rule exists in the SBC IP-to-IP Routing table where the destination is the user's User-type IP Group (i.e., call survivability scenarios) and if the parameter is enabled.</td>
</tr>
<tr>
<td></td>
<td>- [0] Disable = The device rejects REGISTER requests from new users that were not authenticated by a proxy server.</td>
</tr>
<tr>
<td></td>
<td>- [1] Enable = (Default) The device accepts REGISTER requests from new users even if they were not authenticated by a proxy server, and registers the user in its registration database.</td>
</tr>
<tr>
<td>Note:</td>
<td>Regardless of the parameter, the device always accepts registration refreshes from users that are already registered in its database.</td>
</tr>
<tr>
<td></td>
<td>For a SIP Interface that is associated with the SRD, if the corresponding parameter in the SIP Interfaces table (SIPInterface_EnableUnAuthenticatedRegistrations) is configured to Disable or Enable, the parameter in the SRD is ignored for calls belonging to the SIP Interface.</td>
</tr>
</tbody>
</table>
19.2.1 Filtering Tables in Web Interface by SRD

When your configuration includes multiple SRDs, you can filter tables in the Web interface by SRD. The filter is configured in the SRD Filter drop-down list, located on the Web interface’s toolbar, as shown below.

![Figure 19-8: Filtering Display by SRD](image)

The filter is applied throughout the Web GUI. When you select an SRD for filtering, the Web interface displays only table rows associated with the filtered SRD. When you add a new row to a table, the filtered SRD is automatically selected as the associated SRD. For example, if you filter the Web display by SRD "Comp-A" and you then add a new Proxy Set, the Proxy Set is automatically associated with this SRD (i.e., the 'SRD' parameter is set to "Comp-A"). All other parameters in the dialog box are also automatically set to values associated with the filtered SRD.

The SRD filter also affects display of number of configured rows and invalid rows by status icons on table items in the Navigation tree. The status icons only display information relating to the filtered SRD.

SRD filtering is especially useful in multi-tenant setups where multiple SRDs may be configured. In such a setup, SRD filtering eliminates configuration clutter by "hiding" SRDs that are irrelevant to the current configuration and facilitates configuration by automatically associating the filtered SRD, and other configuration elements associated with the filtered SRD, wherever applicable.

19.2.2 Multiple SRDs for Multi-tenant Deployments

The device can be deployed in a multi-tenant architecture, serving multiple customers (tenants) from a single, shared physical entity. The device’s multi-tenant feature is fully scalable, offering almost “non-bleeding” partition per tenant, whereby users of one tenant can’t infringe on the space of users of another tenant. The device provides per tenant configuration, monitoring, reporting, analytics, alarms and interfacing. The device is a real-time multi-tenant system that provides each tenant with optimal real-time performance, as each session received by the device is classified and processed only through the tenant’s “orbit”.

While some enterprises are large enough to justify a dedicated standalone device, many enterprises require only a fraction of the device’s capacity and capabilities. Service providers offering SIP Trunking services can funnel multiple enterprises into a single device and thereby, reap significant cost improvements over a device-per-customer model. Tenant size in a multi-tenant architecture can vary and therefore, the instance CPU, memory and interface allocations should be optimized so as not to waste resources for small-sized
tenants on the one hand, and not to allocate too many instances for a single tenant/customer on the other. For example, it would be a waste to allocate a capacity of 100 concurrent sessions to a small tenant for which 10 concurrent sessions suffice.

In a multi-tenant deployment, each tenant is represented by a dedicated SRD. The different Layer-3 networks (e.g., LAN IP-PBX users, WAN SIP Trunk, and WAN far-end users) of the tenant are represented by SIP Interfaces, which are all associated with the tenant's SRD. As related configuration entities (SIP Interfaces, IP Groups, Proxy Sets, Classification rules, and IP-to-IP Routing rules) are associated with the specific SRD, each SRD has its own logically separated configuration tables (although configured in the same tables). Therefore, full logical separation (on the SIP application layer) between tenants is achieved by SRD.

To create a multi-tenant configuration topology that is as non-bleeding as possible, you can configure an SRD (tenant) as **Isolated and Shared**:

- **Isolated SRD**: An Isolated SRD has its own dedicated SIP resources (SIP Interfaces, Proxy Sets, and IP Groups). No other SRD can use the SIP resources of an Isolated SRD. Thus, call traffic of an Isolated SRD is kept separate from other SRDs (tenants), preventing any risk of traffic "leakage" with other SRDs.

  Isolated SRDs are more relevant when each tenant needs its own separate (dedicated) routing "table" for non-bleeding topology. Separate routing tables are implemented using Routing Policies. In such a non-bleeding topology, routing between Isolated SRDs is not possible. This enables accurate and precise routing per SRD, eliminating any possibility of erroneous call routing between SRDs, restricting routing to each tenant's (SRD's) sphere. Configuring only one Routing Policy that is shared between Isolated SRDs is not best practice for non-bleeding environments, since it allows routing between these SRDs.

- **Shared SRD**: Isolated SRDs have their own dedicated SIP resources (SIP Interfaces, Proxy Sets, and IP Groups). This may not be possible in some deployments. For example, in deployments where all tenants use the same SIP Trunking service, or use the same SIP Interface due to limited SIP interface resources (e.g., multiple IP addresses cannot be allocated and SIP port 5060 must be used). In contrast to Isolated SRDs, a Shared SRD can share its' SIP resources with all other SRDs (Shared and Isolated). This is typically required when tenants need to use common resources. In the SIP Trunk example, the SIP Trunk would be associated with a Shared SRD, enabling all tenants to route calls with the SIP Trunk.

Another configuration entity that can be used for multi-tenant deployments is the Routing Policy. Routing Policies allow each SRD (or tenant) to have its own routing rules, manipulation rules, Least Cost Routing (LCR) rules, and/or LDAP-based routing configuration. However, not all multi-tenant deployments need multiple Routing Policies and typically, their configuration is not required. Isolated SRDs are more relevant only when each tenant requires its own dedicated Routing Policy to create separate, dedicated routing "tables"; for all other scenarios, SRDs can be Shared. For more information on Routing Policies, see "Configuring SBC Routing Policy Rules" on page 521.

The figure below illustrates a multi-tenant architecture with Isolated SRD tenants ("A" and "B") and a Shared SRD tenant ("Data Center") serving as a SIP Trunk:
To facilitate multi-tenant configuration through CLI, you can access a specific tenant "view". Once in a specific tenant view, all configuration commands apply only to the currently viewed tenant. Only table rows (indexes) belonging to the viewed tenant can be modified. New table rows are automatically associated with the viewed tenant (i.e., SRD name). The display of tables and show running-configuration commands display only rows relevant to the viewed tenant (and shared tenants). The show commands display only information relevant to the viewed tenant. To support this CLI functionality, use the following commands:

- To access a specific tenant view:
  
  ```
  # srd-view <SRD name>
  ```

  Once accessed, the tenant's name (i.e., SRD name) forms part of the CLI prompt, for example:

  ```
  # srd-view datacenter
  (srd-datacenter)#
  ```

- To exit the tenant view:

  ```
  # no srd-view
  ```
19.2.3 Cloning SRDs

You can clone (duplicate) existing SRDs. This is especially useful when operating in a multi-tenant environment and you need to add new tenants (SRDs). The new tenants can quickly and easily be added by simply cloning one of the existing SRDs. Once cloned, all you need to do is tweak configuration entities associated with the SRD clone.

When an SRD is cloned, the device adds the new SRD clone to the next available index row in the SRDs table. The SRD clone is assigned a unique name in the following syntax format: `<unique clone ID>_<original SRD index>_CopyOf_<name, or index if no name, of original SRD>`. For example, if you clone SRD "SIP-Trunk" at index 2, the new SRD clone is assigned the name, "36454371_2_CopyOf_SIP-Trunk".

The SRD clone has identical settings as the original SRD. In addition, all configuration entities associated with the original SRD are also cloned and these clones are associated with the SRD clone. The naming convention of these entities is the same as the SRD clone (see above) and all have the same unique clone ID ("36454371" in the example above) as the cloned SRD. These configuration entities include IP Groups, SIP Interfaces, Proxy Sets (without addresses), Classification rules, and Admission Control rules. If the Routing Policy associated with the original SRD is not associated with any other SRD, the Routing Policy is also cloned and its' clone is associated with the SRD clone. All configuration entities associated with the original Routing Policy are also cloned and these clones are associated with the Routing Policy clone. These configuration entities include IP-to-IP Routing rules, Inbound Manipulation rules, and Outbound Manipulation rules.

When any configuration entity is cloned (e.g., an IP-to-IP Route rule) as a result of a cloned SRD, all fields of the entity's row which "point" to other entities (e.g., SIP Interface, Source IP Group, and Destination IP Group) are replaced by their corresponding clones.

Note: For some cloned entities such as SIP Interfaces, some parameter values may change. This occurs in order to avoid the same parameter having the same value in more than one table row (index), which would result in invalid configuration. For example, a SIP Interface clone will have an empty Network Interface setting. After the clone process finishes, you thus need to update the Network Interface for valid configuration.

To clone an SRD:

- Web interface: In the SRDs table, select an SRD to clone, and then click the Clone button.
- CLI:
  ```
  (config-voip)# srd clone <SRD index that you want cloned>
  ```
19.2.4 Color-Coding of SRDs in Web Interface

To easily identify your configured SRDs, the Web interface displays each SRD in a unique color. The color is automatically and randomly assigned to new SRDs and is displayed in a box alongside the name of the SRD in tables where the SRD is configured or assigned. This is applied throughout the Web interface. The following example shows SRDs assigned with unique color codes.

![Figure 19-9: Color-Coding of SRDs](image)

19.2.5 Automatic Configuration based on SRD

To facilitate configuration and eliminate possible flaws in configuration due to invalid associations between configuration entities, the Web interface automatically configures configuration entities based on SRD:

- If you delete an SRD (in the SRDs table) that is associated with other configuration entities in other tables, the device automatically deletes the associated table rows. For example, if you delete an SRD that is associated with a Proxy Set, the device automatically deletes the Proxy Set.

- If you associate an SRD with a configuration entity in another table (i.e., other than the SRDs table), the device automatically configures certain parameters of the configuration entity according to the SRD or associated SRD. For example, if you add a rule in the IP-to-IP Routing table and you select a Routing Policy, the ‘Source IP Group’ and ‘Destination IP Group’ parameters list only IP Groups that are associated with the SRD to which the Routing Policy is assigned (and IP Groups belonging to a Shared SRD, if exists).

- If your configuration setup includes only a single SRD, the device automatically selects the SRD when adding related configuration entities. For example, when adding an IP Group, the single SRD is automatically selected in the Add Row dialog box.

19.3 Configuring SIP Interfaces

The SIP Interfaces table lets you configure up to 1,200 SIP Interfaces. A SIP Interface represents a Layer-3 network in your deployment environment, by defining a local, listening port number and type (e.g., UDP), and assigning an IP network interface for SIP signaling traffic. For example, if your deployment consists of an IP PBX in the LAN, a SIP Trunk in the WAN, and remote far-end users in the WAN, you would need to configure a SIP Interface for each of these SIP entities. You can also configure various optional features for the SIP Interface such as assigning it a Media Realm, blocking calls received on the SIP Interface from users not registered with the device, and enabling direct media.

Each SIP Interface can be associated with only one SRD. As the SRD configuration entity represents your VoIP deployment SIP network (Layer 5), you need to associate your SIP Interfaces with a specific SRD in order to represent your Layer-3 networks. For most
deployments (except multi-tenant deployments), your SRD represents your entire network and thus, only one SRD is required. The device provides a default SRD and in such scenarios where only a single SRD is required, your SIP Interfaces are automatically assigned to the default SRD. Therefore, there is no need to even handle SRD configuration entity.

Once configured, you can apply SIP Interfaces to calls, by assigning them to the following configuration entities in their respective tables:

- (Mandatory) Proxy Set to specify the SIP Interface for communication with the proxy server (i.e., IP Group). For more information, see "Configuring Proxy Sets" on page 359.
- Intrusion Detection System (IDS) for applying the IDS policy to a specific SIP Interface. For more information, see "Configuring IDS Policies" on page 172.
- SBC application:
  - IP-to-IP Routing rules for specifying the destination SIP Interface to where you want to route the call. For more information, see Configuring SBC IP-to-IP Routing Rules on page 501.
  - Classification rules for specifying the SIP Interface as a matching characteristic of the incoming call. This is especially useful for the single SRD-configuration topology, where each SIP Interface represents a Layer-3 network (SIP entity). Therefore, classification of calls to IP Groups (SIP entities) can be based on SIP Interface. For more information, see Configuring Classification Rules on page 493.
  - Admission Control rules to apply call admission control per SIP Interface. For more information, see "Configuring Admission Control" on page 489.

Note: The device terminates active calls associated with a SIP Interface in the following scenarios:

- If you delete the associated SIP Interface.
- If you edit any of the following fields of the associated SIP Interface: ‘Application Type’, ‘UDP Port’, ‘TCP Port’, ‘TLS Port’ or ‘SRD’ fields.
- If you edit or delete a network interface in the IP Interfaces table that is associated with the SIP Interface.

The following procedure describes how to configure SIP interfaces through the Web interface. You can also configure it through ini file (SIPInterface) or CLI (configure voip > sip-interface).

To configure a SIP Interface:

1. Open the SIP Interfaces table (Setup menu > Signaling & Media tab > Core Entities folder > SIP Interfaces).
2. Click New; the following dialog box appears:
3. Configure a SIP Interface according to the parameters described in the table below.
4. Click **Apply**.

### Table 19-5: SIP Interfaces table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SRD</strong></td>
<td>Assigns an SRD to the SIP Interface.</td>
</tr>
</tbody>
</table>
| srd-name           | If only one SRD is configured in the SRDs table, the SRD is assigned to the SIP Interface by default. If multiple SRDs are configured in the SRDs table, no value is defined and you must assign an SRD. To configure SRDs, see "Configuring SRDs" on page 327. Note:  
  - The parameter is mandatory.  
  - You can assign the same SRD to multiple SIP Interfaces. |
| **General**        |                                                                                                                                              |
| Index [SIPInterface_Index] | Defines an index for the new table row.                                                                                                           |
| Name interface-name [SIPInterface_InterfaceName] | Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. By default, if you do not configure a name, the device automatically assigns the name "SIPInterface_<row index>" (e.g., "SIPInterface_1" when added to Index 1). |
| Topology Location topology-location [SIPInterface_TopologyLocation] | Defines the display location of the SIP Interface in the Topology view in the Web interface.  
  - [0] Down = (Default) The SIP Interface element is displayed on the lower border of the view.  
  - [1] Up = The SIP Interface element is displayed on the upper border of the view.  
  For more information on the Topology view, see "Building and Viewing SIP Entities in Topology View" on page 369. |
<p>| Network Interface network-interface | Assigns a Control-type IP network interface to the SIP Interface. By default, no value is defined. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [SIPInterface_NetworkInterface] | To configure network interfaces, see "Configuring IP Network Interfaces" on page 137.  
**Note:** The parameter is mandatory. |
| Application Type application-type [SIPInterface_ApplicationType] | Defines the application for which the SIP Interface is used.  
| UDP Port udp-port [SIPInterface_UDPPort] | Defines the device's listening and source port for SIP signaling traffic over UDP.  
The valid range is 1 to 65534. The default is 5060.  
**Note:**  
- The port **must** be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.  
- Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). |
| TCP Port tcp-port [SIPInterface_TCPPort] | Defines the device's listening port for SIP signaling traffic over TCP.  
The valid range is 1 to 65534. The default is 5060.  
**Note:**  
- The port **must** be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.  
- Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). |
| TLS Port tls-port [SIPInterface_TLSPort] | Defines the device's listening port for SIP signaling traffic over TLS.  
The valid range is 1 to 65534. The default is 5061.  
**Note:**  
- The port **must** be different from ports configured for RTP traffic (i.e., ports configured for Media Realms). For example, if the RTP port range is 6000 to 6999, the SIP port can either be less than 6000 or greater than 6999.  
- Each SIP Interface must have a unique signaling port (i.e., no two SIP Interfaces can share the same port - no port overlapping). |
| Additional UDP Ports additional-udp-ports [SIPInterface_AdditionalUDPPorts] | Defines a port range for the device's local, listening and source ports for SIP signaling traffic over UDP. The parameter is used to assign a specific local port to each SIP entity (e.g., PBX) communicating with a common SIP entity (e.g., proxy server). This applies to the device's local ports on the leg interfacing with the proxy server. In other words, this SIP Interface is associated with the proxy server.  
The valid range is 1,025 to 65535. The range is configured using the syntax x-y, where x is the starting port and y the ending port of the range (e.g., 6000-7000). By default, the parameter is not configured. For more information on configuring specific UDP ports for SIP entities communicating with a common proxy server, see Configuring Specific UDP Ports using Tag-based Routing on page 515.  
**Note:** |
### 19. Control Network

**Parameter** | **Description**
--- | ---
- The parameter's port range value must not overlap with the UDP port configured by the 'UDP Port' parameter (SIPInterface_UDPPort). For example, if the 'UDP Port' parameter is configured to 5070, you cannot configure the 'Additional UDP Ports' parameter with a range of 5060-6000.
- The parameter's port range value must not overlap with UDP port ranges of Media Realms that are configured on the same network interface. For example, if the RTP port range is 6000-6999, you must configure the 'Additional UDP Ports' parameter to a range that is less than 6000 or greater than 6999.
- The maximum number of ports in the range is limited to the maximum number of IP Groups that can be configured (see Configuring IP Groups on page 345).

<table>
<thead>
<tr>
<th>Encapsulating Protocol</th>
<th>Defines the type of incoming traffic (SIP messages) expected on the SIP Interface.</th>
</tr>
</thead>
</table>
- [1] WebSocket = Traffic received on the SIP Interface is identified by the device as WebSocket signaling traffic (encapsulated by WebSocket frames). For outgoing traffic, the device encapsulates the traffic using the WebSocket protocol (frames) on the TCP/TLS ports.  
For more information on WebSocket, see SIP over WebSocket on page 568.  
**Note:** WebSocket encapsulation is not supported for UDP ports. |

<table>
<thead>
<tr>
<th>Enable TCP Keepalive</th>
<th>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.</th>
</tr>
</thead>
</table>
| [SIPInterface_TCPKeepaliveEnable] | - [0] Disable (default)  
- [1] Enable  
**Note:** To configure TCP keepalive, use the following ini file parameters: TCPKeepAliveTime, TCPKeepAliveInterval, and TCPKeepAliveRetry. |

<table>
<thead>
<tr>
<th>Used By Routing Server</th>
<th>Enables the SIP Interface to be used by a third-party routing server for call routing decisions.</th>
</tr>
</thead>
</table>
| [SIPInterface_UsedByRoutingServer] | - [0] Not Used (default)  
- [1] Used  
For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 275. |

| Pre-Parsing Manipulation Set | Assigns a Pre-Parsing Manipulation Set to the SIP Interface. This lets you apply pre-parsing SIP message manipulation rules on any incoming SIP message received on this SIP Interface.  
By default, no Pre-Parsing Manipulation Set is assigned.  
To configure Pre-Parsing Manipulation Sets, see Configuring Pre-parsing Manipulation Rules on page 403.  
**Note:**  
- Pre-Parsing Manipulation is done only on incoming calls.  
The device performs Pre-Parsing Manipulation before Pre-Classification Manipulation and Classification. |
| [SIPInterface_PreParsingManSetName] |  

---

**Version 7.2**  
341  
**Mediant VE/SE SBC**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification</strong></td>
<td></td>
</tr>
<tr>
<td>Classification Failure Response Type</td>
<td>Defines the SIP response code that the device sends if a received SIP request (OPTIONS, REGISTER, or INVITE) fails the SBC Classification process.</td>
</tr>
</tbody>
</table>
| [SIPInterface_ClassificationFailureResponseType] | The valid value can be a SIP response code from 400 through 699, or it can be set to 0 to not send any response at all. The default response code is 500 (Server Internal Error). This feature is important for preventing Denial of Service (DoS) attacks, typically initiated from the WAN. Malicious attackers can use SIP scanners to detect ports used by SIP devices. These scanners scan devices by sending UDP packets containing a SIP request to a range of specified IP addresses, listing those that return a valid SIP response. Once the scanner finds a device that supports SIP, it extracts information from the response and identifies the type of device (IP address and name) and can execute DoS attacks. A way to defend the device against such attacks is to not send a SIP reject response to these unclassified "calls" so that the attacker assumes that no device exists at such an IP address and port. **Note:**
  - The parameter is applicable only if you configure the device to reject unclassified calls, which is done using the 'Unclassified Calls' parameter (see Configuring Classification Rules on page 493). |
| Pre Classification Manipulation Set ID | 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. To configure Message Manipulation rules, see Configuring SIP Message Manipulation on page 393. **Note:**
  - 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. |
| [SIPInterface_PreClassificationManipulationSet] | 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. |
| **Media** |  |
| Media Realm | Assigns a Media Realm to the SIP Interface. By default, no value is defined. To configure Media Realms, see "Configuring Media Realms" on page 319. |
| [SIPInterface_MediaRealm] |  |
| Direct Media | Enables direct media (RTP/SRTP) flow (i.e., no Media Anchoring) between endpoints associated with the SIP Interface. **Note:**
  - [0] Disable = (Default) Media Anchoring is employed, whereby the media stream traverses the device (and each leg uses a different coder or coder parameters).  
  - [1] Enable = No Media Anchoring. Media stream flows directly |
| [SIPInterface_SBCDirectMedia] |  |
### Control Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter</td>
<td>between endpoints (i.e., does not traverse the device - no Media Anchoring).</td>
</tr>
<tr>
<td></td>
<td>[2] Enable when Same NAT = No Media Anchoring. Media stream flows directly between endpoints if they are located behind the same NAT.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>If the parameter is enabled for direct media and the two endpoints belong to the same SIP Interface, calls cannot be established if the following scenario exists:</td>
</tr>
<tr>
<td></td>
<td>a. One of the endpoints is defined as a foreign user (for example, &quot;follow me service&quot;)</td>
</tr>
<tr>
<td></td>
<td>b. and one endpoint is located on the WAN and the other on the LAN.</td>
</tr>
<tr>
<td></td>
<td>The reason for the above is that in direct media, the device does not interfere in the SIP signaling such as manipulation of IP addresses, which is necessary for calls between LAN and WAN.</td>
</tr>
<tr>
<td></td>
<td>To enable direct media for all calls, use the global parameter SBCDirectMedia. If enabled, even if the SIP Interface is disabled for direct media, direct media is employed for calls belonging to the SIP Interface.</td>
</tr>
<tr>
<td></td>
<td>If you enable direct media for the SIP Interface, make sure that your Media Realm provides sufficient ports, as media may traverse the device for mid-call services (e.g., call transfer).</td>
</tr>
<tr>
<td></td>
<td>For more information on direct media, see Direct Media on page 464.</td>
</tr>
</tbody>
</table>

#### Security

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS Context Name</td>
<td>Assigns a TLS Context (SSL/TLS certificate) to the SIP Interface.</td>
</tr>
<tr>
<td>tls-context-name</td>
<td>The default TLS Context (&quot;default&quot; at Index 0) is assigned to the SIP Interface by default.</td>
</tr>
<tr>
<td>[SIPInterface_TLSContext]</td>
<td>Note:</td>
</tr>
<tr>
<td></td>
<td>For incoming calls: The assigned TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call or classification to an IP Group based on Proxy Set fails.</td>
</tr>
<tr>
<td></td>
<td>For outgoing calls: The assigned TLS Context is used if no TLS Context is configured for the Proxy Set associated with the call.</td>
</tr>
<tr>
<td></td>
<td>To configure TLS Contexts, see &quot;Configuring SSL/TLS Certificates&quot; on page 105.</td>
</tr>
<tr>
<td>TLS Mutual Authentication</td>
<td>Enables TLS mutual authentication for the SIP Interface (when the device acts as a server).</td>
</tr>
<tr>
<td>tls-mutual-auth</td>
<td>[0] Disable = Device does not request the client certificate for TLS connection on the SIP Interface.</td>
</tr>
<tr>
<td>[SIPInterface_TLSMutualAuthentication]</td>
<td>[1] Enable = Device requires receipt and verification of the client certificate to establish the TLS connection on the SIP Interface.</td>
</tr>
<tr>
<td></td>
<td>By default, no value is defined and the SIPSRequireClientCertificate global parameter setting is applied.</td>
</tr>
<tr>
<td>Message Policy</td>
<td>Assigns a SIP message policy to the SIP interface.</td>
</tr>
<tr>
<td>message-policy-name</td>
<td>To configure SIP Message Policy rules, see &quot;Configuring SIP Message Policy Rules&quot;.</td>
</tr>
<tr>
<td>[SIPInterface_MessagePolicyName]</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| **User Security Mode**<br>block-un-reg-users<br>[SIPInterface_BlockUnRegUsers]** | Defines the blocking (reject) policy for incoming SIP dialog-initiating requests (e.g., INVITE messages) from registered and unregistered users belonging to the SIP Interface.  
- [-1] Not Configured = (Default) The corresponding parameter in the SRDs table (SRD_BlockUnRegUsers) of the SRD that is associated with the SIP Interface is applied.  
- [0] Accept All = Accepts requests from registered and unregistered users.  
- [1] Accept Registered Users = Accepts requests only from users registered with the device. Requests from users not registered are rejected.  
- [2] Accept Registered Users from Same Source = Accepts requests only from registered users whose source address is the same as that registered with the device (during the REGISTER message process). All other requests are rejected. The device verifies whether the IP address and port are different only if the transport protocol is UDP; otherwise, the device verifies only the IP address. The verification is performed before any of the device's call handling processes (i.e., Classification, Manipulation and Routing).  

**Note:**  
- The parameter is applicable only to calls belonging to User-type IP Groups.  
- The feature is not applicable to REGISTER requests.  
- The option, Accept Registered Users from Same Source [2] does not apply to registration refreshes. These requests are accepted even if the source address is different to that registered with the device.  
- When the device rejects a call, it sends a SIP 500 "Server Internal Error" response to the user. In addition, it reports the rejection (Dialog establish failure - Classification failure) using the Intrusion Detection System (IDS) feature (see Configuring IDS Policies on page 172), by sending an SNMP trap.  
- If you configure the parameter to any value other than default [-1], it overrides the corresponding parameter in the SRDs table (SRD_BlockUnRegUsersInterface) for the SRD associated with the SIP Interface. |
| **Enable Un-Authenticated Registrations**<br>enable-un-auth-registrs<br>[SIPInterface_EnableUnAuthenticatedRegistrations]** | Enables the device to accept REGISTER requests and register them in its registration database from new users that have not been authenticated by a proxy/registrar server (due to proxy down) and thus, re-routed to a User-type IP Group.  
In normal operation scenarios in which the proxy server is available, the device forwards the REGISTER request to the proxy and if authenticated by the proxy (i.e., device receives a success response), the device adds the user to its registration database. The routing to the proxy is according to the SBC IP-to-IP Routing table where the destination is the proxy's IP Group. However, when the proxy is unavailable (e.g., due to network connectivity loss), the device can accept REGISTER requests from new users if a matching alternative routing rule exists in the SBC IP-to-IP Routing table where the destination is the user's User-type IP Group (i.e., call survivability scenarios) and if the parameter is enabled.  
- [-1] Not Configured = (Default) The corresponding parameter in
19. Control Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Number of Registered Users</td>
<td>Defines the maximum number of users belonging to the SIP Interface that can register with the device. By default, no value is defined (i.e., the number of allowed user registrations is unlimited).</td>
</tr>
</tbody>
</table>

### 19.4 Configuring IP Groups

The IP Groups table lets you configure up to 5,000 IP Groups. An IP Group represents a SIP entity in the network with which the device communicates. This can be a server (e.g., IP PBX or ITSP) or a group of users (e.g., LAN IP phones). For servers, the IP Group is typically used to define the server's IP address by associating it with a Proxy Set (see Configuring Proxy Sets on page 359).

You can use IP Groups for the following:

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

- Representing the source and destination of the call in IP-to-IP Routing rules (see Configuring SBC IP-to-IP Routing Rules on page 501).

- 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 Accounts table (see "Configuring Registration Accounts" on page 375).

- Included in routing decisions by a third-party routing server. If deemed necessary for routing, the routing server can even create an IP Group. For more information, see Centralized Third-Party Routing Server on page 275.

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. To configure Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 307.

- **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. To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 312.

**Note:**

- If you delete an IP Group or modify the 'Type' or 'SRD' parameters, the device immediately terminates currently active calls that are associated with the IP Group. In addition, all users belonging to the IP Group are removed from the device's users database.
- The device supports up to 5,000 IP Groups only when it has 32-GB or 64-GB RAM. For less than 32-GB RAM, the maximum is 1,500 IP Groups.

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

➢ **To configure an IP Group:**

1. Open the IP Groups table (Setup menu > Signaling & Media tab > Core Entities folder > IP Groups).
2. Click New; the following dialog box appears:

3. Configure an IP Group according to to the parameters described in the table below.
4. Click Apply.

**Table 19-6: IP Groups Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRD srd-name</td>
<td>Assigns an SRD to the IP Group. If only one SRD is configured in the SRDs table, the SRD is assigned by default. If multiple SRDs are configured in the SRDs table, no value is assigned by default and you must assign one.</td>
</tr>
<tr>
<td>[IPGroup_SRDName]</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index for the new table row.</td>
</tr>
<tr>
<td>[IPGroup_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td>[IPGroup_Name]</td>
<td>The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td>[IPGroup_Name]</td>
<td><strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>Topology Location</td>
<td>Defines the display location of the IP Group in the Topology view of the Web interface.</td>
</tr>
<tr>
<td>topology-location</td>
<td>• [0] Down = (Default) The IP Group element is displayed on the lower border of the view.</td>
</tr>
<tr>
<td>[IPGroup_TopologyLocation]</td>
<td>• [1] Up = The IP Group element is displayed on the upper border of the view.</td>
</tr>
<tr>
<td></td>
<td>For more information on the Topology view, see &quot;Building and Viewing SIP Entities in Topology View&quot; on page 369.</td>
</tr>
<tr>
<td>Type</td>
<td>Defines the type of IP Group:</td>
</tr>
<tr>
<td>type</td>
<td>• [0] Server = Applicable when the destination address of the IP Group (e.g., ITSP, Proxy, IP-PBX, or Application server) is known. The address is configured by the Proxy Set that is associated with the IP Group.</td>
</tr>
<tr>
<td>[IPGroup_Type]</td>
<td>• [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).</td>
</tr>
<tr>
<td></td>
<td>Typically, this IP Group is configured with a Serving IP Group that represents an IP-PBX, Application or Proxy server that serves this User-type IP Group. Each SIP request sent by a user of this IP Group is proxied to the Serving IP Group. For registrations, the device updates its registration database with the AOR and contacts of the users. Digest authentication using SIP 401/407 responses (if needed) is performed by the Serving IP Group. The device forwards these responses directly to the SIP users. To route a call to a registered user, a rule must be configured in the SBC IP-to-IP Routing table. The device searches the dynamic database (by using the Request-URI) for an entry that matches a registered AOR or Contact. Once an entry is found, the IP destination is obtained from this entry and a SIP request is sent to the destination. The device also supports NAT traversal for the SIP clients located behind NAT. In this case, the device must be defined with a global IP address.</td>
</tr>
</tbody>
</table>
|                 | • [2] Gateway = In scenarios where the device receives requests to and from a gateway representing multiple users. This IP

To configure SRDs, see Configuring SRDs on page 327.

**Note:**
- The parameter is mandatory.
- For the parameter to take effect, a device reset is required.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Group type is necessary for any of the following scenarios:</td>
</tr>
<tr>
<td></td>
<td>✓ The IP Group cannot be defined as a Server-type since its address is initially unknown and therefore, a Proxy Set cannot be configured for it.</td>
</tr>
<tr>
<td></td>
<td>✓ The IP Group cannot be defined as a User-type 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.</td>
</tr>
<tr>
<td></td>
<td>The IP address of the Gateway-type IP Group is obtained dynamically from the host part of the Contact header in the REGISTER request received from the IP Group. Therefore, routing to this IP Group is possible only once a REGISTER request is received (i.e., IP Group is registered with the device).</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>You can view the registration status of the Gateway-type IP Group in the ‘GW Group Registered Status’ field, and view the IP address of the IP Group in the ‘GW Group Registered IP Address’ field if it is registered with the device.</td>
</tr>
</tbody>
</table>

**Proxy Set**

proxy-set-name [IPGroup_ProxySetName]

Assigns a Proxy Set to the IP Group. All INVITE messages destined to the IP Group are sent to the IP address configured for the Proxy Set.

To configure Proxy Sets, see "Configuring Proxy Sets" on page 359.

**Note:**

- The Proxy Set must be associated with the same SRD as that assigned to the IP Group.
- You can assign the same Proxy Set to multiple IP Groups.
- Proxy Sets are used for Server-type IP Groups, but may in certain scenarios also be used for User-type IP Groups. For example, this is required in deployments where the device mediates between an IP PBX and a SIP Trunk, and the SIP Trunk requires SIP registration for each user that requires service. In such a scenario, the device must register all the users to the SIP Trunk on behalf of the IP PBX. This is done by using the User Info table where each user is associated with the source IP Group (i.e., the IP PBX). To configure the User Info table, see SBC User Information for SBC User Database on page 663.

**IP Profile**

ip-profile-name [IPGroup_ProfileName]

Assigns an IP Profile to the IP Group.

By default, no value is defined.

To configure IP Profiles, see "Configuring IP Profiles" on page 417.

**Media Realm**

media-realm-name [IPGroup_MediaRealm]

Assigns a Media Realm to the IP Group. The Media Realm determines the UDP port range and maximum sessions on a specific interface for media traffic associated with the IP Group.

By default, no value is defined.

To configure Media Realms, see Configuring Media Realms on
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Contact User</strong>&lt;br&gt;contact-user&lt;br&gt; [IPGroup_ContactUser]</td>
<td>Defines the user part of the From, To, and Contact headers of SIP REGISTER messages, and the user part of the Contact header of INVITE messages received from this IP Group and forwarded by the device to another IP Group. By default, no value is defined. <strong>Note:</strong>&lt;br&gt;• The parameter is applicable only to Server-type IP Groups.&lt;br&gt;• The parameter is overridden by the ‘Contact User’ parameter in the Accounts table (see &quot;Configuring Registration Accounts&quot; on page 375).</td>
</tr>
<tr>
<td><strong>SIP Group Name</strong>&lt;br&gt;sip-group-name&lt;br&gt; [IPGroup_SIPGroupName]</td>
<td>Defines the SIP Request-URI host name in INVITE and REGISTER messages sent to the IP Group, or the host name in the From header of INVITE messages received from the IP Group. In other words, it replaces the original host name. The valid value is a string of up to 100 characters. By default, no value is defined. <strong>Note:</strong>&lt;br&gt;• If the parameter is not configured, the value of the global parameter, ProxyName is used instead (see &quot;Configuring Proxy and Registration Parameters&quot; on page 382).&lt;br&gt;• The parameter overrides inbound message manipulation rules that manipulate the host name in Request-URI, To, and/or From SIP headers. If you configure the parameter and you want to manipulate the host name in any of these SIP headers, you must apply your manipulation rule (Manipulation Set ID) to the IP Group as an Outbound Message Manipulation Set (see the IPGroup_OutboundManSet parameter), when the IP Group is the destination of the call. If you apply the Manipulation Set as an Inbound Message Manipulation Set (see the IPGroup_InboundManSet parameter), when the IP Group is the source of the call, the manipulation rule is overridden by the SIP Group Name parameter.</td>
</tr>
<tr>
<td><strong>Created By Routing Server</strong>&lt;br&gt;[IPGroup_CreatedByRoutingServer]</td>
<td>(Read-only) Indicates whether the IP Group was created by a third-party routing server:&lt;br&gt;• [0] No&lt;br&gt;• [1] Yes&lt;br&gt;For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 275.</td>
</tr>
<tr>
<td><strong>Used By Routing Server</strong>&lt;br&gt;used-by-routing-server&lt;br&gt; [IPGroup_UsedByRoutingServer]</td>
<td>Enables the IP Group to be used by a third-party routing server for call routing decisions.&lt;br&gt;• [0] Not Used (default)&lt;br&gt;• [1] Used&lt;br&gt;For more information on the third-party routing server feature, see Centralized Third-Party Routing Server on page 275.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Proxy Set Connectivity show voip proxy sets status [IPGroup_ProxySetConnectivity] | (Read-only field) Displays the connectivity status with Server-type IP Groups. As the Proxy Set defines the address of the IP Group, the connectivity check (keep-alive) by the device is done to this address.  
  - "NA": Functionality is not applicable due to one of the following:  
    - User-type IP Group.  
    - Server-type IP Group, but the keep-alive mechanism of its' associated Proxy Set is disabled.  
  - "Not Connected": Keep-alive failure (i.e., no connectivity with the IP Group).  
  - "Connected": Keep-alive success (i.e., connectivity with the IP Group).  
  The connectivity status is also displayed in the Topology View page (see "Building and Viewing SIP Entities in Topology View" on page 369).  
  **Note:**  
  - The feature is applicable only to Server-type IP Groups.  
  - To support the feature, you must enable the keep-alive mechanism of the Proxy Set that is associated with the IP Group (see "Configuring Proxy Sets" on page 359).  
  - If the Proxy Set is configured with multiple proxies (addresses) and at least one of them is "alive", the displayed status is "Connected". To view the connected proxy server, see "Viewing Proxy Set Status" on page 750.  
  - The "Connected" status also applies to scenarios where the device rejects calls with the IP Group due to low QoE (e.g., low MOS), despite connectivity. |
| SBC General Classify By Proxy Set classify-by-proxy-set [IPGroup_ClassifyByProxySet] | Enables classification of incoming SIP dialogs (INVITEs) to Server-type IP Groups based on Proxy Set (assigned using the IPGroup_ProxySetName parameter).  
  - [0] Disable  
  - [1] Enable = (Default) The device searches the Proxy Sets table for a Proxy Set that is configured with the same source IP address as that of the incoming INVITE (if host name, then according to the dynamically resolved IP address list). If such a Proxy Set is found, the device classifies the INVITE as belonging to the IP Group associated with the Proxy Set.  
  **Note:**  
  - The parameter is applicable only to Server-type IP Groups.  
  - For security, it is recommended to classify SIP dialogs based on Proxy Set only if the IP address of the IP Group is unknown. In other words, if the Proxy Set associated with the IP Group is configured with an FQDN. In such cases, the device classifies incoming SIP dialogs to the IP Group based on the DNS-resolved IP address. If the IP address is known, it is recommended to use a Classification rule instead (and disable the Classify by Proxy Set feature), where the rule is configured with not only the IP address, but also with SIP message characteristics to increase the strictness of the classification process (see Configuring Classification Rules on page 493). |
## Control Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td></td>
<td>The reason for preferring classification based on Proxy Set when the IP address is unknown is that IP address forgery (commonly known as IP spoofing) is more difficult than malicious SIP message tampering and therefore, using a Classification rule without an IP address offers a weaker form of security. When classification is based on Proxy Set, the Classification table for the specific IP Group is ignored.</td>
</tr>
<tr>
<td></td>
<td>- If you have assigned the same Proxy Set to multiple IP Groups, disable the parameter and instead, use Classification rules to classify incoming SIP dialogs to these IP Groups. If the parameter is enabled, the device is unable to correctly classify incoming INVITEs to their appropriate IP Groups.</td>
</tr>
<tr>
<td></td>
<td>- Classification by Proxy Set occurs only if classification based on the device's registration database fails (i.e., the INVITE is not from a registered user).</td>
</tr>
<tr>
<td><strong>SBC Operation Mode</strong></td>
<td>Defines the device's operational mode for the IP Group.</td>
</tr>
<tr>
<td><strong>sbc-operation-mode</strong></td>
<td>-1 Not Configured = (Default)</td>
</tr>
<tr>
<td><strong>[IPGroup_SBCOperationMode]</strong></td>
<td>0 B2BUA = Device operates as a back-to-back user agent (B2BUA), changing the call identifiers and headers between the inbound and outbound legs.</td>
</tr>
<tr>
<td></td>
<td>1 Call Stateful Proxy = Device operates as a Stateful Proxy, passing the SIP message transparently between inbound and outbound legs. In other words, the same SIP dialog identifiers (tags, Call-Id and CSeq) occur on both legs (as long as no other configuration disrupts the CSeq compatibleness).</td>
</tr>
<tr>
<td></td>
<td>2 Microsoft Server = Operating mode for the One-Voice Resiliency feature, whereby the device is deployed together with Skype for Business-compatible IP Phones at small remote branch offices in a Microsoft® Skype for Business™ environment.</td>
</tr>
<tr>
<td></td>
<td>For more information on B2BUA and Stateful Proxy modes, see B2BUA and Stateful Proxy Operating Modes on page 454.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If configured, the parameter overrides the 'SBC Operation Mode' parameter in the SRDs table.</td>
</tr>
<tr>
<td><strong>SBC Client Forking Mode</strong></td>
<td>Defines call forking of INVITE messages to up to five separate SIP outgoing legs for User-type IP Groups. This occurs if multiple contacts are registered under the same AOR in the device's registration database.</td>
</tr>
<tr>
<td><strong>sbc-client-forking-mode</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>[IPGroup_EnableSBCClientForking]</strong></td>
<td>1 Parallel = Sends the INVITE simultaneously to all contacts. The call is established with the first contact that answers.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Note:</strong> 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.</td>
<td></td>
</tr>
</tbody>
</table>

### Advanced

**Local Host Name**

- **local-host-name [IPGroup_ContactName]**
  - Defines the host name (string) that the device uses in the SIP message's Via and Contact headers. This is typically used to define an FQDN as the host name. The device uses this string for Via and Contact headers in outgoing INVITE messages sent to a specific IP Group, and the Contact header in SIP 18x and 200 OK responses for incoming INVITE messages received from a specific IP Group. The IP-to-Tel Routing table can be used to identify the source IP Group from where the INVITE message was received.
  - If the parameter is not configured, these headers are populated with the device's dotted-decimal IP address of the network interface on which the message is sent.
  - **Note:** To ensure proper device handling, the parameter should be a valid FQDN.

**UUI Format**

- **uui-format [IPGroup_UUIFormat]**
  - Enables the generation of the Avaya UCID value, adding it to the outgoing INVITE sent to this IP Group.
  - **0** Disabled (default)
  - **1** Enabled
  - This provides support for interworking with Avaya equipment by generating Avaya's UCID value in outgoing INVITE messages sent to Avaya's network. The device adds the UCID in the User-to-User SIP header.
  - Avaya's UCID value has the following format (in hexadecimal): 00 + FA + 08 + node ID (2 bytes) + sequence number (2 bytes) + timestamp (4 bytes)
  - This is interworked in to the SIP header as follows:
    - User-to-User: 00FA08001038F725B3;encoding=hex
  - **Note:** To define the Network Node Identifier of the device for Avaya UCID, use the 'Network Node ID' (NetworkNodeId) parameter.

**Always Use Src Address**

- **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 more information on NAT traversal, see "Remote UA behind NAT" on page 151.
### Control Network

#### Source URI Input

**Parameter:** 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

**Note:**
- The parameter is applicable only when classification is done according to the Classification table.
- If the configured SIP header does not exist in the incoming INVITE message, the classification of the message to a source IP Group fails.
- If the device receives an INVITE as a result of a REFER request or a 3xx response, then the incoming INVITE is routed according to the Request-URI. The device identifies such INVITEs according to a specific prefix in the Request-URI header, configured by the SBCXferPrefix parameter. Therefore, in this scenario, the device ignores the parameter setting.

#### Destination URI Input

**Parameter:** dst-uri-input  
**[IPGroup_DestUriInput]**

Defines the SIP header in the incoming INVITE to use as a call matching characteristic based on destination URIs. The parameter is used for classification and routing purposes. The device first uses the parameter's settings as a matching characteristic (input) to classify the incoming INVITE to an IP Group (source IP Group) in the Classification table. 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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[10] Contact</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
- The parameter is applicable only when classification is done according to the Classification table.
- If the configured SIP header does not exist in the incoming INVITE message, the classification of the message to a source IP Group fails.
- If the device receives an INVITE as a result of a REFER request or a 3xx response, the incoming INVITE is routed according to the Request-URI. The device identifies such INVITEs according to a specific prefix in the Request-URI header, configured by the SBCXferPrefix parameter. Therefore, in this scenario, the device ignores the parameter setting.

**SIP Connect**

**sip-connect**

Defines the IP Group as a registered server that represents multiple users. The device saves registrations received from the IP Group, with the IP address as a key in its registration database. The device classifies incoming SIP dialog requests (e.g., INVITEs) from the IP Group according to the received IP address. For requests routed to the IP Group users, the device replaces the Request-URI header with the incoming To header (which contains the remote phone number).

- [0] No (default)
- [1] Yes

**Note:** The parameter is applicable only to User-type IP Groups.

**SBC PSAP Mode**

**sbc-psap-mode**

Enables E9-1-1 emergency call routing in a Microsoft Skype for Business environment.

- [0] Disable (default)
- [1] Enable

For more information, see E9-1-1 Support for Microsoft Skype for Business on page 288.

**Route Using Request URI Port**

**use-requri-port**

Enables the device to use the port indicated in the Request-URI of the incoming message as the destination port when routing the message to the IP Group. The device uses the IP address (and not port) that is configured for the Proxy Set associated with the IP Group. The parameter thus allows the device to route calls to the same server (IP Group), but different port.

- [0] Disable = (Default) The port configured for the associated Proxy Set is used as the destination port.
- [1] Enable = The port indicated in the Request-URI of the incoming message is used as the destination port.

**DTLS Context**

**dtls-context**

Assigns a TLS Context (certificate) to the IP Group, which is used for DTLS sessions (handshakes) with the IP Group. By default, no value is defined.

To configure TLS Contexts, see Configuring TLS Certificate Contexts on page 105.

**Keep Original Call-ID**

**sbc-keep-call-id**

Enables the device to use the same call identification (SIP Call-ID header value) received in incoming messages for the call identification in outgoing messages. The call identification value is...
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ID | contained in the SIP Call-ID header.  
  - [0] No = (Default) The device creates a new Call-ID value for the outgoing message.  
  - [1] Yes = The device uses the same Call-ID value received in the incoming message for the Call-ID in the outgoing message.  
  **Note:** When the device sends an INVITE as a result of a REFER/3xx termination, the device always creates a new Call-ID value and ignores the parameter's settings. |
| Dial Plan sbc-dial-plan-name [IPGroup_SBCDialPlanName] | Assigns a Dial Plan to the IP Group. The device searches the Dial Plan for a dial plan rule that matches the source number and if not found, for a rule that matches the destination number. If a matching dial plan rule is found, the rule's tag is used in the routing and/or manipulation processes as source and/or destination tags.  
To configure Dial Plans, see Configuring Dial Plans on page 541.  
**Note:** For IP-to-IP Routing rules that are configured for destination based on tags (i.e., 'Destination Type' parameter configured to Destination Tag), the parameter is applicable only to the source IP Group and the device searches the Dial Plan for a dial plan rule that matches the prefix of the destination number only. For more information on routing based on destination tags, see Using Dial Plan Tags for Routing Destinations on page 551. |
| Tags tags [IPGroup_Tags] | Assigns a Dial Plan tag that is used to determine whether the incoming SIP dialog is sent to this IP Group. The parameter is used when IP-to-IP Routing rules are configured for destination based on tags (i.e., 'Destination Type' parameter configured to Destination Tag). For more information on routing based on destination tags, see Using Dial Plan Tags for Routing Destinations on page 551. |
| Call Setup Rules Set ID call-setup-rules-set-id [IPGroup_CallSetupRulesSetId] | Assigns a Call Setup Rule Set ID to the IP Group. The device runs the Call Setup rule immediately before the routing stage (i.e., only after the classification and manipulation stages).  
By default, no value is assigned.  
To configure Call Setup Rules, see Configuring Call Setup Rules on page 384. |
| Quality of Experience |  |
| QoE Profile qoe-profile [IPGroup_QOEProfile] | Assigns a Quality of Experience Profile rule.  
By default, no value is defined.  
To configure Quality of Experience Profiles, see "Configuring Quality of Experience Profiles" on page 307. |
| Bandwidth Profile bandwidth-profile [IPGroup_BWProfile] | Assigns a Bandwidth Profile rule.  
By default, no value is defined.  
To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles" on page 312. |
| Message Manipulation |  |
| Inbound Message Manipulation Set inbound-mesg-manipulation-set [IPGroup_InboundManSet] | Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the inbound leg.  
By default, no value is defined.  
To configure Message Manipulation rules, see Configuring SIP Message Manipulation on page 395. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Outbound Message Manipulation Set</strong></td>
<td>Assigns a Message Manipulation Set (rule) to the IP Group for SIP message manipulation on the outbound leg. By default, no value is defined. To configure Message Manipulation rules, see &quot;Configuring SIP Message Manipulation&quot; on page 393. <strong>Note:</strong> If you assign a Message Manipulation Set ID that includes rules for manipulating the host name in the Request-URI, To, and/or From SIP headers, the parameter overrides the IPGroup_SIPGroupName parameter.</td>
</tr>
</tbody>
</table>
| **Message Manipulation User-Defined String 1** | Defines a value for the SIP user part that can be used in Message Manipulation rules configured in the Message Manipulations table. The Message Manipulation rule obtains this value from the IP Group, by using the following syntax: `param.ipg.<src|dst>.user-defined.<0>`.
- The valid value is a string of up to 30 characters. By default, no value is defined.
- To configure Message Manipulation rules, see "Configuring SIP Message Manipulation" on page 393. |
| **Message Manipulation User-Defined String 2** | 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. By default, no value is defined.
- To configure Message Manipulation rules, see "Configuring SIP Message Manipulation" on page 393. |
| **SBC Registration and Authentication** | Defines the maximum number of users in this IP Group that can register with the device. The default is -1, meaning that no limitation exists for registered users. **Note:** The parameter is applicable only to User-type IP Groups. |
| **Max. Number of Registered Users** | 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 mode. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>

**User Stickiness**  
**sbc-user-stickiness**  
**[IPGroup_SBCUserStickiness]**

Enables user "stickiness" (binding) to a specific registrar server. The registrar server is one of the IP addresses of the Proxy Set associated with this Server-type IP Group. This feature applies to users belonging to a User-type IP Group that are routed to this destination Server-type IP Group.

- **[0] Disable** = After a successful initial registration of the user to a registrar, whenever the device receives a SIP request or registration refresh from the user, the device sends the request to whichever registrar (IP address of the Proxy Set) is currently active. In the case of proxy load-balancing, there is no certainty to which IP address the request is routed.
- **[1] Enable** = The device always routes SIP requests (INVITEs, SUBSCRIBEs and REGISTER refreshes) received from the user to the same registrar server to which the last successful REGISTER request for that user was routed. In other words, once initial registration of the user to one of the IP addresses of the Proxy Set associated with this destination Server-type IP Group is successful (i.e., 200 OK), binding occurs to this specific address (registrar) and all future SIP requests from the user are routed (based on matched routing rule) only to this specific registrar.

**Note:**

- The parameter is applicable only to Server-type IP Groups.
- The Proxy Set associated with the Server-type IP Group must be configured with multiple IP addresses (or an FQDN that resolves into multiple IP addresses).
- This feature is also applicable to IP Group Sets (see Configuring IP Group Sets on page 525). If a user is bound to a registrar associated with this Server-type IP Group which also belongs to an IP Group Set, IP Group Set logic of choosing an IP Group is ignored and instead, the device always routes requests from this user to this specific registrar.
- A user's "stickiness" to a specific registrar ends upon the following scenarios:
  - If you modify the Proxy Set.
  - If the Proxy Set is configured with an FQDN and a DNS resolution refresh removes the IP address to which the user is bound.
  - User registration expires or the user initiates an unregister request.
- The Proxy Set's Hot-Swap feature (for proxy redundancy) is not supported for users that are already bound to a registrar. However, you can achieve proxy "hot-swap" for failed initial (non-bounded) REGISTER requests. If a failure response is received for the initial REGISTER request and the response’s code appears in the Alternative Routing Reasons table (see Configuring SIP Response Codes for Alternative Routing Reasons on page 519), "hot-swap" to the other IP addresses of the Proxy Set is done until a success response is received from one of the addresses. In the case of failed REGISTER
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>refresh requests from users that are already bound to a registrar, no &quot;hot-swap&quot; occurs for that request; only for subsequent refresh requests.</td>
<td></td>
</tr>
<tr>
<td>▪ When using the User Info table (see SBC User Information for SBC User Database on page 663), registrar &quot;stickiness&quot; is supported only when the user initiates the REGISTER request. Therefore, you must configure the 'Registration Mode' parameter of the IP Group (User-type) to which the user belongs, to <strong>User Initiates Registration</strong>.</td>
<td></td>
</tr>
<tr>
<td>▪ This feature is also supported when the device operates in HA mode; registrar &quot;stickiness&quot; is retained even after an HA switchover.</td>
<td></td>
</tr>
<tr>
<td><strong>Authentication Mode</strong> authentication-mode [IPGroup_AuthenticationMode]</td>
<td>Defines the authentication mode.</td>
</tr>
<tr>
<td>▪ [0] User Authenticates = (Default) The device does not handle the authentication, but simply forwards the authentication messages between the SIP user agents.</td>
<td></td>
</tr>
<tr>
<td>▪ [1] SBC as Client = The device authenticates as a client. It receives the 401/407 response from the proxy requesting for authentication. The device sends the proxy the authorization credentials (i.e., username and password) according to one of the following: 1)Account configured in the Accounts table (only if authenticating Server-type IP Group), 2) global username and password parameters (only if authenticating Server-type IP Group), 3) User Information file, or 4) sends request to users requesting credentials (only if authenticating User-type IP Group). For more information on Accounts, see Configuring Registration Accounts on page 375.</td>
<td></td>
</tr>
<tr>
<td>▪ [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 663). This is applicable only to User-type IP Groups. ✓ Authenticates SIP severs. This is applicable only to Server-type IP Groups.</td>
<td></td>
</tr>
<tr>
<td><strong>Authentication Method List</strong> authentication-method-list [IPGroup_MethodList]</td>
<td>Defines SIP methods received from the IP Group that must be challenged by the device when the device acts as an Authentication server. If no methods are configured, the device doesn't challenge any methods.</td>
</tr>
<tr>
<td>By default, no value is defined. To define multiple SIP methods, use the backslash ( \ ) to separate each method (e.g., INVITE\REGISTER). <strong>Note:</strong> The parameter is applicable only if the 'Authentication Mode' parameter is set to SBC as Server [2].</td>
<td></td>
</tr>
<tr>
<td><strong>Username</strong> username [IPGroup_Username]</td>
<td>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. <strong>Note:</strong></td>
</tr>
<tr>
<td>▪ The parameter is applicable only to Server-type IP Groups and when the 'Authentication Mode' parameter is set to SBC as Server (i.e., authentication of servers).</td>
<td></td>
</tr>
</tbody>
</table>
19. Control Network

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▪ To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.</td>
</tr>
<tr>
<td>Password</td>
<td>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. <strong>Note:</strong> ▪ The parameter is applicable only to Server-type IP Groups and when the 'Authentication Mode' parameter is set to SBC as Server (i.e., authentication of servers). ▪ To specify the SIP request types (e.g., INVITE) that must be challenged by the device, use the 'Authentication Method List' parameter.</td>
</tr>
</tbody>
</table>

**GW Group Status**

| GW Group Registered IP Address | (Read-only field) Displays the IP address of the IP Group entity (gateway) if registered with the device; otherwise, the field is blank. **Note:** The field is applicable only to Gateway-type IP Groups (i.e., the 'Type' parameter is configured to Gateway). |
| GW Group Registered Status     | (Read-only field) Displays whether the IP Group entity (gateway) is registered with the device ("Registered" or "Not Registered"). **Note:** The field is applicable only to Gateway-type IP Groups (i.e., the 'Type' parameter is configured to Gateway). |

19.5 Configuring Proxy Sets

The Proxy Sets table lets you configure up to 5,000 Proxy Sets. A Proxy Set defines the address and transport type (e.g., UDP or TCP) of a SIP server (e.g., SIP proxy and SIP registrar server). The Proxy Set represents the destination (address) of the IP Group configuration entity. Each Proxy Set can be configured with up to 10 addresses configured as an IP address and/or DNS host name (FQDN), enabling you to implement load balancing and redundancy (Proxy Hot-Swap feature) between multiple servers. Each Proxy Set can be assigned a specific SSL/TLS certificate (the TLS Context), enabling you to use different TLS certificates per SIP entity (IP Group). In addition, each Proxy Set must be assigned a SIP Interface (and SRD), which determines, amongst others, the device’s local network interface through which communication with the Proxy Set is done.

You can enable the device’s keep-alive feature per Proxy Set, which determines whether proxies (addresses) configured for the Proxy Set are online or offline. If offline, the device will not route the call to the specific proxy. You can configure the device to send either SIP OPTIONS or REGISTER messages for the keep-alive. The keep-alive feature is required when using the proxy load-balancing or redundancy feature. For load-balancing, the device performs keep-alive on all proxies. For Parking-type redundancy, the device performs keep-alive only on the currently active proxy. For Homing-type redundancy, the device performs keep-alive on the current proxy as well as the "main" proxy. When using SIP OPTIONS, you can configure the device to consider the proxy as offline if specific SIP response codes are received from the keep-alive messages. To ensure that a previously offline proxy is now online, you can configure the number of required consecutive successful keep-alive messages (SIP OPTIONS only) before the device considers the proxy as being online. This mechanism avoids the scenario in which the device falsely
detects a proxy as being online when it is actually offline, resulting in call routing failure. To view the connectivity status of Proxy Sets, see Viewing Proxy Set Status on page 750.

You can also enable the device to classify incoming SBC SIP dialogs to IP Groups, based on Proxy Set. If the source address of the incoming SIP dialog is the same as the address of a Proxy Set, the device classifies the SIP dialog as belonging to the IP Group that is associated with the Proxy Set.

To use a configured Proxy Set, you need to assign it to an IP Group in the IP Groups table (see "Configuring IP Groups" on page 345). When the device sends INVITE messages to an IP Group, it sends it to the address configured for the Proxy Set. You can assign the same Proxy Set to multiple IP Groups (belonging to the same SRD).

**Note:**
- It is recommended to classify incoming SIP dialogs to IP Groups, based on the Classification table (see Configuring Classification Rules on page 493) instead of based on Proxy Set.
- You can view the device's connectivity status with proxy servers in the Tel-to-IP Routing table, for Tel-to-IP routing rules whose destination is an IP Group that is associated with a Proxy Set. The status is only displayed for Proxy Sets enabled with the Proxy Keep-Alive feature.
- The device supports up to 5,000 Proxy Sets only when it has 32-GB or 64-GB RAM. For less than 32-GB RAM, the maximum is 1,500 Proxy Sets.

The Proxy Set is configured using two tables, one a "child" of the other:

- **Proxy Sets table:** Defines the attributes of the Proxy Set such as associated SIP Interface and redundancy features - ini file parameter, ProxySet or CLI command, configure voip > proxy-set
- **Proxy Set Address table ("child"):** Defines the addresses of the Proxy Set - table ini file parameter, ProxyIP or CLI command, configure voip > proxy-ip > proxy-set-id

➢ **To configure a Proxy Set:**

1. Open the Proxy Sets table (Setup menu > Signaling & Media tab > Core Entities folder > Proxy Sets).
2. Click New; the following dialog box appears:
3. Configure a Proxy Set according to the parameters described in the table below.
4. Click **Apply**.
5. Select the index row of the Proxy Set that you added, and then click the **Proxy Address** link located below the table; the Proxy Address table opens.
6. Click **New**; the following dialog box appears:

   **Figure 19-10: Proxy Address Table - Add Dialog Box**

7. Configure the address of the Proxy Set according to the parameters described in the table below.
8. Click **Apply**.

**Table 19-7: Proxy Sets Table and Proxy Address Table Parameter Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRD</td>
<td>Assigns an SRD to the Proxy Set.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| voip-network proxy-set > srd-id [ProxySet_SRDName]                        | **Note:**  
  - The parameter is mandatory and must be configured first before you can configure the other parameters in the table.  
  - To configure SRDs, see Configuring SRDs on page 327.                                                                                           |

**General**

| Index | configure voip > voip-network proxy-set [ProxySet_Index] | Defines an index number for the new table row.  
**Note:** Each row must be configured with a unique index. |
|-------|--------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Name  | proxy-name [ProxySet_ProxyName]                        | Defines a descriptive name, which is used when associating the row in other tables.  
The valid value is a string of up to 40 characters.  
**Note:** Each row must be configured with a unique name. |
| SBC IPv4 SIP Interface | sbcipv4-sip-int-name [ProxySet_SBCIPv4SIPInterface Name] | Assigns an IPv4-based SIP Interface for SBC calls to the Proxy Set.  
**Note:**  
  - At least one SIP Interface must be assigned to the Proxy Set.  
  - The parameter appears only if you have configured a network interface with an IPv4 address in the IP Interfaces table (see Configuring IP Network Interfaces on page 137).  
  - To configure SIP Interfaces, see "Configuring SIP Interfaces" on page 337. |
| SBC IPv6 SIP Interface | sbcipv6-sip-int-name [ProxySet_SBCIPv6SIPInterface Name] | Assigns an IPv6-based SIP Interface for SBC calls to the Proxy Set.  
**Note:**  
  - At least one SIP Interface must be assigned to the Proxy Set.  
  - The parameter appears only if you have configured a network interface with an IPv6 address in the IP Interfaces table. |
| TLS Context Index | tls-context-name [ProxySet_TLSContextName] | Assigns a TLS Context (SSL/TLS certificate) to the Proxy Set.  
By default, no TLS Context is assigned. If you assign a TLS Context, the TLS Context is used as follows:  
- **Incoming calls:** If the ‘Transport Type’ parameter (in this table) is set to **TLS** and the incoming call is successfully classified to an IP Group based on the Proxy Set, this TLS Context is used. If the ‘Transport Type’ parameter is set to **UDP** or classification to this Proxy Set fails, the TLS Context is not used. Instead, the device uses the TLS Context configured for the SIP Interface (see "Configuring SIP Interfaces" on page 337) used for the call; otherwise, the default TLS Context (ID 0) is used.  
- **Outgoing calls:** If the ‘Transport Type’ parameter is set to **TLS** and the outgoing call is sent to an IP Group that is associated with this Proxy Set, this TLS Context is used. Instead, the device uses the TLS Context configured for the SIP Interface used for the call; otherwise, the default TLS Context (ID 0) is used. If the ‘Transport Type’ parameter is set to **UDP**, the device uses UDP to communicate with the proxy and no TLS Context is used.  
To configure TLS Contexts, see "Configuring TLS Certificate" |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Keep Alive</strong></td>
<td>Enables the device's Proxy Keep-Alive feature, which checks communication with the proxy server.</td>
</tr>
<tr>
<td>Proxy Keep-Alive</td>
<td>Enables the device's Proxy Keep-Alive feature, which checks communication with the proxy server.</td>
</tr>
<tr>
<td>proxy-enable-keep-alive</td>
<td>Enables the device's Proxy Keep-Alive feature, which checks communication with the proxy server.</td>
</tr>
<tr>
<td>[ProxySet_EnableProxyKeepAlive]</td>
<td>Enables the device's Proxy Keep-Alive feature, which checks communication with the proxy server.</td>
</tr>
<tr>
<td><strong>Proxy Keep-Alive Time</strong></td>
<td>Defines the interval (in seconds) between keep-alive messages sent by the device when the Proxy Keep-Alive feature is enabled (see the 'Proxy Keep-Alive' parameter in this table). The valid range is 5 to 2,000,000. The default is 60.</td>
</tr>
<tr>
<td>proxy-keep-alive-time</td>
<td>Defines the interval (in seconds) between keep-alive messages sent by the device when the Proxy Keep-Alive feature is enabled (see the 'Proxy Keep-Alive' parameter in this table). The valid range is 5 to 2,000,000. The default is 60.</td>
</tr>
<tr>
<td>[ProxySet_ProxyKeepAliveTime]</td>
<td>Defines the interval (in seconds) between keep-alive messages sent by the device when the Proxy Keep-Alive feature is enabled (see the 'Proxy Keep-Alive' parameter in this table). The valid range is 5 to 2,000,000. The default is 60.</td>
</tr>
<tr>
<td><strong>Keep-Alive Failure Responses</strong></td>
<td>Defines SIP response codes that if any is received in response to a keep-alive message using SIP OPTIONS, the device considers the proxy as down. Up to three response codes can be configured, where each code is separated by a comma (e.g., 407,404). By default, no response code is defined. If no response code is configured, or if response codes received are not those configured, the proxy is considered</td>
</tr>
<tr>
<td>keepalive-fail-resp</td>
<td>Defines SIP response codes that if any is received in response to a keep-alive message using SIP OPTIONS, the device considers the proxy as down. Up to three response codes can be configured, where each code is separated by a comma (e.g., 407,404). By default, no response code is defined. If no response code is configured, or if response codes received are not those configured, the proxy is considered</td>
</tr>
<tr>
<td>[ProxySet_KeepAliveFailureResp]</td>
<td>Defines SIP response codes that if any is received in response to a keep-alive message using SIP OPTIONS, the device considers the proxy as down. Up to three response codes can be configured, where each code is separated by a comma (e.g., 407,404). By default, no response code is defined. If no response code is configured, or if response codes received are not those configured, the proxy is considered</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>&quot;alive&quot;.</td>
<td>Note: The SIP 200 response code is not supported for this feature.</td>
</tr>
<tr>
<td>Success Detection Retries</td>
<td>Defines the minimum number of consecutive, successful keep-alive messages that the device sends to an offline proxy, before the device considers the proxy as being online. The interval between the sending of each consecutive successful keep-alive is configured by the 'Success Detection Interval' parameter (see below). For an example of using this parameter, see the 'Success Detection Interval' parameter. The valid range is 1 to 100. The default is 1. Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options.</td>
</tr>
<tr>
<td>Success Detection Interval</td>
<td>Defines the interval (in seconds) between each keep-alive retries (as configured by the 'Success Detection Retries' parameter) that the device performs for offline proxies. The valid range is 1 to 200. The default is 10. For example, assume that the 'Success Detection Retries' parameter is configured to 3 and the 'Success Detection Interval' parameter to 5 (seconds). When connectivity is lost with the proxy, the device sends keep-alive messages to the proxy. If the device receives a successful response from the proxy, it sends another (1st) keep-alive after 5 seconds, and if successful, sends another (2nd) keep-alive after 5 seconds, and if successful, sends another (3rd) keep-alive after 5 seconds, and if successful, considers connectivity with the proxy as being restored. Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options.</td>
</tr>
<tr>
<td>Failure Detection Retransmissions</td>
<td>Defines the maximum number of UDP retransmissions that the device sends to an offline proxy, before the device considers the proxy as being offline. The valid range is -1 to 255. The default is -1 (i.e., the settings of the global parameter SIPMaxRtx is applied). Note: The parameter is applicable only if the 'Proxy Keep-Alive' parameter is set to Using Options.</td>
</tr>
<tr>
<td>Redundancy</td>
<td>Determines whether the device switches from a redundant proxy to the primary proxy when the primary proxy becomes available again.</td>
</tr>
<tr>
<td>Redundancy Mode</td>
<td></td>
</tr>
<tr>
<td>[-1] = Not configured (Default). Proxy redundancy method is according to the settings of the global parameter, ProxyRedundancyMode.</td>
<td></td>
</tr>
<tr>
<td>[0] Parking = The device continues operating with the redundant (now active) proxy even if the primary proxy returns to service. If the redundant proxy subsequently becomes unavailable, the device operates with the next configured redundant proxy.</td>
<td></td>
</tr>
<tr>
<td>[1] Homing = The device always attempts to operate with the primary proxy. The device switches back to the primary proxy whenever it becomes available.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td>- To enable this functionality, you must also enable the Proxy Keep-Alive feature (see the 'Proxy Keep-Alive' parameter in this table).</td>
<td></td>
</tr>
<tr>
<td>- The Homing option can only be used if the 'Proxy Keep-Alive' parameter is set to <strong>Using Options</strong>.</td>
<td></td>
</tr>
</tbody>
</table>

### Proxy Hot Swap

**is-proxy-hot-swap**

[ProxySet_IsProxyHotSwap]

Enables the Proxy Hot-Swap feature, whereby the device switches to a redundant proxy upon a failure in the primary proxy (no response is received).

- **[0]** Disable = (Default) Disables the Proxy Hot-Swap feature. If a failure occurs in the primary proxy, the device does not connect with any other address (proxy) configured for the Proxy Set.
- **[1]** Enable = The device sends SIP INVITE/REGISTER messages to the first address listed in the Proxy Address table that is configured for the Proxy Set. If a SIP response is received and this response code is configured in the Alternative Routing Reasons table (see Configuring SIP Response Codes for Alternative Routing Reasons on page 519) for SBC, the device assumes that the proxy is down and sends the message to the next available proxy (address) in the list.

### Proxy Load Balancing Method

**proxy-load-balancing-method**

[ProxySet_ProxyLoadBalancingMethod]

Enables load balancing between proxy servers of the Proxy Set.

- **[0]** Disable = (Default) Disables proxy load balancing.
- **[1]** Round Robin = A list of all possible proxy IP addresses is compiled. This list includes all IP addresses of the Proxy Set after DNS resolutions (including NAPTR and SRV, if configured). After this list is compiled, the Proxy Keep-Alive feature (enabled by the 'Proxy Keep-Alive' and 'Proxy Keep-Alive Time' parameters in this table) tags each entry as "offline" or "online". Load balancing is only performed on proxy servers that are tagged as "online". All outgoing messages are equally distributed across the list of IP addresses. REGISTER messages are also distributed unless a RegistrarIP is configured. The IP address list is refreshed every user-defined interval (see the ProxyIPListRefreshTime parameter). If a change in the order of the IP address entries in the list occurs, all load statistics are erased and balancing starts over again.
- **[2]** Random Weights = The outgoing requests are not distributed equally among the Proxies. The weights are received from the DNS server, using SRV records. The device sends the requests in such a fashion that each proxy receives a percentage of the requests according to its assigned weight. A single FQDN should be configured as a proxy IP address. Random Weights Load Balancing is not used in the following scenarios:
  - More than one IP address has been configured for the Proxy Set.
  - The proxy address is not configured as an FQDN (only IP address).
  - SRV is disabled (see the DNSQueryType parameter).
  - The SRV response includes several records with a
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Min. Active Servers for Load Balancing</strong>&lt;br&gt;min-active-serv-lb [ProxySet_MinActiveServersLB]**</td>
<td>Defines the minimum number of proxies in the Proxy Set that must be online for the device to consider the Proxy Set as online, when proxy load balancing is used. The valid value is 1 to 15. The default is 1. <strong>Note:</strong> The parameter is applicable only if proxy load balancing is enabled (see the 'Proxy Load Balancing Method' parameter, above).</td>
</tr>
</tbody>
</table>
| **Advanced Classification Input**<br>classification-input [ProxySet_ClassificationInput]** | Defines how the device classifies incoming IP calls to the Proxy Set.  
  - [0] IP Address Only = (Default) Classifies calls to the Proxy Set according to IP address only.  
  - [1] IP Address, Port & Transport Type = Classifies calls to the Proxy Set according to IP address, port, and transport type. **Note:**  
    - The parameter is applicable only if the IP Groups table's parameter, 'Classify by Proxy Set' is set to Enable (see Configuring IP Groups on page 345).  
    - If more than one Proxy Set is configured with the same IP address and associated with the same SIP Interface, the device may classify and route the SIP dialog to an incorrect IP Group. In such a scenario, a warning is generated in the Syslog message. However, if some Proxy Sets are configured with the same IP address but different ports (e.g., 10.1.1.1:5060 and 10.1.1.1:5070) and the parameter is configured to IP Address, Port & Transport Type, classification to the correct IP Group is achieved. Therefore, when classification is by Proxy Set, pay attention to the configured IP addresses and this parameter. When more than one Proxy Set is configured with the same IP address, the device selects the matching Proxy Set in the following order:  
      - Selects the Proxy Set whose IP address, port, and transport type match the source of the incoming dialog (regardless of the settings of this parameter).  
      - If no match is found for above, it selects the Proxy Set whose IP address and transport type match the source of the incoming dialog (if the parameter is configured to IP Address Only).  
      - If no match is found for above, it selects the Proxy Set whose IP address match the source of the incoming dialog (if the parameter is configured to IP Address Only). |
| **DNS Resolve Method**<br>dns-resolve-method [ProxySet_DNSResolveMethod]** | Defines the DNS query record type for resolving the proxy server's host name (FQDN) into an IP address(es).  
  - [-1] = Not configured. DNS resolution method is according to the settings of the global parameter, ProxyDNSQueryType.  
  - [0] A-Record = (Default) DNS A-record query is used to resolve DNS to IP addresses.  
  - [1] SRV = If the proxy address is configured with a domain name without a port (e.g., domain.com), an SRV query is done. The SRV query returns the host names (and their weights). The device then performs DNS A-record queries per
### Parameter | Description
--- | ---
host name (according to the received weights). If the configured proxy address contains a domain name with a port (e.g., domain.com:5080), the device performs a regular DNS A-record query.

- **[2]** NAPTR = NAPTR query is done. If successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is done according to the configured transport type. If the configured proxy address contains a domain name with a port (e.g., domain.com:5080), the device performs a regular DNS A-record query. If the transport type is configured for the proxy address, a NAPTR query is not performed.

- **[3]** Microsoft Skype for Business = SRV query as required by Microsoft when the device is deployed in a Microsoft Skype for Business environment. The device sends a special SRV query to the DNS server according to the transport protocol configured in the 'Transport Type' parameter (described later in this section):
  - **TLS:** 
    - 
    - "_sipinternaltls_tcp.<domain>" and "_sip_tls.<domain>". For example, if the configured domain name (in the 'Proxy Address' parameter) is "ms-server.com", the device queries for "_sipinternaltls_tcp.ms-server.com" and "_sip_tls.ms-server.com".
  - **TCP:** 
    - 
    - "_sipinternal._tcp.<domain>" and "_sip_tcp.<domain>".
  - **Undefined:** 
    - 
    - "_sipinternaltls_tcp.<domain>", "_sipinternal._tcp.<domain>", "_sip_lts.<domain>" and "_sip_tcp.<domain>".

The SRV query returns the host names (and their weights). The device then performs DNS A-record queries per host name (according to the received weights) to resolve into IP addresses.

**Note:** An SRV query can return up to four host names. For each host name, the subsequent DNS A-record query can resolve into up to 15 IP addresses. However, the device supports up to 30 DNS-resolved IP addresses. If the device receives more than this number of IP addresses, it uses the first 30 IP addresses in the received list and ignores the rest.

### Proxy Address Table

<table>
<thead>
<tr>
<th>Index</th>
<th>Defines an index number for the new table row.</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy-ip-index</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>[ProxyIp_ProxyIpIndex]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Proxy Address</th>
<th>Defines the address of the proxy.</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy-address</td>
<td>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:</td>
</tr>
<tr>
<td>[ProxyIp_IpAddress]</td>
<td></td>
</tr>
<tr>
<td>IPv4 address: &lt;IP address&gt;:&lt;port&gt; (e.g., 201.10.8.1:5060)</td>
<td>IPv6 address: &lt;[IPV6 address]&gt;::&lt;port&gt; (e.g., [2000::1:200:200:86:14]:5060)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td><strong>When configured with an FQDN, you can configure the periodic rate at which the device performs DNS queries to resolve the FQDN into IP addresses. For more information, see the ProxyIPListRefreshTime parameter.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>When configured with an FQDN, you can configure the method (e.g., A-record) for resolving the domain name into an IP address, using the 'DNS Resolve Method' parameter in this table (see above).</strong></td>
</tr>
<tr>
<td></td>
<td><strong>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.</strong></td>
</tr>
<tr>
<td></td>
<td><strong>You can configure the device to use the port indicated in the Request-URI of the incoming message, instead of the port configured for the parameter. To enable this, use the IPGroup_SBCRouteUsingRequestURIPort parameter for the IP Group that is associated with the Proxy Set (Configuring IP Groups on page 345).</strong></td>
</tr>
<tr>
<td>Transport Type</td>
<td>Defines the transport type for communicating with the proxy.</td>
</tr>
<tr>
<td>transport-type</td>
<td></td>
</tr>
<tr>
<td>[ProxyIp_TransportType]</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>[0] UDP</strong> <strong>[1] TCP</strong> <strong>[2] TLS</strong> <strong>[-1] = (Default) The transport type is according to the settings of the global parameter, SIPTransportType.</strong></td>
</tr>
</tbody>
</table>
19.6 Building and Viewing SIP Entities in Topology View

The Topology view lets you easily build and view your main SIP entities, including IP Groups, SIP Interfaces, and Media Realms. The Topology view graphically displays these entities and the associations between them, giving you a better understanding of your SIP topology and configuration. The Topology view also lets you configure additional SIP settings that are important to your deployment such as routing and manipulation. You can use the Topology view as an alternative to configuring the entities in their respective Web pages or you can use it in combination.

➢ To access the Topology view:

〢 Click the Topology View home icon (Setup menu > Signaling & Media tab > Topology View).

The main areas of the Topology view is shown below and described in the subsequent table.

Figure 19-11: Areas of Topology View

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Demarcation area of the topology. By default, the Topology view displays the following names to represent the different demarcations of your voice configuration:</td>
</tr>
<tr>
<td></td>
<td>➢ &quot;WAN&quot;: Indicates the external network side</td>
</tr>
<tr>
<td></td>
<td>➢ &quot;LAN&quot;: Indicates the internal network (e.g., inside the Enterprise)</td>
</tr>
<tr>
<td></td>
<td>To modify a demarcation name, do the following:</td>
</tr>
<tr>
<td></td>
<td>1 Click the demarcation name; the name becomes editable in a text box, as shown in the example below:</td>
</tr>
<tr>
<td>Item #</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>2</td>
<td>Type a name as desired, and then click anywhere outside of the text box to apply the name. You can use demarcation to visually separate your voice network to provide a clearer understanding of your topology. This is especially useful for IP Groups, SIP Interfaces, and Media Realms, where you can display them on the top or bottom border of the Topology view (as shown in the figure below for callouts #1 and #2, respectively). For example, on the top border you can position all entities relating to WAN, and on the bottom border all entities relating to LAN.</td>
</tr>
<tr>
<td></td>
<td><img src="image1" alt="Figure 19-12: Display Location in Topology View" /></td>
</tr>
<tr>
<td></td>
<td><img src="image2" alt="Figure 19-13: Configuration Position in Topology View" /></td>
</tr>
<tr>
<td>2</td>
<td>Configured SIP Interfaces. Each SIP Interface is displayed using the following &quot;SIP Interface&quot;-titled icon, which includes the name and row index number: If you hover your mouse over the icon, a pop-up appears displaying the following basic information (example):</td>
</tr>
<tr>
<td></td>
<td><img src="image3" alt="SIP Interface Icon" /></td>
</tr>
<tr>
<td></td>
<td>If you click the icon, a drop-down menu appears listing the following commands:</td>
</tr>
</tbody>
</table>
### Item # Description

- **Edit**: Opens a dialog box in the SIP Interfaces table to modify the SIP Interface.
- **Show List**: Opens the SIP Interfaces table.
- **Delete**: Opens the SIP Interfaces table where you are prompted to confirm deletion of the SIP Interface.

To add a SIP Interface, do the following:

1. Click the **Add SIP Interface** plus icon. The icon appears next to existing SIP Interfaces, or as when no SIP Interfaces exist on a topology border, or as when there are no SIP Interfaces at all.

The SIP Interfaces table opens with a new dialog box for adding a SIP Interface to the next available index row.

2. Configure the SIP Interface as desired, and then click **Apply**; the SIP Interfaces table closes and you are returned to the Topology View, displaying the new SIP Interface.

For more information on configuring SIP Interfaces, see "Configuring SIP Interfaces" on page 337.

3. **Configured Media Realms. Each Media Realm is displayed using the following "Media Realm"-titled icon, which includes the name and row index number:**

If you hover your mouse over the icon, a pop-up appears displaying the following basic information (example):

If you click the icon, a drop-down menu appears listing the following commands:

- **Edit**: Opens a dialog box in the Media Realms table to modify the Media Realm.
- **Show List**: Opens the Media Realms table.
- **Delete**: Opens the Media Realms table where you are prompted to confirm deletion of the Media Realm.

To add a Media Realm, do the following:

1. Click the **Add Media Realm** plus icon. The icon appears next to existing Media Realms, or as when no Media Realms exist on a topology.
The Media Realms table opens with a new dialog box for adding a Media Realm to the next available index row.

2 Configure the Media Realm as desired, and then click **Apply**; the Media Realms table closes and you are returned to the Topology View, displaying the new Media Realm.

For more information on configuring Media Realms, see "Configuring Media Realms" on page 319.

4 Configured IP Groups. Each IP Group is displayed using the following "IP Group [Server]" or "IP Group [User]" titled icon (depending on whether it's a Server- or User-type IP Group respectively), which includes the name and row index number (example of a Server-type):

If you hover your mouse over the icon, a pop-up appears displaying the following basic information (example):

If you click the icon, a drop-down menu appears listing the following commands:

- **Edit**: Opens a dialog box in the IP Groups table to modify the IP Group.
- **Show List**: Opens the IP Groups table.
- **Delete**: Opens the IP Groups table where you are prompted to confirm deletion of the IP Group.

To add an IP Group, do the following:

1 Click the **Add IP Group** plus icon. The icon appears next to existing IP Groups, or as when no IP Groups exist on a topology border, or as when there are no IP Groups at all.

The IP Groups table opens with a new dialog box for adding a IP Group to the next available index row.

2 Configure the IP Group as desired, and then click **Apply**; the IP Groups table closes and you are returned to the Topology View, displaying the new IP Group.

For more information on configuring IP Groups, see “Configuring IP Groups” on page 345.

IP Group icons also display connectivity status with Server-type IP Groups:

- (Green with check mark): Keep-alive is successful and connectivity exists with IP Group.
- (Red with "x"): Keep-alive has failed and there is a loss of connectivity with the IP Group.
The line type connecting between an IP Group and a SIP Interface indicates whether a routing rule has been configured for the IP Group. A solid line indicates that you have configured a routing rule for the IP Group; a dashed line indicates that you have yet to configure a routing rule.

**Note:**
- You can also view connectivity status in the IP Groups table.
- To support the connectivity status feature, you must enable the keep-alive mechanism for the Proxy Set that is associated with the IP Group (see "Configuring Proxy Sets" on page 359).
- The green-color state also applies to scenarios where the device rejects calls with the IP Group due to low QoE (e.g., low MOS), despite connectivity.

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>Links to Web pages relating to commonly required SBC configuration:</td>
</tr>
<tr>
<td></td>
<td>Classification: Opens the Classification table where you can configure Classification rules (see &quot;Configuring Classification Rules&quot; on page 493).</td>
</tr>
<tr>
<td></td>
<td>Number Manipulation: Opens the Outbound Manipulations table where you can configure manipulation rules on SIP Request-URI user parts (source or destination) or calling names in outbound SIP dialog requests (see &quot;Configuring IP-to-IP Outbound Manipulations&quot; on page 535).</td>
</tr>
<tr>
<td></td>
<td>Routing: Opens the IP-to-IP Routing table where you can configure IP-to-IP routing rules (see &quot;Configuring SBC IP-to-IP Routing Rules&quot; on page 501).</td>
</tr>
<tr>
<td></td>
<td>SBC Settings: Opens the SBC General Settings page where you can configure miscellaneous settings.</td>
</tr>
</tbody>
</table>
This page is intentionally left blank.
20 SIP Definitions

This section describes configuration of various SIP-related functionalities.

20.1 Configuring Registration Accounts

The Accounts table lets you configure up to 1,500 Accounts. An Account defines registration information for registering and authenticating (digest) IP Groups (e.g., IP PBX) with a "serving" IP Group (e.g., ITSP).

The device initiates registration with a "serving" IP Group on behalf of the "served" IP Group. Therefore, Accounts are typically required when the "served" IP Group is unable to register or authenticate itself for whatever reason. Registration information includes username, password, host name (AOR), and contact user name (AOR). The device includes this information in the REGISTER message sent to the serving IP Group. Up to 10 Accounts can be configured per "served" IP Group. A IP Group can register to more than one IP Group (e.g., multiple ITSPs). This is done by configuring multiple entries in the Accounts table for the same served IP Group, but with different serving IP Groups, username/password, host name, and contact user values.

Authentication is typically required for INVITE messages sent to the "serving" IP Group. If the device receives a SIP 401 (Unauthorized) in response to a sent INVITE, the device checks for a matching "serving" and "served" entry in the Accounts table. If a matching row exists, the device authenticates the INVITE by providing the corresponding MD5 authentication username and password to the "serving" IP Group.

**Note:**
- If no match is found in the Accounts table for incoming or outgoing calls, the username and password is taken from:
  - 'UserName' and 'Password' parameters on the Proxy & Registration page
- See also the following optional, related parameters:
  - UseRandomUser - enables the device to assign a random string to the user part of the SIP Contact header of new Accounts.
  - UnregisterOnStartup - enables the device to unregister and then re-register Accounts upon a device reset.

The following procedure describes how to configure Accounts through the Web interface. You can also configure it through ini file (Account) or CLI (configure voip > sip-definition account).

➢ To configure an Account:

1. Open the Accounts table (Setup menu > Signaling & Media tab > SIP Definitions folder > Accounts).
2. Click **New**; the following dialog box appears:

**Figure 20-1: Accounts Table - Add Dialog Box**

3. Configure an account according to the parameters described in the table below.

4. Click **Apply**.

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 the Account.

To view Account registration status, see "Viewing Registration Status" on page 751.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Served IP Group</td>
<td>Defines the IP Group (e.g., IP-PBX) that you want to register and/or authenticate upon its behalf.</td>
</tr>
<tr>
<td><strong>served-ip-group-name</strong></td>
<td><strong>[Account_ServedIPGroupName]</strong></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>By default, all IP Groups are displayed. However, if you filter the Web display by SRD (using the SRD Filter box), only IP Groups associated with the filtered SRD are displayed.</td>
</tr>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>Defines an index for the new table row.</td>
</tr>
</tbody>
</table>

**Table 20-1: Accounts Table Parameter Descriptions**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Type</td>
<td>Defines the application type:</td>
</tr>
<tr>
<td>Serving IP Group</td>
<td>Defines the IP Group (Serving IP Group) to where the device sends the SIP REGISTER requests (if enabled) for registration and authentication (of the Served IP Group).</td>
</tr>
<tr>
<td>[Account_ServingIPGroupName]</td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>- By default, only IP Groups associated with the SRD to which the Served IP Group is associated are displayed, as well as IP Groups of Shared SRDs. However, if you filter the Web display by SRD (using the SRD Filter box), only IP Groups associated with the filtered SRD are displayed, as well as IP Groups of Shared SRDs.</td>
</tr>
<tr>
<td></td>
<td>- The parameter is mandatory.</td>
</tr>
<tr>
<td>Host Name</td>
<td>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.</td>
</tr>
<tr>
<td>[Account_HostName]</td>
<td><strong>Note:</strong> If the parameter is not configured or if registration fails, the 'SIP Group Name' parameter value configured in the IP Groups table is used instead.</td>
</tr>
<tr>
<td>Register</td>
<td>Enables registration.</td>
</tr>
<tr>
<td>[Account_Register]</td>
<td>- [0] No= (Default) The device only performs authentication (not registration). Authentication is typically done for INVITE messages sent to the &quot;serving&quot; IP Group. If the device receives a SIP 401 (Unauthorized) in response to a sent INVITE, the device checks for a matching &quot;serving&quot; and &quot;served&quot; entry in the table. If a matching row exists, the device authenticates the INVITE by providing the corresponding MD5 authentication username and password to the &quot;serving&quot; IP Group.</td>
</tr>
<tr>
<td></td>
<td>- [1] Regular = Regular registration process. For more information, see &quot;Regular Registration Mode&quot; on page 380.</td>
</tr>
<tr>
<td></td>
<td>- [2] GIN = Registration for legacy PBXs, using Global Identification Number (GIN). For more information, see &quot;Single Registration for Multiple Phone Numbers using GIN&quot; on page 380.</td>
</tr>
<tr>
<td>Contact User</td>
<td>Defines the AOR username. This appears in REGISTER From/To headers as ContactUser@HostName, and in INVITE/200 OK Contact headers as ContactUser@&lt;device's IP address&gt;.</td>
</tr>
<tr>
<td>[Account_ContactUser]</td>
<td><strong>Note:</strong> The account registration is not affected by the IsRegisterNeeded parameter.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Registrar Stickiness</td>
<td>Enables the Registrar Stickiness feature, whereby the device always routes SIP requests of a registered Account to the same registrar server to where the last successful REGISTER request was routed. In other words, once initial registration of the Account to one of the IP addresses in the Proxy Set (associated with the Serving IP Group) is successful (i.e., 200 OK), binding (&quot;stickiness&quot;) occurs to this specific address (registrar). All future SIP requests (INVITEs, SUBSCRIBEs and REGISTER refreshes) whose source and destination match the Account are sent to this registrar only. This applies until the registrar is unreachable or registration refresh fails, for whatever reason.</td>
</tr>
<tr>
<td>Registrar Search Mode</td>
<td>Defines the method for choosing an IP address (registrar) in the Proxy Set (associated with the Serving IP Group) to which the Account initially registers and performs registration refreshes, when the Registrar Stickiness feature is enabled. Once chosen, the Account is binded to the IP address for subsequent SIP requests.</td>
</tr>
</tbody>
</table>

**Note:**
- If the parameter is not configured, the 'Contact User' parameter in the IP Groups table is used instead.
- If registration fails, the user part in the INVITE Contact header contains the source party number.
- [0] Disable = (Default) Disables the Registrar Stickiness feature. After successful initial registration of the Account with a registrar (IP address), whenever the device receives a SIP request or registration refresh, the device sends the request to whichever IP address is the currently working registrar. In other words, there is no binding to a specific IP address in the Proxy Set and at any given time, requests may be sent to a different IP address, whichever is the working one. In the case of proxy load-balancing, there is no certainty as to which IP address in the Proxy Set the request is routed.
- [1] Enable = Enables the Registrar Stickiness feature. The parameter is applicable only if you have enabled Account registration ('Register' parameter configured to Regular or GIN).

- [0] Current Working Server = (Default) For each initial and refresh registration request, the device routes to the currently working server in the list of IP addresses (configured or DNS-resolved IP addresses) in the Proxy Set. In the case of proxy load-balancing, the chosen IP address is according to the load-balancing mechanism.
- [1] According to IMS Specifications = For the initial registration request, the device performs DNS resolution if the address of the Proxy Set is configured as an FQDN. It then attempts to register to one of the listed DNS-resolved addresses (or configured IP addresses), starting with the first listed address and...
then going down the list sequentially. If an address results in an unsuccessful registration, the device immediately tries the next address (without waiting any retry timeout). The device goes through the list of addresses until an address results in a successful registration. If registration is unsuccessful for all addresses, the device waits a configured retry time and then goes through the list again. Once initial registration is successful, periodic registration refreshes are performed as usual. In addition to the periodic refreshes, immediate register refreshes are done upon the following triggers according to the IMS specification:

- The device receives a SIP 408, 480, or 403 response from the Serving IP Group in response to an INVITE.
- The transaction timeout expires for an INVITE sent to the Serving IP Group.
- The device receives an INVITE from the Serving IP Group from an IP address other than the address to which it is currently registered. In this case, it also rejects the INVITE with a SIP 480 response.

If the device's physical Ethernet link to the proxy goes down, the device re-registers this Account with the proxy when the link comes up again. Re-registration occurs even if proxy keep-alive is disabled.

**Note:** The parameter is applicable only if you have enabled the Registrar Stickiness feature (in this table):
- 'Register' parameter to Regular or GIN.
- 'Registrar Stickiness' parameter to Enable.

### Reg Event Package Subscription

**Reg Event Package Subscription**

**reg-event-package-subscription**

[Account_RegEventPackageSubscription]

Enables the device to subscribe to the registration event package service (as defined in RFC 3680) with the registrar server (Serving IP Group) to which the Account is successfully registered and binded, when the Registrar Stickiness feature is enabled. The service allows the device to receive notifications of the Accounts registration state change with the registrar.

The device subscribes to the service by sending a SUBSCRIBE message containing the Event header with the value "reg" (Event: reg). Whenever a change occurs in the registration binding state, the registrar notifies the device by sending a SIP NOTIFY message.

- [0] Disable (default)
- [1] Enable

**Note:** The parameter is applicable only if you have enabled the Registrar Stickiness feature (in this table):
- 'Register' parameter to Regular or GIN.
- 'Registrar Stickiness' parameter to Enable.

### Credentials

**User Name**

Defines the digest MD5 Authentication username.
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>user-name [Account_Username]</td>
<td>The valid value is a string of up to 50 characters.</td>
</tr>
<tr>
<td>Password password [Account_Password]</td>
<td>Defines the digest MD5 Authentication password. The valid value is a string of up to 50 characters.</td>
</tr>
</tbody>
</table>

#### 20.1.1 Regular Registration Mode

When you configure the registration mode in the Accounts 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 Accounts 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.7.20A.000.038
Content-Length: 0
```

#### 20.1.2 Single Registration for Multiple Phone Numbers using GIN

When you configure the registration mode in the Accounts 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:
20.1.3 Registrar Stickiness

You can enable the Registrar Stickiness feature per Account. Registrar Stickiness binds an Account to one of the IP addresses (configured or DNS-resolved) in the Proxy Set associated with the Serving IP Group. Once an Account registers successfully to one of the IP addresses (i.e., SIP registrar server) in the Proxy Set, the device routes all subsequent SIP requests (INVITEs, SUBSCRIBEs and REGISTER refreshes) of the Account to this registrar. This applies until the registrar is unreachable or registration refresh fails, for whatever reason.

To configure the Registrar Stickiness feature, use the following parameters in the Accounts table:

- Registrar Stickiness: Enables the feature.
- Registrar Search Mode: Defines the method for choosing an IP address (registrar) in the Proxy Set to which the Account initially registers and performs registration refreshes. Once chosen, the Account is binded to this registrar.
- Reg Event Package Subscription: Enables the device to subscribe to the registration event package service (as defined in RFC 3680) with the registrar to which the Account is registered and binded. The service allows the device to receive notifications of the Accounts registration state change with the registrar.
20.2 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 831. To configure Proxy servers (Proxy Sets), see "Configuring Proxy Sets" on page 359.

Note: To view the registration status of endpoints with a SIP Registrar/Proxy server, see "Viewing Registration Status" on page 751.

To configure the Proxy and registration parameters:
1. Open the Proxy & Registration page (Setup menu > Signaling & Media tab > SIP Definitions folder > Proxy & Registration).
2. Configure the parameters as required.
3. Click Apply.

To register or un-register the device to a Proxy/Registrar:
- Click the Register button to register.
- Click Un-Register button to un-register.

Instead of registering the entire device, you can register specific entities as listed below by using the Register button located on the page in which these entities are configured:
- Accounts - Accounts table (see "Configuring Registration Accounts" on page 375)

20.2.1 SIP Message Authentication Example

The device supports basic and digest (MD5) authentication types, according to SIP RFC 3261. A proxy server might require authentication before forwarding an INVITE message. A Registrar/Proxy server may also require authentication for client registration. A proxy replies to an unauthenticated INVITE with a 407 Proxy Authorization Required response, containing a Proxy-Authenticate header with the form of the challenge. After sending an ACK for the 407, the user agent can then re-send the INVITE with a Proxy-Authorization header containing the credentials.

User agents, Redirect or Registrar servers typically use the SIP 401 Unauthorized response to challenge authentication containing a WWW-Authenticate header, and expect the re-INVITE to contain an Authorization header.

The following example shows the Digest Authentication procedure, including computation of user agent credentials:

1. The REGISTER request is sent to a Registrar/Proxy server for registration:

```
REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip:122@10.1.1.200>;tag=1c17940
To: <sip:122@10.1.1.200>
Call-ID: 634293194@10.1.1.200
User-Agent: Sip-Gateway/Mediant Software SBC/v.7.20A.000.038
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:
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 "b9c45d0234a5abf5ddf5c704029b3cf".

At this time, a new REGISTER request is issued with the following response:

REGISTER sip:10.2.2.222 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.200
From: <sip:122@10.1.1.200>;tag=1c23940
To: <sip:122@10.1.1.200>
Call-ID: 654982194@10.1.1.200
Server: Audiocodes-Sip-Gateway/Mediant Software SBC/v.7.20A.000.038
CSeq: 1 REGISTER
Contact: sip:122@10.1.1.200;
Expires:3600
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
```

20.3 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 provide 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 (Skype for Business) address, and display name. Call Setup rules provide 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.

- **Dial Plan queries:** For SBC calls, you can use Call Setup rules to query the Dial Plan table (see Configuring Dial Plans on page 541) for a specified search key in a specified Dial Plan to obtain the corresponding Dial Plan tag. Call Setup rules can also change (modify) the name of the obtained tag. The device can then route the call
using an IP-to-IP Routing rule (in the IP-to-IP Routing table) that has a matching tag (source or destination). You can also associate a Call Setup rule with an IP Group (in the IP Group table). Once the device classifies the incoming call to a source IP Group, it processes the associated Call Setup rule and then uses the resultant tag to locate a matching IP-to-IP Routing rule. You can also use Call Setup rules for complex routing schemes by using multiple Dial Plan tags. This is typically required when the source and/or destination of the call needs to be categorized with more than one characteristics. For example, tags can be used to categorize calls by department (source user) within a company, where only certain departments are allowed to place international calls.

- 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 multiple Call Setup rules and group them using a Set ID. This lets you apply multiple Call Setup rules on the same call setup dialog. To use your Call Setup rule(s), you need to assign the Set ID to one of the following, using the ‘Call Setup Rules Set ID’ field:

- SBC IP-to-IP routing rules (see Configuring SBC IP-to-IP Routing Rules on page 501)
- IP Groups (see Configuring IP Groups on page 345)

If assigned to an IP Group, the device processes the Call Setup rule for the classified source IP Group immediately before the routing process. If assigned to a routing rule only, the device first locates a matching routing rule for the incoming call, processes the assigned Call Setup Rules Set ID, and then routes the call according to the destination configured for the routing rule. The device uses the routing rule to route the call, depending on the result of the Call Setup Rules Set ID:

- **Rule’s condition is met:** The device performs the rule’s action and then runs the next rule in the Set ID until the last rule or until a rule with an Exit Action Type. If the Exit rule is configured with a “True” Action Value, the device uses the current routing rule. If the Exit rule is configured with a “False” Action Value, the device moves to the next routing rule. If an Exit Action Type is not configured and the device has run all the rules in the Set ID, the default Action Value of the Set ID is “True” (i.e., use the current routing rule).

- **Rule’s condition is not met:** The device runs the next rule in the Set ID. When the device reaches the end of the Set ID and no Exit was performed, the Set ID ends with a “True” result.

You can also configure a Call Setup rule that determines whether the device must discontinue with the Call Setup Rules Set ID and route the call accordingly. This is done using the Exit optional value of the ‘Action Type’ parameter. When used, the ‘Action Value’ parameter can be configured to one of the following strings:

- “true”: Indicates that if the condition is met, the device routes the call according to the selected routing rule. Note that if the condition is not met, the device also uses the selected routing rule, unless the next Call Setup rule in the Set ID has an Exit option configured to “false” for an empty condition.

- “false”: Indicates that if the condition is met, the device attempts to route the call to the next matching routing rule (if configured). If the condition is not met, the device routes the call according to the selected routing rule.

As the default result of a Call Setup rule is always “true”, please adhere to the following guidelines when configuring the ‘Action Type’ field to Exit: If, for example, you want to exit the Call Setup Rule Set ID with "true" when LDAP query result is found and "false" when LDAP query result is not found:

- Incorrect -this rule will always exit with result = True:
  
  **Condition:** ldap.found exists  
  **Action Type:** Exit  
  **Action Value:** True
Correct:

- Single rule:
  
  **Condition:** ldap.found !exists  
  **Action Type:** Exit  
  **Action Value:** False

- Set of rules:
  
  **Condition:** ldap.found exists  
  **Action Type:** Exit  
  **Action Value:** True

  **Condition:** <leave it blank>  
  **Action Type:** Exit  
  **Action Value:** False

**Note:** If the source and/or destination numbers are manipulated by the Call Setup rules, they revert to their original values if the device moves to the next routing rule.

The following procedure describes how to configure Call Setup Rules through the Web interface. You can also configure it through ini file (CallSetupRules) or CLI (configure voip message call-setup-rules).

➢ **To configure a Call Setup rule:**
  
  1. Open the Call Setup Rules table (Setup menu > Signaling & Media tab > SIP Definitions folder > Call Setup Rules).
  2. Click New; the following dialog box appears:

     **Figure 20-2: Call Setup Rules Table - Add Dialog Box**

  3. Configure a Call Setup rule according to the parameters described in the table below.
  4. Click Apply, and then save your settings to flash memory.

**Table 20-2: Call Setup Rules Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table record.</td>
</tr>
<tr>
<td>[CallSetupRules_Index]</td>
<td><strong>Note:</strong> Each rule must be configured with a unique index.</td>
</tr>
<tr>
<td>Rules Set ID</td>
<td>Defines a Set ID for the rule. You can define the same Set ID for multiple rules to create a group of rules. You can configure up to 10 Set IDs, where each Set ID can include up to 10 rules. The Set ID is used to assign the Call Setup rules to a routing</td>
</tr>
<tr>
<td>rules-set-id</td>
<td></td>
</tr>
<tr>
<td>[CallSetupRules_RulesSetID]</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>rule in the routing table.</td>
<td>The valid value is 0 to 9. The default is 0.</td>
</tr>
<tr>
<td>Query Type</td>
<td>Defines the type of query.</td>
</tr>
<tr>
<td>query-type</td>
<td>- [0] None (default)</td>
</tr>
<tr>
<td></td>
<td>To specify an LDAP server or Dial Plan, use the 'Query Target' parameter (see below).</td>
</tr>
<tr>
<td>Query Target</td>
<td>Defines one of the following, depending on the value configured for the 'Query Type' parameter (above):</td>
</tr>
<tr>
<td>query-target</td>
<td>- LDAP: Specifies an LDAP server (LDAP Server Group) on which to perform an LDAP query for a defined search key.</td>
</tr>
<tr>
<td>[CallSetupRules_QueryTarget]</td>
<td>- Dial Plan: Specifies a Dial Plan (name) in which to search for a defined search key.</td>
</tr>
<tr>
<td></td>
<td>To configure LDAP Server Groups, see Configuring LDAP Server Groups on page 238.</td>
</tr>
<tr>
<td></td>
<td>To configure Dial Plans, see Configuring Dial Plans on page on page 541.</td>
</tr>
<tr>
<td></td>
<td>To configure the search key, use the ‘Search Key’ parameter (see below).</td>
</tr>
<tr>
<td>Note:</td>
<td>The parameter is applicable only if the 'Query Type' parameter is configured to any value other than None.</td>
</tr>
<tr>
<td>Search Key</td>
<td>Defines the key to query. For LDAP queries, the key string is queried in the specified LDAP server. For Dial Plan queries, the key string is searched for in the specified Dial Plan.</td>
</tr>
<tr>
<td>attr-to-query</td>
<td>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’).</td>
</tr>
<tr>
<td>[CallSetupRules_AttributesToQuery]</td>
<td>Examples:</td>
</tr>
<tr>
<td></td>
<td>- To LDAP query the AD attribute &quot;mobile&quot; that has the value of the destination user part of the incoming call:</td>
</tr>
<tr>
<td></td>
<td>'mobile=' + param.call.dst.user</td>
</tr>
<tr>
<td></td>
<td>- To LDAP query the AD attribute &quot;telephoneNumber&quot; that has a redirect number:</td>
</tr>
<tr>
<td></td>
<td>'telephoneNumber=' + param.call.redirect + '*'</td>
</tr>
<tr>
<td></td>
<td>- To query a Dial Plan for the source number:</td>
</tr>
<tr>
<td></td>
<td>param.call.src.user</td>
</tr>
<tr>
<td></td>
<td>Note: The parameter is applicable only if the 'Query Type' parameter is configured to any value other than None.</td>
</tr>
<tr>
<td>Attributes To Get</td>
<td>Defines the attributes of the queried LDAP record that the device must handle (e.g., retrieve value).</td>
</tr>
<tr>
<td>attr-to-get</td>
<td>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).</td>
</tr>
<tr>
<td>[CallSetupRules_AttributesToGet]</td>
<td>Note:</td>
</tr>
<tr>
<td></td>
<td>- msRTCSIP-PrivateLine, msRTCSIP-Line, mobile.</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row Role</td>
<td>Determines which condition must be met in order for this rule to be performed.</td>
</tr>
<tr>
<td>row-role</td>
<td>[CallSetupRules_Role]</td>
</tr>
<tr>
<td></td>
<td>[0] Use Current Condition = The Condition configured for this rule must be matched in order to perform the configured action (default).</td>
</tr>
<tr>
<td></td>
<td>[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.</td>
</tr>
<tr>
<td>Condition</td>
<td>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.</td>
</tr>
<tr>
<td>condition</td>
<td>[CallSetupRules_Condition]</td>
</tr>
<tr>
<td>Action</td>
<td>Defines the element (header, parameter, body, or Dial Plan tag) upon which you want to perform the action if the condition, configured in the 'Condition' parameter (see above) is met. The valid value is a string of up to 100 characters (case-insensitive).</td>
</tr>
<tr>
<td>action-subject</td>
<td>[CallSetupRules_ActionSubject]</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| Action Type action-type [CallSetupRules_ActionType] | Defines the type of action to perform.  
- [0] Add (default) = Adds new message header, parameter or body elements.  
- [1] Remove = Removes message header, parameter, or body elements.  
- [3] Add Prefix = Adds value at the beginning of the string (string element only).  
- [4] Add Suffix = Adds value at the end of the string (string element only).  
- [5] Remove Suffix = Removes value from the end of the string (string element only).  
- [6] Remove Prefix = Removes value from the beginning of the string (string element only).  
- [20] Run Rules Set = Performs a different Rule Set ID, specified in the 'Action Value' parameter (below)  
- [21] Exit = Stops the Rule Set ID and returns a result ("True" or "False"). |
| Action Value action-value [CallSetupRules_ActionValue] | Defines a value that you want to use in the action.  
The valid value is a string of up to 300 characters (case-insensitive).  
Examples:  
- '+9723976'+ldap.attr.alternateNumber  
- '9764000'  
- srctags  
- ldap.attr.displayName  
- true (if the 'Action Type' is set to Exit)  
- false (if the 'Action Type' is set to Exit) |
20.3.1 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
  - 'Query Type': LDAP
  - 'Query Target': LDAP-DC-CORP
  - 'Search Key': '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
  - 'Query Type': LDAP
  - 'Query Target': LDAP-DC-CORP
  - 'Search Key': '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 Skype for Business; if the query fails, the device sends the call to the PBX.

• Call Setup Rules table configuration:
  - 'Rules Set ID': 3
  - 'Query Type': LDAP
  - 'Query Target': LDAP-DC-CORP
  - 'Search Key': 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 (Skype for Business). 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 Skype for Business)
  - Index 2:
    - 'Destination IP Group ID': 4 (IP Group of PBX)
Example 4: The example enables routing based on LDAP queries and destination tags. The device queries the LDAP server for the attribute record "telephoneNumber" whose value is the destination number of the incoming call (e.g., "telephoneNumber=4064"). If the attribute-value combination is found, the device retrieves the string value of the attribute record "ofiSBCRouting" and creates a destination tag with the name of the retrieved string. The destination tag is then used as a matching characteristic in the IP-to-IP Routing table.

- Call Setup Rules table:
  - 'Rules Set ID': 4
  - 'Query Type': LDAP
  - 'Query Target': LDAP-DC-CORP
  - 'Search Key': 'telephoneNumber='+param.call.dst.user
  - 'Attributes to Get': ofiSBCRouting
  - 'Row Role': Use Current Condition
  - 'Condition': ldap.found exists
  - 'Action Subject': dsttags
  - 'Action Type': Modify
  - 'Action Value': ldap.attr.ofiSBCRouting

- IP Groups table: 'Call Setup Rules Set ID': 4

- IP-to-IP Routing table:
  - Index 1:
    - 'Destination Tag': dep-sales
    - 'Destination IP Group': SALES
  - Index 2:
    - 'Destination Tag': dep-mkt
    - 'Destination IP Group': MKT
  - Index 3:
    - 'Destination Tag': dep-rd
    - 'Destination IP Group': RD
21 SIP Message Manipulation

This section describes SIP message manipulation.

21.1 Configuring SIP Message Manipulation

The Message Manipulations table lets you configure up to 1,500 Message Manipulation rules. A Message Manipulation rule defines a manipulation sequence for SIP messages. SIP message manipulation enables the normalization of SIP messaging fields between communicating network segments. For example, it allows service providers to design their own policies on the SIP messaging fields that must be present before a SIP call enters their network. Similarly, enterprises and small businesses may have policies for the information that can enter or leave their networks for policy or security reasons from a service provider. SIP message manipulations can also be implemented to resolve incompatibilities between SIP devices inside the enterprise network.

Each Message Manipulation rule is configured with a Manipulation Set ID. You can create groups (sets) of Message Manipulation rules by assigning each of the relevant Message Manipulation rules to the same Manipulation Set ID. The Manipulation Set ID is then used to assign the rules to specific calls:

- Message manipulation rules can be applied pre- or post-classification:
  - Pre-classification Process: Message manipulation can be done on incoming SIP dialog-initiating messages (e.g., INVITE) prior to the classification process. You configure this by assigning the Manipulation Set ID to the SIP Interface on which the call is received (see Configuring SIP Interfaces on page 337).
  - Post-classification Process: Message manipulation can be done on inbound and/or outbound SIP messages after the call has been successfully classified. Manipulation occurs only after the routing process - inbound message manipulation is done first, then outbound number manipulation (see Configuring IP-to-IP Outbound Manipulations on page 535), and then outbound message manipulation. For viewing the call processing flow, see Call Processing of SIP Dialog Requests on page 457. You configure this by assigning the Manipulation Set ID to the relevant IP Group in the IP Groups table (see Configuring IP Groups on page 345).

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)
- Multiple manipulation rules on the same SIP message
- Apply conditions per rule - the condition can be on parts of the message or call's parameters
- Multiple manipulation rules using the same condition. The following figure shows a configuration example where rules 1 and 2 ('Row Rule' configured to Use Previous Condition) use the condition configured for rule 0 ('Row Rule' configured to Use Current Condition). For more information, see the description of the 'Row Rule' parameter in this section.

**Figure 21-1: Configuration Example of Message Manipulation Rules using Same Condition**

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>MANIPULATION SET ID</th>
<th>MESSAGE TYPE</th>
<th>CONDITION</th>
<th>ACTION SUBJECT</th>
<th>ACTION TYPE</th>
<th>ACTION VALUE</th>
<th>ROW RULE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add emergency</td>
<td>0</td>
<td>invite request</td>
<td>header priority: Add</td>
<td>emergency</td>
<td>Use Previous Condition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>User-agent</td>
<td>0</td>
<td>header request</td>
<td>header: request: Cust &lt;&lt;</td>
<td>header: to Modify header: to + Urgent</td>
<td>Use Previous Condition</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>To Header Urgent</td>
<td>0</td>
<td>invite request</td>
<td>header: request: Cust &lt;&lt;</td>
<td>header: to Modify header: to + Urgent</td>
<td>Use Current Condition</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The figure below illustrates a SIP message manipulation example:

**Figure 21-2: SIP Header Manipulation Example**

- **LAN**
  - 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: 95450459836209155250@10.33.216.1
  - CSeq 1 INVITE
  - Contact: <sip:200@10.33.216.1;transport=tcp>

  Inbound message: Removes IPG1 header

- **Outbound message: Modifies P-Asserted-Identity header**
  - INVITE sip:201@212.3.216.2;user=phone SIP/2.0
  - From: <sip:200@212.3.216.2;user=phone>
  - P-Asserted-Identity: <sip:200@212.3.216.1>
  - Call-ID: 95450459836209155250@10.33.216.1
  - CSeq 1 INVITE
  - Contact: <sip:200@212.3.216.1;transport=tcp>

- **WAN**
21. SIP Message Manipulation

Note:
- For a detailed description of the syntax used for configuring Message Manipulation rules, refer to the "SIP Message Manipulations Quick Reference Guide".
- Inbound message manipulation is done only after the Classification, inbound/outbound number manipulations, and routing processes.
- Each message can be manipulated twice - on the source leg and on the destination leg (i.e., source and destination IP Groups).
- Unknown SIP parts can only be added or removed.
- SIP manipulations do not allow you to remove or add mandatory SIP headers. They can only be modified and only on requests that initiate new dialogs. Mandatory SIP headers include To, From, Via, CSeq, Call-Id, and Max-Forwards.
- The SIP Group Name (IPGroup_SIPGroupName) parameter overrides inbound message manipulation rules that manipulate the host name in Request-URI, To, and/or From SIP headers. If you configure a SIP Group Name for the IP Group (see "Configuring IP Groups" on page 345) and you want to manipulate the host name in these SIP headers, you must apply your manipulation rule (Manipulation Set ID) to the IP Group as an Outbound Message Manipulation Set (IPGroup_OutboundManSet), when the IP Group is the destination of the call. If you apply the Manipulation Set as an Inbound Message Manipulation Set (IPGroup_InboundManSet), when the IP Group is the source of the call, the manipulation rule will be overridden by the SIP Group Name.

The following procedure describes how to configure Message Manipulation rules through the Web interface. You can also configure it through ini file (MessageManipulations) or CLI (configure voip > message message-manipulations).

➢ To configure SIP message manipulation rules:
1. Open the Message Manipulations page (Setup menu > Signaling & Media tab > Message Manipulation folder > Message Manipulations).
2. Click New; the following dialog box appears:

Figure 21-3: Message Manipulations Table - Add Dialog Box
3. Configure a Message Manipulation rule according to the parameters described in the table below.

4. Click **Apply**.

An example of configured message manipulation rules are shown in the figure below:

**Figure 21-4: Example of Configured Message Manipulation Rules**

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>MANIPULATION SET ID</th>
<th>MESSAGE TYPE</th>
<th>CONDITION</th>
<th>ACTION SUBJECT</th>
<th>ACTION TYPE</th>
<th>ACTION VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>ITSP A</td>
<td>0</td>
<td>invite_response 200</td>
<td>header to url User</td>
<td>Add Suffix</td>
<td>.com</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>invite request 200</td>
<td>header from url user</td>
<td>Modify</td>
<td>header presented-id-url User</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>invite request</td>
<td>header from url user=Unknown</td>
<td>Modify</td>
<td>&quot;200&quot;</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>invite request</td>
<td>header from url user</td>
<td>Modify</td>
<td>param slog-url user</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>invite request</td>
<td>header priority</td>
<td>Remove</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **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 21-1: Message Manipulations Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[MessageManipulations_Index]</td>
<td><em>Note:</em> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines an arbitrary name to easily identify the rule.</td>
</tr>
<tr>
<td>manipulation-name</td>
<td>The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td>[MessageManipulations_ManipulationName]</td>
<td></td>
</tr>
<tr>
<td>Manipulation Set ID</td>
<td>Defines a Manipulation Set ID for the rule. You can define the</td>
</tr>
<tr>
<td>manipulation-set-id</td>
<td>same Manipulation Set ID for multiple rules to create a group of rules.</td>
</tr>
<tr>
<td>[MessageManipulations_ManSetID]</td>
<td>The Manipulation Set ID is used to assign the</td>
</tr>
<tr>
<td></td>
<td>manipulation rules to an IP Group (in the IP Groups table) for</td>
</tr>
<tr>
<td></td>
<td>inbound and/or outbound messages.</td>
</tr>
<tr>
<td></td>
<td>The valid value is 0 to 19. The default is 0.</td>
</tr>
<tr>
<td>Row Role</td>
<td>Determines which message manipulation condition (configured by the 'Condition' parameter) to use for the rule.</td>
</tr>
<tr>
<td>row-role</td>
<td></td>
</tr>
<tr>
<td>[MessageManipulations_RowRole]</td>
<td></td>
</tr>
<tr>
<td>Note:</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Use Current Condition = (Default) The condition configured in the table row of the rule is used.</td>
</tr>
<tr>
<td>1</td>
<td>Use Previous Condition = The condition configured in the first table row above the rule that is configured to <strong>Use Current Condition</strong> is used. For example, if Index 3 is configured to <strong>Use Current Condition</strong> and Index 4 and 5 are configured to <strong>Use Previous Condition</strong>, Index 4 and 5 use the condition configured for Index 3. A configuration example is shown in the beginning of this section. The option allows you to use the same condition for multiple manipulation rules.</td>
</tr>
</tbody>
</table>
| When configured to **Use Previous Condition**, the 'Message
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type' and 'Condition' parameters are not applicable and if configured are ignored.</strong></td>
<td></td>
</tr>
<tr>
<td>- When multiple manipulation rules apply to the same header, the next rule applies to the resultant string of the previous rule.</td>
<td></td>
</tr>
<tr>
<td><strong>Match</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Message Type</strong></td>
<td>Defines the SIP message type that you want to manipulate. The valid value is a string (case-insensitive) denoting the SIP message. For example:</td>
</tr>
<tr>
<td>message-type</td>
<td></td>
</tr>
<tr>
<td>[MessageManipulations_MessageType]</td>
<td></td>
</tr>
<tr>
<td>- Empty = rule applies to all messages</td>
<td></td>
</tr>
<tr>
<td>- Invite = rule applies to all INVITE requests and responses</td>
<td></td>
</tr>
<tr>
<td>- Invite.Request = rule applies to INVITE requests</td>
<td></td>
</tr>
<tr>
<td>- Invite.Response = rule applies to INVITE responses</td>
<td></td>
</tr>
<tr>
<td>- subscribe.response.2xx = rule applies to SUBSCRIBE confirmation responses</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> Currently, SIP 100 Trying messages cannot be manipulated.</td>
<td></td>
</tr>
<tr>
<td><strong>Condition</strong></td>
<td>Defines the condition that must exist for the rule to be applied. The valid value is a string (case-insensitive). For example:</td>
</tr>
<tr>
<td>condition</td>
<td></td>
</tr>
<tr>
<td>[MessageManipulations_Condition]</td>
<td></td>
</tr>
<tr>
<td>- header.from.url.user== '100' (indicates that the user part of the From header must have the value &quot;100&quot;)</td>
<td></td>
</tr>
<tr>
<td>- header.contact.param.expires &gt; '3600'</td>
<td></td>
</tr>
<tr>
<td>- header.to.url.host contains 'domain'</td>
<td></td>
</tr>
<tr>
<td>- param.call.dst.user != '100'</td>
<td></td>
</tr>
<tr>
<td><strong>Action</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Action Subject</strong></td>
<td>Defines the SIP header upon which the manipulation is performed. The valid value is a string (case-insensitive).</td>
</tr>
<tr>
<td>action-subject</td>
<td></td>
</tr>
<tr>
<td>[MessageManipulations_ActionSubject]</td>
<td></td>
</tr>
<tr>
<td><strong>Action Type</strong></td>
<td>Defines the type of manipulation.</td>
</tr>
<tr>
<td>action-type</td>
<td></td>
</tr>
<tr>
<td>[MessageManipulations_ActionType]</td>
<td></td>
</tr>
<tr>
<td>- [0] Add (default) = Adds new header/param/body (header or parameter elements).</td>
<td></td>
</tr>
<tr>
<td>- [1] Remove = Removes header/param/body (header or parameter elements).</td>
<td></td>
</tr>
<tr>
<td>- [3] Add Prefix = Adds value at the beginning of the string (string element only).</td>
<td></td>
</tr>
<tr>
<td>- [4] Add Suffix = Adds value at the end of the string (string element only).</td>
<td></td>
</tr>
<tr>
<td>- [5] Remove Suffix = Removes value from the end of the string (string element only).</td>
<td></td>
</tr>
<tr>
<td>- [6] Remove Prefix = Removes value from the beginning of the string (string element only).</td>
<td></td>
</tr>
</tbody>
</table>
21.2 Configuring Message Condition Rules

The Message Conditions table lets you configure up to 1,200 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 493)
- IP-to-IP routing rules in the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 501)
- Outbound Manipulation rules in the Outbound Manipulations table (see "Configuring IP-to-IP Outbound Manipulations" on page 535)

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 393). 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]\|\s8[\|\n])" can be used to enable routing based on payload type 8 in the incoming SDP message.

Note: For a description on SIP message manipulation syntax, refer to the SIP Message Manipulations Quick Reference Guide.

The following procedure describes how to configure Message Condition rules through the Web interface. You can also configure it through ini file (ConditionTable) or CLI (configure voip > sbc routing condition-table).
To configure a Message Condition rule:

1. Open the Message Conditions table (Setup menu > Signaling & Media tab > Message Manipulation folder > Message Conditions).
2. Click New; the following dialog box appears:

   ![Message Conditions Table - Add Dialog Box]

3. Configure a Message Condition rule according to the parameters described in the table below.
4. Click Apply.

An example of configured Message Condition rules is shown in the figure below:

   ![Example of Configured SIP Message Conditions]

- **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 21-2: Message Conditions Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[ConditionTable_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a brief description of the Condition rule.</td>
</tr>
<tr>
<td>[ConditionTable_Name]</td>
<td>The valid value is a string of up to 59 characters.</td>
</tr>
<tr>
<td>Condition</td>
<td>Defines the Condition rule of the SIP message.</td>
</tr>
<tr>
<td>[ConditionTable_Condition]</td>
<td>The valid value is a string.</td>
</tr>
<tr>
<td></td>
<td>Note: User and host parts must be enclosed in single quotes.</td>
</tr>
</tbody>
</table>
21.3 Configuring SIP Message Policy Rules

The Message Policies 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 SIP messages. This feature is helpful against VoIP fuzzing (also known as robustness testing), which sends different types of packets to its "victims" for finding bugs and vulnerabilities. For example, the attacker might try sending a SIP message containing either an oversized parameter or too many occurrences of a parameter.

You can also enable the Message Policy to protect the device against incoming SIP messages with malicious signature patterns, which identify specific scanning tools used by attackers to search for SIP servers in a network. To configure Malicious Signatures, see "Configuring Malicious Signatures" on page 559.

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
- Malicious Signatures

The Message Policies table provides a default Message Policy called "Malicious Signature DB Protection" (Index 0), which is based only on Malicious Signatures and discards SIP messages identified with any of the signature patterns configured in the Malicious Signature table.

To apply a SIP Message Policy rule to calls, you need to assign it to the SIP Interface associated with the relevant IP Group (see "Configuring SIP Interfaces" on page 337).

The following procedure describes how to configure Message Policy rules through the Web interface. You can also configure it through ini file (MessagePolicy) or CLI (configure voip > message message-policy).

➢ To configure SIP Message Policy rules:

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

**Figure 21-7: Message Policies Table - Add Dialog Box**

![Message Policies Table - Add Dialog Box](image)

3. Configure a Message Policy rule according to the parameters described in the table below.

4. Click **Apply**.

**Table 21-3: Message Policies Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[MessagePolicy_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. <strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>name</td>
<td>[MessagePolicy_Name]</td>
</tr>
<tr>
<td><strong>Limits</strong></td>
<td></td>
</tr>
<tr>
<td>Max Message Length</td>
<td>Defines the maximum SIP message length. The valid value is up to 32,768 characters. The default is 32,768.</td>
</tr>
<tr>
<td>max-message-length</td>
<td>[MessagePolicy_MaxMessageLength]</td>
</tr>
<tr>
<td>Max Header Length</td>
<td>Defines the maximum SIP header length. The valid value is up to 512 characters. The default is 512.</td>
</tr>
<tr>
<td>max-header-length</td>
<td>[MessagePolicy_MaxHeaderLength]</td>
</tr>
<tr>
<td>Max Body Length</td>
<td>Defines the maximum SIP message body length. This</td>
</tr>
<tr>
<td>max-body-length</td>
<td></td>
</tr>
<tr>
<td>Max Num Headers</td>
<td></td>
</tr>
<tr>
<td>Max Num Bodies</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>max-body-length</td>
<td>is the value of the Content-Length header. The valid value is up to 1,024 characters. The default is 1,024.</td>
</tr>
<tr>
<td>[MessagePolicy_MaxBodyLength]</td>
<td></td>
</tr>
<tr>
<td>Max Num Headers</td>
<td>Defines the maximum number of SIP headers. The valid value is any number up to 32. The default is 32.</td>
</tr>
<tr>
<td>max-num-headers</td>
<td>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.</td>
</tr>
<tr>
<td>[MessagePolicy_MaxNumHeaders]</td>
<td></td>
</tr>
<tr>
<td>Max Num Bodies</td>
<td>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.</td>
</tr>
<tr>
<td>max-num-bodies</td>
<td></td>
</tr>
<tr>
<td>Policies</td>
<td></td>
</tr>
<tr>
<td>Send Rejection</td>
<td>Defines whether the device sends a SIP response if it rejects a message request due to the Message Policy. The default response code is SIP 400 &quot;Bad Request&quot;. To configure a different response code, use the MessagePolicyRejectResponseType parameter.</td>
</tr>
<tr>
<td>send-rejection</td>
<td></td>
</tr>
<tr>
<td>[MessagePolicy_SendRejection]</td>
<td></td>
</tr>
<tr>
<td>Method List Blacklist-Whitelist Policy</td>
<td></td>
</tr>
<tr>
<td>Method List</td>
<td>Defines SIP methods (e.g., INVITE\BYE) to blacklist or whitelist. Multiple methods are separated by a backslash (). The method values are case-insensitive.</td>
</tr>
<tr>
<td>method-list</td>
<td></td>
</tr>
<tr>
<td>[MessagePolicy_MethodList]</td>
<td></td>
</tr>
<tr>
<td>Method List Type</td>
<td>Defines the policy (blacklist or whitelist) for the SIP methods specified in the 'Method List' parameter (above).</td>
</tr>
<tr>
<td>method-list-type</td>
<td></td>
</tr>
<tr>
<td>[MessagePolicy_MethodListType]</td>
<td></td>
</tr>
<tr>
<td>SIP Body Blacklist-Whitelist Policy</td>
<td></td>
</tr>
<tr>
<td>Body List</td>
<td>Defines the SIP body type (i.e., value of the Content-Type header) to blacklist or whitelist. For example, application/sdp. The values of the parameter are case-sensitive.</td>
</tr>
<tr>
<td>body-list</td>
<td></td>
</tr>
<tr>
<td>[MessagePolicy_BodyList]</td>
<td></td>
</tr>
<tr>
<td>Body List Type</td>
<td>Defines the policy (blacklist or whitelist) for the SIP body specified in the 'Body List' parameter (above).</td>
</tr>
<tr>
<td>body-list-type</td>
<td></td>
</tr>
<tr>
<td>[MessagePolicy_BodyListType]</td>
<td></td>
</tr>
</tbody>
</table>
21. SIP Message Manipulation

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Malicious Signature</td>
<td></td>
</tr>
<tr>
<td>Malicious Signature Database</td>
<td></td>
</tr>
<tr>
<td>signature-db-enable</td>
<td>Enables the use of the Malicious Signature database</td>
</tr>
<tr>
<td></td>
<td>[0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>[1] Enable</td>
</tr>
<tr>
<td></td>
<td>To configure Malicious Signatures, see &quot;Configuring Malicious Signatures&quot;</td>
</tr>
<tr>
<td></td>
<td>on page 559.</td>
</tr>
</tbody>
</table>

21.4 Configuring Pre-Parsing Manipulation Rules

The Pre-Parsing Manipulation Set table lets you configure up to 10 Pre-Parsing Manipulation Sets. Pre-Parsing Manipulation allows you to manipulate incoming SIP messages (dialog-initiating and in-dialog) before they are parsed (as an object) by the device. In other words, messages can be manipulated in their original format (plain text) as received from the network. Pre-Parsing Manipulation may be useful, for example, to overcome parser strictness or to “allow” possible parsing errors.

To use a configured Pre-Parsing Manipulation Set, you need to assign it to a SIP Interface (see Configuring SIP Interfaces on page 337). The device performs Pre-Parsing Manipulation before Pre-Classification Manipulation and Classification.

Pre-Parsing Manipulation rules are defined by the SIP message element to manipulate (for example, INVITE), the pattern based on regular expression (regex) to search for (match) in the incoming message, and the regex pattern to replace the matched pattern.

**Note:** For a detailed description of supported regex syntax, refer to the Message Manipulation Reference Guide.

Pre-Parsing Manipulation is configured using two tables with parent-child type relationship:

- Parent table: Pre-Parsing Manipulation Sets table, which defines a descriptive name for the Pre-Parsing Manipulation Set.
- Child table: Pre-Parsing Manipulation Rules table, which defines the actual manipulation rule. You can configure up to 10 rules per Pre-Parsing Manipulation Set.

The following procedure describes how to configure Pre-Parsing Manipulation Sets through the Web interface. You can also configure it through other management platforms:

- **Pre-Parsing Manipulation Sets table:** ini file (PreParsingManipulationSets) or CLI (configure voip > message pre-parsing-manip-sets)
- **Pre-Parsing Manipulation Rules table:** ini file (PreParsingManipulationRules) or CLI (configure voip > message pre-parsing-manip-rules)

**To configure Pre-Parsing Manipulation Sets:**

1. Open the Pre-Parsing Manipulation Sets table (Setup menu > Signaling & Media tab > Message Manipulation folder > Pre-Parsing Manipulation Sets).
2. Click **New**; the following dialog box appears:

   **Figure 21-8: Pre-Parsing Manipulation Sets Table - Add Dialog Box**

   ![Pre-Parsing Manipulation Sets Table - Add Dialog Box](image)

3. Configure a Pre-Parsing Manipulation Set name according to the parameters described in the table below.

4. Click **Apply**.

   **Pre-Parsing Manipulation Set Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
   | Index [PreParsingManipulation Sets_Index] | Defines an index number for the new table row.  
   **Note:** Each row must be configured with a unique index. |
   | Name [PreParsingManipulation Sets_Name] | Defines a descriptive name, which is used when associating the row in other tables.  
The valid value is a string of up to 40 characters.  
**Note:** Each row must be configured with a unique name. |

5. In the Pre-Parsing Manipulation Sets table, select the row, and then click the **Pre-Parsing Manipulation Rules** link located below the table; the Pre-Parsing Manipulation Rules table appears.

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

   **Figure 21-9: Pre-Parsing Manipulation Rules Table - Add Dialog Box**

   ![Pre-Parsing Manipulation Rules Table - Add Dialog Box](image)

7. Configure a rule according to the parameters described in the table below.

8. Click **New**, and then save your settings to flash memory.
## Pre-Parsing Manipulation Rules Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Match</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row. <strong>Note</strong>: Each row must be configured with a unique index.</td>
</tr>
</tbody>
</table>
| Message Type | Defines the SIP message type to which you want to apply the rule. The following syntax is supported:  
  - To apply the rule to any message type, leave the field empty or configure it to **any**.  
  - SIP requests:  
    - **any.request**: The rule is applied to any request.  
    - `<SIP Method>.request`: The rule is applied to the specified SIP Method (e.g., invite.request).  
  - SIP responses:  
    - **any.response**: The rule is applied to any response.  
    - **response.<response code>**: The rule is applied to messages with the specified response (e.g., response.200 for SIP 200 or response.1xx for any provisional response). |

| **Action** |             |
| Pattern | Defines a pattern, based on regex, to search for (match) in the incoming message. For more information on regex, refer to the *Message Manipulation Reference Guide*. |
| Replace-With | Defines a pattern, based on regex, to replace the matched pattern (defined above). For more information on regex, refer to the *Message Manipulation Reference Guide*. |
22 Coders and Profiles

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

22.1 Configuring Coder Groups

The Coder Groups table lets you configure up to 21 Coder Groups. The Coder Group determines the audio (voice) coders used for calls. Each Coder Group can include up to 10 coders, where the packetization time (ptime), bit rate, payload type, and silence suppression can be configured per coder. 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.

The Coder Groups table provides a pre-defined Coder Group (index 0) that is configured with the G.711 A-law coder. If no other Coder Groups are configured, the default Coder Group (which you can modify) is used for all calls. Alternatively, if you want to use specific coders or coder settings (e.g., packetization time) for different calls (entities), you need to configure a Coder Group for each entity and then assign each Coder Group to the IP Profile (see "Configuring IP Profiles" on page 417) associated with the entity (IP Group). If an IP Group is not associated with a Coder Group, the default Coder Group is used.

You can also use Coder Groups for audio coder transcoding of SBC calls. If two SIP entities need to communicate, but one does not support a coder required by the other, the device can add the required coder to the SDP offer. The added coder is referred to as an extension coder. For more information on extension coders, see Coder Transcoding on page 467.

To apply a Coder Group for transcoding to a SIP entity:

1. Configure a Coder Group in the Coder Groups table (see description below).

2. In the IP Profile associated with the SIP entity (see Configuring IP Profiles on page 417):
   a. Assign the Coder Group (using the IpProfile_SBCExtensionCodersGroupName parameter).
   b. Enable the use of the Coder Group for transcoding (by configuring the IpProfile_SBCAllowedCodersMode parameter to Restriction or Restriction and Preference).
Note:
- For supported audio coders, see "Supported Audio Coders" on page 410.
- Some coders are license-dependent and are available only if purchased from AudioCodes and included in the License Key installed on your device. For more information, contact your AudioCodes sales representative.
- Only the packetization time of the first coder listed in the Coder Group is declared in INVITE/200 OK SDP even if multiple coders are configured. 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 some fields is hard-coded according to common standards (e.g., payload type of G.711 U-law is always 0).
- The G.722 coder provides Packet Loss Concealment (PLC) capabilities, ensuring higher voice quality.
- Opus coder:
  √ If one leg uses a narrowband coder (e.g., G.711) and the other leg uses the Opus coder, the device maintains the narrowband coder flavor by using the narrowband Opus coder. Alternatively, if one leg uses a wideband coder (e.g., G.722) and the other leg uses the Opus coder, the device maintains the wideband coder flavor by using the wideband Opus coder.
- For more information on V.152 and implementation of T.38 and VBD coders, see "Supporting V.152 Implementation" on page 193.

The following procedure describes how to configure the Coder Groups table through the Web interface. You can also configure it through ini file (AudioCodersGroups and AudioCoders) or CLI (configure voip > coders-and-profiles audio-coders-groups).

➢ To configure a Coder Group:

1. Open the Coder Groups table (Setup menu > Signaling & Media tab > Coders & Profiles folder > Coder Groups).

![Figure 22-1: Coder Group Table](image)

2. From the 'Coder Group Name' drop-down list, select the desired Coder Group index number and name.
3. Configure the Coder Group according to the parameters described in the table below.
4. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.
You can delete a Coder Group, as described in the following procedure.

To delete a Coder Group:

1. From the 'Coder Group Name' drop-down list, select the Coder Group that you want to delete.
2. Click Delete Group.

### Table 22-1: Coder Groups Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coder Group Name</td>
<td>Defines the name and index for the Coder Group.</td>
</tr>
<tr>
<td>[AudioCodersGroups_Index]</td>
<td></td>
</tr>
<tr>
<td>[AudioCodersGroups_Name]</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The Coder Group index/name cannot be configured.</td>
</tr>
<tr>
<td>[AudioCoders_AudioCodersIndex]</td>
<td>Index row of the coder per Coder Group</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The parameter is applicable only to the ini file.</td>
</tr>
<tr>
<td>Coder Name name</td>
<td>Defines the coder type. For coder names, see &quot;Supported Audio Coders&quot; on page 410.</td>
</tr>
<tr>
<td>[AudioCoders_Name]</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Each coder type (e.g., G.729) can be configured only once in the table.</td>
</tr>
<tr>
<td>Packetization Time p-time</td>
<td>Defines the packetization time (in msec) for the coder. The packetization time determines how many coder payloads are combined into a single RTP packet. For ptime, see &quot;Supported Audio Coders&quot; on page 410.</td>
</tr>
<tr>
<td>[AudioCoders_pTime]</td>
<td></td>
</tr>
<tr>
<td>Rate rate</td>
<td>Defines the bit rate (in kbps) for the coder. For rates, see &quot;Supported Audio Coders&quot; on page 410.</td>
</tr>
<tr>
<td>[AudioCoders_rate]</td>
<td></td>
</tr>
<tr>
<td>Payload Type payload-type</td>
<td>Defines the payload type if the payload type (i.e., format of the RTP payload) for the coder is dynamic. For payload types, see &quot;Supported Audio Coders&quot; on page 410.</td>
</tr>
<tr>
<td>[AudioCoders_PayloadType]</td>
<td></td>
</tr>
<tr>
<td>Silence Suppression silence-suppression</td>
<td>Enables silence suppression for the coder.</td>
</tr>
<tr>
<td>[AudioCoders_Sce]</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>Option [2] Enable w/o Adaptation is applicable only to G.729.</td>
</tr>
<tr>
<td></td>
<td>If you disable silence suppression for G.729, the device includes 'annexb=no' in the SDP of the relevant SIP messages. If you enable silence suppression, 'annexb=yes' is included. An exception is when the remote gateway is Cisco equipment (IsCiscoSCEMode).</td>
</tr>
<tr>
<td>Coder Specific coder-specific</td>
<td>Defines additional settings specific to the coder. Currently, the parameter is applicable only to the AMR coder and is used to configure the payload format type.</td>
</tr>
<tr>
<td>[AudioCoders_CoderSpecific]</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The AMR payload type can be configured globally using the AmrOctetAlignedEnable parameter. However, the Coder</td>
</tr>
</tbody>
</table>
### 22.1.1 Supported Audio Coders

The table below lists the coders supported by the device.

**Table 22-2: Supported Audio Coders**

<table>
<thead>
<tr>
<th>Coder Name</th>
<th>Packetization Time (msec)</th>
<th>Rate (kbps)</th>
<th>Payload Type</th>
<th>Silence Suppression</th>
</tr>
</thead>
<tbody>
<tr>
<td>G.722 g722 [20]</td>
<td>20 (default), 40, 60, 80, 100, 120</td>
<td>[90] 64 (default)</td>
<td>9</td>
<td>N/A</td>
</tr>
<tr>
<td>AMR</td>
<td>20 (default)</td>
<td>▪ [4] 4.75</td>
<td>Dynamic</td>
<td>▪ [0] Disable</td>
</tr>
</tbody>
</table>
### 22. Coders and Profiles

<table>
<thead>
<tr>
<th>Coder Name</th>
<th>Packetization Time (msec)</th>
<th>Rate (kbps)</th>
<th>Payload Type</th>
<th>Silence Suppression</th>
</tr>
</thead>
<tbody>
<tr>
<td>[14]</td>
<td>[9] 5.90</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[14] 6.70</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[16] 7.40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[18] 7.95</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[27] 10.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[30] 12.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMR-WB</td>
<td>20 (default)</td>
<td></td>
<td>Dynamic</td>
<td>[0] Disable</td>
</tr>
<tr>
<td>[15]</td>
<td>[21] 8.85</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[32] 12.65</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[37] 14.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[41] 15.85</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[48] 18.25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[49] 19.85</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[53] 23.05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[55] 23.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(default)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SILK-NB</td>
<td>20 (default), 40, 60, 80, 100</td>
<td>[19] 8</td>
<td>Dynamic (default 76)</td>
<td>N/A</td>
</tr>
<tr>
<td>silk-nb</td>
<td>[35]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SILK-WB</td>
<td>20 (default), 40, 60, 80, and 100</td>
<td>[43] 16</td>
<td>Dynamic (default 77)</td>
<td>N/A</td>
</tr>
<tr>
<td>silk-wb</td>
<td>[36]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T.38</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>t-38</td>
<td>[4]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Opus</td>
<td>20 (default), 40, 60, 80, 120</td>
<td>N/A</td>
<td>Dynamic (default 111)</td>
<td>N/A</td>
</tr>
<tr>
<td>opus</td>
<td>[40]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### 22.1.2 Configuring Various Codec Attributes

The following procedure describes how to configure various coder attributes such as bitrate.

- **To configure codec attributes:**
  1. Open the Coder Settings page (Setup menu > Signaling & Media tab > Coders & Profiles folder > Coder Settings).
  2. Configure the following parameters:
     - AMR coder:
- 'AMR Payload Format' (AmrOctetAlignedEnable): Defines the AMR payload format type:

<table>
<thead>
<tr>
<th>AMR CODER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AMR Payload Format</strong></td>
</tr>
</tbody>
</table>

- 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.

  **Figure 22-2: Configuring SILK Coder Attributes**

<table>
<thead>
<tr>
<th>SILK CODER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Silk Tx Inband FEC</strong></td>
</tr>
<tr>
<td><strong>Silk Max Average Bit Rate</strong></td>
</tr>
</tbody>
</table>

- Opus coder:
  - 'Opus Max Average Bitrate' (OpusMaxAverageBitRate): Defines the maximum average bit rate (in bps) for the Opus coder.

  **Figure 22-3: Configuring Opus Coder Attributes**

<table>
<thead>
<tr>
<th>OPUS CODER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Opus Max Average Bitrate [bps]</strong></td>
</tr>
</tbody>
</table>

3. Click Apply.

### 22.2 Configuring Allowed Audio Coder Groups

The Allowed Audio Coders Groups table lets you configure up to 20 Allowed Audio Coders Groups. For each Allowed Audio Coders Group, you can configure up to 10 audio coders. The coders can include pre-defined coders and user-defined (string) coders for non-standard or unknown coders.

Allowed Audio Coders Groups restrict coders for SIP entities. Only coders listed in the Allowed Audio Coders Group (i.e., allowed coders) that is associated with the SIP entity can be used. If the coders in the SDP offer ('a=rtpmap' field) of the incoming SIP message are not listed in the Allowed Audio Coders Group, the device rejects the calls, unless transcoding is configured, whereby "extension" coders are added to the SDP, as described in Coder Transcoding on page 467. If the SDP offer contains some coders that are listed in the Allowed Audio Coders Group, the device manipulates the SDP offer by removing the coders that are not listed in the Allowed Audio Coders Group, before routing the SIP message to its destination. Thus, only coders that are common between the coders in the SDP offer and the coders in the Allowed Audio Coders Group are used. For more information on coder restriction, see "Restricting Audio Coders" on page 466.

For example, assume the following:

- The SDP offer in the incoming SIP message contains the G.729, G.711, and G.723 coders.
- The allowed coders configured for the SIP entity include G.711 and G.729.
The device removes the G.723 coder from the SDP offer, re-orders the coder list so that G.711 is listed first, and sends the SIP message containing only the G.711 and G.729 coders in the SDP.

To apply an Allowed Audio Coders Group for restricting coders to a SIP entity:

1. Configure an Allowed Audio Coders Group in the Allowed Audio Coders Groups table (see description below).
2. In the IP Profile associated with the SIP entity (see "Configuring IP Profiles" on page 417):
   - Assign the Allowed Audio Coders Group (using the IpProfile_SBCAllowedAudioCodersGroupName parameter).
   - Enable the use of Allowed Audio Coders Groups (by configuring the IpProfile_SBCAllowedCodersMode parameter to **Restriction** or **Restriction and Preference**).

The device also re-orders (prioritizes) the coder list in the SDP according to the order of appearance of the coders listed in the Allowed Audio Coders Group. The first listed coder has the highest priority and the last coder has the lowest priority. For more information, see "Prioritizing Coder List in SDP Offer" on page 470.

### Note:

- The Allowed Audio Coders Group for coder restriction takes precedence over the Coder Group for extension coders. In other words, if an extension coder is not listed as an allowed coder, the device does not add the extension coder to the SDP offer.
- To configure "extension" coders for adding to the SDP offer for audio transcoding, use the Coder Groups table (see Configuring Coder Groups on page 407).

The following procedure describes how to configure Allowed Audio Coders Groups through the Web interface. You can also configure it through ini file (AllowedAudioCodersGroups and AllowedAudioCoders) or CLI (configure voip > coders-and-profiles allowed-audio-coders-groups; configure voip > coders-and-profiles allowed-audio-coders <group index/coder index>).

#### To configure an Allowed Audio Coders Group:

1. Open the Allowed Audio Coders Groups table (Setup menu > Signaling & Media tab > Coders & Profiles folder > Allowed Audio Coders Groups).
2. Click **New**; the following dialog box appears:

   ![Figure 22-4: Allowed Audio Coders Groups table - Add Dialog Box](image)

   - **Index**
   - **Name**

3. Configure a name for the Allowed Audio Coders Group according to the parameters described in the table below.
4. Click **Apply**.
5. Select the new row that you configured, and then click the **Allowed Audio Coders** link located below the table; the Allowed Audio Coders table opens.

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

   **Figure 22-5: Allowed Audio Coders Table - Add Dialog Box**

7. Configure coders for the Allowed Audio Coders Group according to the parameters described in the table below.

8. Click **Apply**.

   **Table 22-3: Allowed Audio Coders Groups and Allowed Audio Coders Tables Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Allowed Audio Coders Groups Table</strong></td>
<td></td>
</tr>
<tr>
<td>Index allowed-audio-coders-groups &lt;index&gt;</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[AllowedAudioCodersGroups_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name coders-group-name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td>[AllowedAudioCodersGroups_Name]</td>
<td><strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td><strong>Allowed Audio Coders Table</strong></td>
<td></td>
</tr>
<tr>
<td>Index allowed-audio-coders &lt;group index/coder index&gt;</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[AllowedAudioCoders_AllowedAudioCodersIndex]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Coder coder</td>
<td>Selects a coder from the list of coders.</td>
</tr>
<tr>
<td>[AllowedAudioCoders_CoderID]</td>
<td><strong>Note:</strong> Each coder can be configured only once per Allowed Audio Coders Group.</td>
</tr>
<tr>
<td>User-defined Coder user-define-coder</td>
<td>Defines a user-defined coder. The valid value is a string of up to 24 characters (case-insensitive). For example, &quot;HD.123&quot; (without quotes).</td>
</tr>
<tr>
<td>[AllowedAudioCoders_UserDefineCoder]</td>
<td><strong>Note:</strong> Each coder can be configured only once per Allowed Audio Coders Group.</td>
</tr>
</tbody>
</table>
22.3 Configuring Allowed Video Coder Groups

The Allowed Video Coders Groups table lets you configure up to four Allowed Video Coders Groups. Each Allowed Video Coders Group can be configured with up to 10 coders. An Allowed Video Coders Group defines a list of video coders that can be used when forwarding video streams to a specific SIP entity. The coders can include pre-defined video coders and user-defined (string) video coders for non-standard or unknown coders.

Allowed Video Coders Groups are assigned to SIP entities, using IP Profiles (see "Configuring IP Profiles" on page 417). The video coders appear in the SDP media type "video" ('m=video' line). Coders that are not listed in the Allowed Video Coders Group are removed from the SDP offer that is sent to the SIP entity. Only coders that are common between the coders in the SDP offer and the coders listed in the Allowed Video Coders Group are used. Thus, Allowed Video Coders Groups enable you to enforce the use of only specified coders. For more information, see "Restricting Audio Coders" on page 466.

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 470.

The following procedure describes how to configure Allowed Video Coders Groups through the Web interface. You can also configure it through ini file (AllowedVideoCodersGroups and AllowedVideoCoders) or CLI (configure voip > coders-and-profiles allowed-video-coders-groups; configure voip > coders-and-profiles allowed-video-coders <group index/coder index>).

- To configure an Allowed Video Coders Group:
  1. Open the Allowed Video Coders Groups table (Setup menu > Signaling & Media tab > Coders & Profiles folder > Allowed Video Coders Groups).
  2. Click New; the following dialog box appears:

  Figure 22-6: Allowed Video Coders Groups Table - Add Dialog Box

  ![Allowed Video Coders Groups Table - Add Dialog Box](image)

  3. Configure a name for the Allowed Video Coders Group according to the parameters described in the table below.

  4. Click Apply.

  5. Select the new row that you configured, and then click the Allowed Video Coders link located below the table; the Allowed Video Coders table opens.
6. Click **New**; the following dialog box appears:

   **Figure 22-7: Allowed Video Coders Table - Add Dialog Box**

   ![Allowed Video Coders Table - Add Dialog Box](image)

7. Configure coders for the Allowed Video Coders Group according to the parameters described in the table below.

8. Click **Apply**.

   **Table 22-4: Allowed Video Coders Groups and Allowed Video Coders Tables Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Allowed Video Coders Groups Table</strong></td>
<td></td>
</tr>
<tr>
<td>Index allowed-video-coders-groups &lt;index&gt; [AllowedVideoCodersGroups_Index]</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name coders-group-name [AllowedVideoCodersGroups_Name]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. <strong>Note</strong>: Each row must be configured with a unique name.</td>
</tr>
<tr>
<td><strong>Allowed Video Coders Table</strong></td>
<td></td>
</tr>
<tr>
<td>Index allowed-video-coders &lt;group index/coder index&gt; [AllowedVideoCoders_AllowedVideoCodersIndex]</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>User Define Coder user-define-coder [AllowedVideoCoders_UserDefineCoder]</td>
<td>Defines a user-defined coder. The valid value is a string of up to 24 characters (case-insensitive). For example, &quot;HD.123&quot; (without quotes). <strong>Note</strong>: Each coder can be configured only once per Allowed Video Coders Group.</td>
</tr>
</tbody>
</table>
22.4 Configuring IP Profiles

The IP Profiles table lets you configure up to 300 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 345

Many of the parameters in the IP Profiles table have a corresponding "global" parameter. For calls that are not associated with any IP Profile, the settings of the "global" parameters are applied.

Note: IP Profiles can also be implemented when using a Proxy server (when the AlwaysUseRouteTable parameter is set to 1).

The following procedure describes how to configure IP Profiles through the Web interface. You can also configure it through ini file (IPProfile) or CLI (configure voip > coders-and-profiles ip-profile).

➢ **To configure an IP Profile:**

1. Open the IP Profiles table (Setup menu > Signaling & Media tab > Coders & Profiles folder > IP Profiles).
2. Click New; the following dialog box appears:

   ![Figure 22-8: IP Profiles Table - Add Dialog Box]

3. Configure an IP Profile according to the parameters described in the table below.
4. Click Apply.
### Table 22-5: IP Profiles Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[IpProfile_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td>profile-name</td>
<td>The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td>[IpProfile_ProfileName]</td>
<td></td>
</tr>
<tr>
<td><strong>Media Security</strong></td>
<td></td>
</tr>
<tr>
<td>SBC Media Security Mode</td>
<td>Defines the handling of RTP and SRTP for the SIP entity associated with the IP Profile.</td>
</tr>
<tr>
<td>sbc-media-security-behaviour</td>
<td>- [0] As is = (Default) No special handling for RTP\SRTP is done.</td>
</tr>
<tr>
<td>[IpProfile_SBCMediaSecurityBehaviour]</td>
<td>- [1] SRTP = SBC legs negotiate only SRTP media lines, and RTP media lines are removed from the incoming SDP offer-answer.</td>
</tr>
<tr>
<td></td>
<td>- [2] RTP = SBC legs negotiate only RTP media lines, and SRTP media lines are removed from the incoming offer-answer.</td>
</tr>
<tr>
<td></td>
<td>- [3] Both = Each offer-answer is extended (if not already) to two media lines - one RTP and the other SRTP.</td>
</tr>
<tr>
<td>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:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- At least one supported SDP &quot;crypto&quot; attribute and parameters.</td>
</tr>
<tr>
<td></td>
<td>- EnableMediaSecurity must be set to 1.</td>
</tr>
<tr>
<td>If one of the above transcoding prerequisites is not met, then:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- any value other than “As is” is discarded.</td>
</tr>
<tr>
<td></td>
<td>- if the incoming offer is SRTP, force transcoding, coder transcoding, and DTMF extensions are not applied.</td>
</tr>
<tr>
<td>Symmetric MKI</td>
<td>Enables symmetric MKI negotiation.</td>
</tr>
<tr>
<td>enable-symmetric-mki</td>
<td>- [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).</td>
</tr>
<tr>
<td>[IpProfile_EnableSymmetricMKI]</td>
<td>- [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:</td>
</tr>
<tr>
<td></td>
<td>a=crypto:2 AES_CM_128_HMAC_SHA1_80 inline:TAaxNnQt8/qLQMnDuG4vxYfwL6k7eBK/ufk04pR4</td>
</tr>
<tr>
<td></td>
<td>a=crypto:3 AES_CM_128_HMAC_SHA1_80 inline:bnuY2nMxSfUigtviWJzmrz7OF3AiRO015V</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>nh0kH\2^31</td>
<td>The first crypto line includes the MKI parameter &quot;1:1&quot;. 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/qwBjkekl4kJy13wCtYeZLq1/ QFuxw</td>
</tr>
<tr>
<td></td>
<td>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). <strong>Note:</strong> The corresponding global parameter is EnableSymmetricMKI.</td>
</tr>
<tr>
<td>MKI Size</td>
<td>mki-size</td>
</tr>
</tbody>
</table>
| [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). **Note:**  
  - 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.                                                                                                                                                                                                                                                                                                       |
| SBC Enforce MKI Size            | sbc-enforce-mki-size                                                                                                                                                                                                                                                                                                                                                                                                              |
| [IpProfile_SBCEnforceMKISize]   | Enables negotiation of the Master Key Identifier (MKI) length for SRTP-to-SRTP flows between SIP networks (i.e., IP Groups). This includes the capability of modifying the MKI length on the inbound or outbound SBC call leg for the SIP entity associated with the IP Profile.  
  - [0] Don't enforce = (Default) Device forwards the MKI size as is.  
  - [1] Enforce = Device changes the MKI length according to the settings of the IP Profile parameter, MKISize.                                                                                                                                                                                                                                                                                                      |
| SBC Media Security Method       | sbc-media-security-method                                                                                                                                                                                                                                                                                                                                          |
| [IpProfile_SBCMediaSecurityMethod] | Defines the media security protocol for SRTP, for the SIP entity associated with the IP Profile.  
  - [0] SDES = (Default) The device secures RTP using the Session Description Protocol Security Descriptions (SDES) protocol to negotiate the cryptographic keys (RFC 4568). The keys are sent in the SDP body ('a=crypto') of the SIP message and are typically secured using SIP over TLS (SIPS). The encryption of the keys is in plain text in the SDP. SDES implements TLS over TCP.  
  - [1] DTLS = The device uses Datagram Transport Layer Security (DTLS) protocol to secure UDP-based media streams (RFCs 5763 and 5764). For more information  
  - [2] DTLSv2 = The device uses Datagram Transport Layer Security version 2 (DTLSv2) protocol to secure UDP-based media streams (RFCs 8443 and 8444). For more information  
  - [3] DTLSv3 = The device uses Datagram Transport Layer Security version 3 (DTLSv3) protocol to secure UDP-based media streams (RFCs 9518 and 9519). For more information |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>on DTLS, see SRTP using DTLS Protocol on page 207.</td>
<td>- [2] Both = SDES and DTLS protocols are supported.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>To support DTLS, you must also configure the following for the SIP entity:</td>
</tr>
<tr>
<td></td>
<td>✓ TLS Context for DTLS (see Configuring TLS Certificate Contexts on page 105). The server cipher (&quot;Cipher Server&quot;) must be configured to All.</td>
</tr>
<tr>
<td></td>
<td>✓ IpProfile_SBCMediaSecurityBehaviourMedia configured to SRTP or Both.</td>
</tr>
<tr>
<td></td>
<td>✓ IpProfile_SBCRTCPMux configured to Supported. The setting is required as the DTLS handshake is done for the port used for RTP. Therefore, RTCP and RTP should be multiplexed over the same port.</td>
</tr>
<tr>
<td></td>
<td>The device does not support forwarding of DTLS transparently between endpoints (SIP entities).</td>
</tr>
<tr>
<td></td>
<td>As DTLS has been defined by the WebRTC standard as mandatory for encrypting media channels for SRTP key exchange, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 566.</td>
</tr>
<tr>
<td><strong>Reset SRTP Upon Re-key</strong></td>
<td>Enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. This feature ensures that the roll-over counter (ROC), one of the parameters used in the SRTP encryption/decryption process of the SRTP packets is synchronized on both sides for transmit and receive packets.</td>
</tr>
<tr>
<td>reset-srtp-upon-re-key</td>
<td>- [0] Disable = (Default) ROC is not reset on the device side.</td>
</tr>
<tr>
<td></td>
<td>- [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.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>If this feature is disabled and the server resets the ROC upon a re-key generation, one-way voice may occur.</td>
</tr>
<tr>
<td></td>
<td>The corresponding global parameter is ResetSRTPStateUponRekey.</td>
</tr>
<tr>
<td><strong>Generate SRTP Keys Mode</strong></td>
<td>Enables the device to generate a new SRTP key upon receipt of a re-INVITE with the SIP entity associated with the IP Profile.</td>
</tr>
<tr>
<td>generate-srtp-keys</td>
<td>- [0] Only If Required= (Default) The device generates an SRTP key only if necessary.</td>
</tr>
<tr>
<td></td>
<td>- [1] Always = The device always generates a new SRTP key.</td>
</tr>
<tr>
<td><strong>SBC Remove Crypto Lifetime in SDP</strong></td>
<td>Defines the handling of the lifetime field in the 'a=crypto' attribute of the SDP for the SIP entity associated with the IP Profile. The SDP field defines the lifetime of the master key as measured in maximum number of SRTP or SRTCP packets using the master key.</td>
</tr>
<tr>
<td>sbc-sdp-remove-crypto-lifetime</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td><strong>Parameter</strong></td>
<td><strong>Description</strong></td>
</tr>
</tbody>
</table>
| | - [0] No = (Default) The device retains the lifetime field (if present) in the SDP.  
| | - [1] Yes = The device removes the lifetime field from the 'a=crypto' attribute.  
| **Note:** If you configure the parameter to Yes, the following IP Profile parameters must be configured as follows:  
| | - IpProfile_EnableSymmetricMKI configured to Enable [1].  
| | - IpProfile_MKISize configured to 0.  
| | - IpProfile_SBCEnforceMKISize configured to Enforce [1]. |

### SBC Early Media

#### Remote Early Media  
**sbc-rmt-early-media-supp**  
[IpProfile_SBCRemoteEarlyMediaSupport]  
Defines whether the remote side can accept early media or not.  
- [0] Not Supported = Early media is not supported.  
- [1] Supported = (Default) Early media is supported.  

#### Remote Multiple 18x  
**sbc-rmt-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 the SIP entity associated with the IP Profile.  
- [0] Not Supported = Only the first 18x response is forwarded to the caller.  
- [1] Supported = (Default) Multiple 18x responses are forwarded to the caller.  

#### Remote Early Media Response Type  
**sbc-rmt-early-media-resp**  
[IpProfile_SBCRemoteEarlyMediaResponseType]  
Defines the SIP provisional response type - 180 or 183 - for forwarding early media to the caller, for the SIP entity associated with the IP Profile.  
- [0] Transparent = (Default) All early media response types are supported; the device forwards all responses as is (unchanged).  
- [1] 180 = Early media is sent as 180 response only.  
- [2] 183 = Early media is sent as 183 response only.  

#### Remote Multiple Early Dialogs  
**sbc-multi-early-diag**  
[IpProfile_SBCRemoteMultipleEarlyDialogs]  
Defines the device's handling of To-header tags in call forking responses (i.e., multiple SDP answers) sent to the SIP entity associated with the IP Profile. When the SIP entity initiates an INVITE that is subsequently forked (for example, by a proxy server) to multiple endpoints, the endpoints respond with a SIP 183 containing an SDP answer. Typically, each endpoint's response has a different To-header tag. For example, a call initiated by the SIP entity (100@A) is forked and two endpoints respond with ringing, each with a different tag:  
- **Endpoint "tag 2":**  
  SIP/2.0 180 Ringing  
  From: <sip:100@A>;tag=tag1  
  To: sip:200@B;tag=tag2  
  Call-ID: c2  
- **Endpoint "tag 3":**  
  SIP/2.0 180 Ringing  
  From: <sip:100@A>;tag=tag1  
  To: sip:200@B;tag=tag3
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call-ID: c2</td>
<td>In non-standard behavior (when the parameter is configured to Disable), the device forwards all the SDP answers with the same tag. In the example, endpoint &quot;tag 3&quot; is sent with the same tag as endpoint &quot;tag 2&quot; (i.e., To: sip:200@B;tag=tag2).</td>
</tr>
</tbody>
</table>
| -1 | According to Operation Mode = (Default) Depends on the setting of the 'Operation Mode' parameter in the IP Group or SRDs table:  
B2BUA: Device operates as if the parameter is set to Disable [0].  
Call State-full Proxy: Device operates as if the parameter is set to Enable [1]. In addition, the device preserves the From tags and Call-IDs of the endpoints in the SDP answer sent to the SIP entity. |
| 0 | Disable = Device sends the multiple SDP answers with the same To-header tag, to the SIP entity. In other words, this option is relevant if the SIP entity does not support multiple dialogs (and multiple tags). However, non-standard, multiple answer support may still be configured by the SBCRemoteMultipleAnswersMode parameter. |
| 1 | Enable = Device sends the multiple SDP answers with different To-header tags, to the SIP entity. In other words, the SIP entity supports standard multiple SDP answers (with different To-header tags). In this case, the SBCRemoteMultipleAnswersMode parameter is ignored. |
| Note: | If the parameter and the SBCRemoteMultipleAnswersMode parameter are disabled, multiple SDP answers are not reflected to the SIP entity (i.e., the device sends the same SDP answer in multiple 18x and 200 responses). |

Remote Multiple Answers Mode
sbc-multi-answers
[IpProfile_SBCRemoteMultipleAnswers Mode]
Enables interworking multiple SDP answers within the same SIP dialog (non-standard). The parameter enables the device to forward multiple answers to the SIP entity associated with the IP Profile. The parameter is applicable only when the IpProfile_SBCRemoteMultipleEarlyDialogs parameter is disabled.

- 0 | Disable = (Default) Device always sends the same SDP answer, which is based on the first received answer that it sent to the SIP entity, for all forked responses (even if 'Forking Handling Mode' is Sequential), and thus, may result in transcoding. |
| 1 | Enable = If the 'Forking Handling Mode' parameter is configured to Sequential, the device sends multiple SDP answers. |

Remote Early Media RTP Detection Mode
sbc-rmt-early-media-rtp
[IpProfile_SBCRemoteEarlyMediaRTP]
Defines whether the destination UA sends RTP immediately after it sends a 18x response.

- 0 | By Signaling = (Default) Remote client sends RTP immediately after it sends 18x response with early media. The device forwards 18x and RTP as is. |
| 1 | By Media = After sending 18x response, the remote
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Remote RFC 3960 Support**<br>sbc-rmt-rfc3960-supp<br>[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 479.  
- [1] Supported = UA is capable of receiving 18x messages with delayed RTP. |
| **Remote Can Play Ringback**<br>sbc-rmt-can-play-ringback<br>[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 479. |
| **Generate RTP**<br>sbc-generate-rtp<br>[IPProfile_SBCGenerateRTP] | Enables the device to generate "silence" RTP packets to the SIP entity until it detects audio RTP packets from the SIP entity. The parameter provides support for interworking with SIP entities that wait for the first incoming packets before sending RTP (e.g., early media used for ringback tone or IVR) during media negotiation.  
- [0] None (Default) = Silence packets are not generated.  
- [1] Until RTP Detected = The device generates silence RTP packets to the SIP entity upon receipt of a SIP response (183 with SDP) from the SIP entity. In other words, these packets serve as the first incoming packets for the SIP entity. The device stops sending silence packets when it receives RTP packets from the peer side (which it then forwards to the SIP entity).  
**Note:** To generate silence packets, DSP resources are required (except for calls using the G.711 coder). |
| **SBC Media**<br>**Transcoding Mode**<br>transcoding-mode<br>[IpProfile_TranscodingMode] | Defines the transcoding mode (media negotiation) for the SIP entity associated with the IP Profile.  
- [0] Only if Required = (Default) Transcoding is done only when necessary. Many of the media settings (such as gain control) are not implemented on the voice stream. The device forwards RTP packets transparently (RTP-to-RTP), without processing them.  
- [1] Force = Transcoding is always done on the outgoing leg. The device interworks the media for the SIP entity (as both legs have different media capabilities), by implementing DSP transcoding. This enables the device to receive capabilities that are not negotiated between the SIP entities. For example, it can enforce gain control to use voice transcoding even though both legs have negotiated without the device's intervention (such as... |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension Coders Group</td>
<td>Assigns a Coder Group used for extension coders, added to the SDP offer in the outgoing leg for the SIP entity associated with the IP Profile. This is used when transcoding is required between two IP entities (i.e., the SDP answer from one doesn’t include any coder included in the offer previously sent by the other). For more information on extension coders and transcoding, see Coder Transcoding on page 467, To configure Coder Groups, see Configuring Coder Groups on page 407.</td>
</tr>
<tr>
<td>Allowed Audio Coders</td>
<td>Assigns an Allowed Audio Coders Group, which defines audio (voice) coders that can be used for the SIP entity associated with the IP Profile. To configure Allowed Audio Coders Groups, see Configuring Allowed Audio Coder Groups on page 412. For a description of the Allowed Coders feature, see &quot;Restricting Coders&quot; on page 466.</td>
</tr>
<tr>
<td>Allowed Coders Mode</td>
<td>Defines the mode of the Allowed Coders feature for the SIP entity associated with the IP Profile.</td>
</tr>
</tbody>
</table>

- **[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' parameter, above), these coders are added to the SDP offer if they also appear in Allowed coders.

- **[1] Preference =** The device re-arranges the priority (order) of the coders in the incoming SDP offer according to their order of appearance in the Allowed Audio Coders Group or Allowed Video Coders Group. The coders in the original SDP offer are listed after the Allowed coders.

- **[2] Restriction and Preference =** Performs both Restriction and Preference.

**Note:**
- The parameter is applicable only if Allowed coders are assigned to the IP Profile (see the 'Allowed Audio Coders' or 'Allowed Video Coders' parameters).
- For more information on the Allowed Coders feature, see Restricting Coders on page 466.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allowed Video Coders</td>
<td>Assigns an Allowed Video Coders Group. This defines permitted video coders when forwarding video streams to the SIP entity associated with the IP Profile. The video coders are listed in the &quot;video&quot; 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. 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 415.</td>
</tr>
<tr>
<td>Allowed Media Types</td>
<td>Defines media types permitted for the SIP entity associated with the IP Profile. The media type appears in the SDP 'm=' line (e.g., 'm=audio'). The device permits only media types that appear in both the SDP offer and this configured list. If no common media types exist between the SDP offer and this list, the device drops the call. The valid value is a string of up to 64 characters. To configure multiple media types, separate the strings with a comma, e.g., &quot; audio, text&quot; (without quotes). By default, no media types are configured (i.e., all media types are permitted).</td>
</tr>
<tr>
<td>Direct Media Tag</td>
<td>Defines an identification tag for enabling direct media (no Media Anchoring) for the SIP entity associated with the IP Profile. Direct media occurs between all endpoints whose IP Profiles have the same tag value (non-empty value). For example, if you set the parameter to &quot;direct-rtp&quot; for two IP Profiles &quot;IP-PBX-1&quot; and &quot;IP-PBX-2&quot;, the device employs direct media for calls amongst endpoints associated with IP Profile &quot;IP-PBX-1&quot;, for calls amongst endpoints associated with IP Profile &quot;IP-PBX-2&quot;, and for calls between endpoints associated with IP Profile &quot;IP-PBX-1&quot; and IP Profile &quot;IP-PBX-2&quot;. The valid value is a string of up to 16 characters. By default, no value is defined. For more information on direct media, see Direct Media on page 464. <strong>Note:</strong> If you enable direct media for the IP Profile, make sure that your Media Realm provides sufficient ports, as media may traverse the device for mid-call services (e.g., call transfer).</td>
</tr>
</tbody>
</table>
| RFC 2833 Mode | Defines the handling of RFC 2833 SDP offer-answer negotiation for the SIP entity associated with the IP Profile.  
  - [0] As is = (Default) The device does not intervene in the RFC 2833 negotiation.  
  - [1] Extend = Each outgoing offer-answer includes RFC 2833 in the offered SDP. The device adds RFC 2833 only if the incoming offer does not include RFC 2833.  
**Note:** If the device interworks between different DTMF
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFC 2833 DTMF Payload Type</td>
<td>Defines the payload type of DTMF digits for the SIP entity associated with the IP Profile. This enables the interworking of the DTMF payload type for RFC 2833 between different SBC call legs. For example, if two entities require different DTMF payload types, the SDP offer received by the device from one entity is forwarded to the destination entity with its payload type replaced with the configured payload type, and vice versa. The value range is 0 to 200. The default is 0 (i.e., the device forwards the received payload type as is).</td>
</tr>
<tr>
<td>Alternative DTMF Method</td>
<td>The device’s first priority for DTMF method at each leg is RFC 2833. Thus, if the device successfully negotiates RFC 2833 for the SIP entity associated with the IP Profile, the chosen DTMF method for this leg is RFC 2833. When RFC 2833 negotiation fails, the device uses the parameter to define the DTMF method for the leg.</td>
</tr>
<tr>
<td>Send Multiple DTMF Methods</td>
<td>Enables the device to send DTMF digits out-of-band (not with audio stream) using both the SIP INFO and RFC 2833 methods for the same call on the leg to which this IP Profile is associated. The RFC 2833 method sends out-of-band DTMF digits using the RTP protocol while the SIP INFO method sends the digits using the SIP protocol.</td>
</tr>
</tbody>
</table>

**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.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Name=' | If the IP Profile name contains one or more spaces, you must enclose the name in double quotation marks, for example:  
  X-AC-Action: 'switch-profile;profile-name="My IP Profile"' |

The Message Manipulation rule adds the proprietary header with the value of the new IP Profile to the incoming re-INVITE or UPDATE message and as a result, the device uses the new IP Profile for the SIP entity and stops sending it SIP INFO messages. You can also configure an additional Message Manipulation rule to re-start the sending of the SIP INFO. For example, you can configure two Message Manipulation rules where the sending of both SIP INFO and RFC 2833 depends on the negotiated media port — the device stops sending SIP INFO if the SDP of the re-INVITE or UPDATE message contains port 7550 and re-starts sending if the port is 8660. The rule that re-starts the SIP INFO switches the IP Profile back to the initial IP Profile that enables the sending of DTMF digits using both methods (i.e., "Send Multiple DTMF Methods" is configured to **Enable**). The configured Message Manipulation rules for this example are shown below:

- **Index 1**
  - Message Type: reinvite.request
  - Condition: body.sdp regex (.*)(m=audio 7550 RTP/AVP)(.*)
  - Action Subject: header.X-AC-Action
  - Action Type: Add
  - Action Value: 'switch-profile;profile-name=ITSP-Profile-2'

- **Index 2**
  - Message Type: reinvite.request
  - Condition: body.sdp regex (.*)(m=audio 8660 RTP/AVP)(.*)
  - Action Subject: header.X-AC-Action
  - Action Type: Add
  - Action Value: 'switch-profile;profile-name=ITSP-Profile-1'

The Message Manipulation rules must be assigned to the SIP entity's IP Group, using the 'Inbound Message Manipulation Set' parameter.

**Note:**
- To send DTMF digits using both methods (i.e., when the parameter is enabled), you need to also configure the following:
  - Configure the 'Alternative DTMF Method' (IPProfile_SBCAlternativeDTMFMethod) parameter to one of the SIP INFO options (**INFO – Cisco**, **INFO – Nortel**, or **INFO – Lucent**).
  - Enable the sending of DTMF digits using the RFC 2833 method, by configuring the 'RFC 2833 Mode' (ipProfile_SBCRFC2833Behavior) parameter to **As**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Is or Extend.**                  | - When using the X-AC-Action header to switch IP Profiles, it is recommended that the settings of the switched IP Profile are identical (except for the 'Send Multiple DTMF Methods' parameter) to the initial IP Profile in order to avoid any possible call handling errors.  
- This feature requires DSP resources (for detection and generation of RFC 2833). |
| SDP Ptime Answer                   | Defines the packetization time (ptime) of the coder in RTP packets for the SIP entity associated with the IP Profile. This is useful when implementing transrating.  
- [0] Remote Answer = (Default) Use ptime according to SDP answer.  
- [1] Original Offer = Use ptime according to SDP offer.  
- [2] Preferred Value= Use preferred ptime for negotiation, if configured by the 'Preferred Ptime' parameter. |
| Preferred Ptime                    | Defines the packetization time (in msec) for the SIP entity associated with the IP Profile if the 'SBC SDP Ptime Answer' parameter (see above) is set to Preferred Value. The valid range is 0 to 200. The default is 0 (i.e., preferred ptime is not used). |
| Use Silence Suppression            | Defines silence suppression support for the SIP entity associated with the IP Profile  
- [0] Transparent = (Default) Forward as is.  
- [1] Add = Enable silence suppression for each relevant coder listed in the SDP.  
- [2] Remove = Disable silence suppression for each relevant coder listed in the SDP.  
**Note:** The parameter requires DSP resources. |
| RTP Redundancy Mode                | Enables interworking RTP redundancy negotiation support between SIP entities in the SDP offer-answer exchange (according to RFC 2198). The parameter defines the device's handling of RTP redundancy for the SIP entity associated with the IP Profile. According to the RTP redundancy SDP offer/answer negotiation, the device uses or discards the RTP redundancy packets. The parameter enables asymmetric RTP redundancy, whereby the device can transmit and receive RTP redundancy packets to and from a specific SIP entity, while transmitting and receiving regular RTP packets (no redundancy) for the other SIP entity involved in the voice path.  
The device can identify the RTP redundancy payload type in the SDP for indicating that the RTP packet stream includes redundant packets. RTP redundancy is indicated in SDP using the "red" coder type, for example:  
`a=rtpmap:<payload type> red/8000/1`  
RTP redundancy is useful when there is packet loss; the missing information may be reconstructed at the receiver side from the redundant packets. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>▪ [0] As Is = (Default) The device does not interfere in the RTP redundancy negotiation and forwards the SDP offer/answer (incoming and outgoing calls) as is without interfering in the RTP redundancy negotiation.</td>
</tr>
<tr>
<td></td>
<td>▪ [1] Enable = The device always adds RTP redundancy capabilities in the outgoing SDP offer sent to the SIP entity. Whether RTP redundancy is implemented depends on the subsequent incoming SDP answer from the SIP entity. The device does not modify the incoming SDP offer received from the SIP entity, but if RTP redundancy is required, it will be supported. Select the option if the SIP entity requires RTP redundancy.</td>
</tr>
<tr>
<td></td>
<td>▪ [2] Disable = The device removes the RTP redundancy payload (if present) from the SDP offer/answer for calls received from or sent to the SIP entity. Select the option if the SIP entity does not support RTP redundancy.</td>
</tr>
</tbody>
</table>

**Note:**
- To enable the device to generate RFC 2198 redundant packets, use the IPProfile_RTPRedundancyDepth parameter.
- To configure the payload type in the SDP offer for RTP redundancy, use the RFC2198PayloadType.

**RTCP Mode**

`sbc-rtcp-mode [IPProfile_SBCRTCPMode]`

Defines how the device handles RTCP packets during call sessions for the SIP entity associated with the IP Profile. This is useful for interworking RTCP between SIP entities. For example, this may be necessary when incoming RTCP is not compatible with the destination SIP entity's (this IP Profile) RTCP support. In such a scenario, the device can generate the RTCP and send it to the SIP entity.

▪ [0] Transparent = (Default) RTCP is forwarded as is (unless transcoding is done, in which case, the device generates RTCP on both legs).

▪ [1] Generate Always = Generates RTCP packets during active and inactive (e.g., during call hold) RTP periods (i.e., media is 'a=recvonly' or 'a=inactive' in the INVITE SDP).

▪ [2] Generate only if RTP Active = Generates RTCP packets only during active RTP periods. In other words, the device does not generate RTCP when there is no RTP traffic (such as when a call is on hold).

**Note:** The corresponding global parameter is SBCRTCPMode.

**Jitter Compensation**

`sbc-jitter-compensation [IpProfile_SBCJitterCompensation]`

Enables the on-demand jitter buffer for SBC calls. The jitter buffer can be used when other functionality such as voice transcoding are not done on the call. The jitter buffer is useful when incoming packets are received at inconsistent intervals (i.e., packet delay variation). The jitter buffer stores the packets and sends them out at a constant rate (according to the coder's settings).

▪ [0] Disable (default)

▪ [1] Enable
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>- 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.</td>
</tr>
<tr>
<td></td>
<td>- This functionality may require DSP resources. For more information, contact your AudioCodes sales representative.</td>
</tr>
<tr>
<td></td>
<td><strong>ICE Mode</strong></td>
</tr>
<tr>
<td>ice-mode</td>
<td>Enables Interactive Connectivity Establishment (ICE) Lite for the SIP entity associated with the IP Profile. ICE is a methodology for NAT traversal, employing the Session Traversal Utilities for NAT (STUN) and Traversal Using Relays around NAT (TURN) protocols to provide a peer with a public IP address and port that can be used to connect to a remote peer.</td>
</tr>
<tr>
<td>[IPProfile_SBCIceMode]</td>
<td>- [0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>- [1] Lite</td>
</tr>
<tr>
<td></td>
<td>For more information on ICE Lite, see ICE Lite.</td>
</tr>
<tr>
<td></td>
<td><strong>SDP Handle RTCP</strong></td>
</tr>
<tr>
<td>sbc-sdp-handle-rtcp</td>
<td>Enables the interworking of the RTCP attribute, ‘a=rtcp’ (RTCP) in the SDP, for the SIP entity associated with the IP Profile. The RTCP attribute is used to indicate the RTCP port for media when that port is not the next higher port number following the RTP port specified in the media line (‘m=’).</td>
</tr>
<tr>
<td>[IPProfile_SBCSDPHandleRTCPAttribute]</td>
<td>The parameter is useful for SIP entities that either require the attribute or do not support the attribute. For example, Google Chrome and Web RTC do not accept calls without the RTCP attribute in the SDP. In Web RTC, Chrome (SDES) generates the SDP with ‘a=rtcp’, for example:</td>
</tr>
<tr>
<td></td>
<td>- [0] Don’t Care = (Default) The device forwards the SDP as is without interfering in the RTCP attribute (regardless if present or not).</td>
</tr>
<tr>
<td></td>
<td>- [1] Add = The device adds the ‘a=rtcp’ attribute to the outgoing SDP offer sent to the SIP entity if the attribute was not present in the original incoming SDP offer.</td>
</tr>
<tr>
<td></td>
<td>- [2] Remove = The device removes the ‘a=rtcp’ attribute, if present in the incoming SDP offer received from the other SIP entity, before sending the outgoing SDP offer to the SIP entity.</td>
</tr>
<tr>
<td></td>
<td>For more information on SDP Handle RTCP, see page 466.</td>
</tr>
<tr>
<td></td>
<td><strong>RTCP Mux</strong></td>
</tr>
<tr>
<td></td>
<td>Enables interworking of multiplexing of RTP and RTCP</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>sbc-rtcp-mux</td>
<td>onto a single local port, between SIP entities. The parameter enables multiplexing of RTP and RTCP traffic onto a single local port, for the SIP entity associated with the IP Profile.</td>
</tr>
<tr>
<td>[IPProfile_SBCRTCPMux]</td>
<td>Multiplexing of RTP data packets and RTCP packets onto a single local UDP port is done for each RTP session (according to RFC 5761). If multiplexing is not enabled, the device uses different (but adjacent) ports for RTP and RTCP packets. With the increased use of NAT and firewalls, maintaining multiple NAT bindings can be costly and also complicate firewall administration since multiple ports must be opened to allow RTP traffic. To reduce these costs and session setup times, support for multiplexing RTP data packets and RTCP packets onto a single port is advantageous. For multiplexing, the initial SDP offer must include the &quot;a=rtcp-mux&quot; attribute to request multiplexing of RTP and RTCP onto a single port. If the SDP answer wishes to multiplex RTP and RTCP, it must also include the &quot;a=rtcp-mux&quot; attribute. If the answer does not include the attribute, the offerer must not multiplex RTP and RTCP packets. If both ICE and multiplexed RTP-RTCP are used, the initial SDP offer must also include the &quot;a=candidate:&quot; attribute for both RTP and RTCP along with the &quot;a=rtcp:&quot; attribute, indicating a fallback port for RTCP in case the answerer does not support RTP and RTCP multiplexing.</td>
</tr>
<tr>
<td></td>
<td>• [0] Not Supported = (Default) RTP and RTCP packets use different ports.</td>
</tr>
<tr>
<td></td>
<td>• [1] Supported = Device multiplexes RTP and RTCP packets onto a single port.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> As RTP multiplexing has been defined by the WebRTC standard as mandatory, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 566.</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RTCP Feedback</td>
<td>Enables RTCP-based feedback indication in outgoing SDPs sent to the SIP entity associated with the IP Profile. The parameter supports indication of RTCP-based feedback, according to RFC 5124, during RTP profile negotiation between two communicating SIP entities. RFC 5124 defines an RTP profile (S)AVPF for (secure) real-time communications to provide timely feedback from the receivers to a sender. For more information on RFC 5124, see <a href="http://tools.ietf.org/html/rfc5124">http://tools.ietf.org/html/rfc5124</a>. Some SIP entities may require RTP secure-profile feedback negotiation (AVPF/SAVPF) in the SDP offer/answer exchange, while other SIP entities may not support it. The device indicates whether or not feedback is supported on behalf of the SIP entity. It does this by adding an &quot;F&quot; or removing the &quot;F&quot; from the SDP media line ('m=') for AVP and SAVP. For example, the following shows &quot;AVP&quot; appended with an &quot;F&quot;, indicating that the SIP entity is capable of receiving feedback.</td>
</tr>
<tr>
<td>sbc-rtcp-feedback</td>
<td></td>
</tr>
<tr>
<td>[IPProfile_SBCRTCPFeedback]</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>m=audio 49170 RTP/SAVPF 0 96</td>
<td></td>
</tr>
</tbody>
</table>
  - [0] Feedback Off = (Default) The device does not send the feedback flag ("F") in SDP offers/answers that are sent to the SIP entity. If the SDP 'm=' attribute of an incoming message that is destined to the SIP entity includes the feedback flag, the device removes it before sending the message to the SIP entity.  
  - [1] Feedback On = The device includes the feedback flag ("F") in the SDP offer sent to the SIP entity. The device includes the feedback flag in the SDP answer sent to the SIP entity only if it was present in the SDP offer received from the other SIP entity.  
  - [2] As Is = The device does not involve itself in the feedback, but simply forwards any feedback indication as is.  
  
  **Note:**  
  - As RTCP-based feedback has been defined by the WebRTC standard as mandatory, the support is important for deployments implementing WebRTC. For more information on WebRTC, see WebRTC on page 566.  
  - RTCP-based feedback is required for the VolPerfect feature (see VolPerfect on page 580).  

**Voice Quality Enhancement**

sbc-voice-quality-enhancement

[IpProfile_SBCVoiceQualityEnhancement]

Enables the device to detect speech and network quality (packet loss and bandwidth reduction) and triggers the device to overcome adverse conditions to ensure high call quality.  

- [0] Disable (default)  
- [1] Enable  

**Note:** The parameter is applicable only to the VolPerfect feature (see VolPerfect on page 580).

**Max Opus Bandwidth**

sbc-max-opus-bandwidth

[IpProfile_SBCMaxOpusBW]

Defines the VolPerfect mode of operation, which is based on the Opus coder.  

- [0] = (Default) Managed Opus  
- 80000 = Smart Transcoding  

**Note:** The parameter is applicable only to the VolPerfect feature (see VolPerfect on page 580).

**Quality of Service**

**RTP IP DiffServ**

rtp-ip-diffserv

[IpProfile_IPDiffServ]

Defines the DiffServ value for Premium Media class of service (CoS) content.  

The valid range is 0 to 63. The default is 46.  

**Note:** The corresponding global parameter is PremiumServiceClassMediaDiffServ.

**Signaling DiffServ**

signaling-diffserv

[IpProfile_SigIPDiffServ]

Defines the DiffServ value for Premium Control CoS content (Call Control applications).  

The valid range is 0 to 63. The default is 40.  

**Note:** The corresponding global parameter is PremiumServiceClassControlDiffServ.

**Jitter Buffer**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Jitter Buffer Minimum Delay</td>
<td>Defines the minimum delay (in msec) of the device’s dynamic Jitter Buffer. The valid range is 0 to 150. The default delay is 10. For more information on Jitter Buffer, see Configuring the Dynamic Jitter Buffer on page 193. <strong>Note:</strong> The corresponding global parameter is DJBufMinDelay.</td>
</tr>
<tr>
<td>jitter-buffer-minimum-delay [IpProfile_JitterBufMinDelay]</td>
<td></td>
</tr>
</tbody>
</table>
| Dynamic Jitter Buffer Optimization Factor           | Defines the Dynamic Jitter Buffer frame error/delay optimization factor. The valid range is 0 to 12. The default factor is 10. For more information on Jitter Buffer, see Configuring the Dynamic Jitter Buffer on page 193. **Note:**  
  - For data (fax and modem) calls, set the parameter to 12.  
  - The corresponding global parameter is DJBufOptFactor.                                                                                          |
| jitter-buffer-optimization-factor [IpProfile_JitterBufOptFactor] |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| Jitter Buffer Max Delay                             | Defines the maximum delay and length (in msec) of the Jitter Buffer. The valid range is 150 to 2,000. The default is 250.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| [IpProfile_JitterBufMaxDelay]                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| **Voice**                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Echo Canceler                                       | Enables the device’s Echo Cancellation feature (i.e., echo from voice calls is removed).  
  - [0] Disable  
  - [1] Line (default)  
  For a detailed description of the Echo Cancellation feature, see Configuring Echo Cancellation on page 181. **Note:** The corresponding global parameter is EnableEchoCanceller.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| echo-canceller [IpProfile_EnableEchoCanceller]       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Input Gain                                          | Defines the pulse-code modulation (PCM) input gain control (in decibels). The valid range is -32 to 31 dB. The default is 0 dB. **Note:** The corresponding global parameter is InputGain.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| [IpProfile_InputGain]                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Voice Volume                                        | Defines the voice gain control (in decibels). The valid range is -32 to 31 dB. The default is 0 dB. **Note:** The corresponding global parameter is VoiceVolume.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| [IpProfile_VoiceVolume]                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| **SBC Signaling**                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| PRACK Mode                                          | Defines the device’s handling of SIP PRACK messages for the SIP entity associated with the IP Profile.  
  - [1] Optional = PRACK is optional. If required, the device performs the PRACK process on behalf of the SIP entity.  
  - [2] Mandatory = PRACK is required for this SIP entity. Calls from endpoints that do not support PRACK are                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
<p>| sbc-prack-mode [IpProfile_SbcPrackMode]             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **P-Asserted-Identity Header Mode**<br>sbc-assert-identity<br>[IpProfile_SBCAssertIdentity] | Defines the device’s handling of the SIP P-Asserted-Identity header for the SIP entity associated with the IP Profile. This header indicates how the outgoing SIP message asserts identity.  
- **[0]** As Is = (Default) P-Asserted Identity header is not affected and the device uses the same P-Asserted-Identity header (if present) in the incoming message for the outgoing message.  
- **[1]** Add = Adds a P-Asserted-Identity header. The header’s values are taken from the source URL.  
- **[2]** Remove = Removes the P-Asserted-Identity header.  
**Note:**  
- The parameter affects only the initial INVITE request.  
- The corresponding global parameter is SBCAssertIdentity. |
| **Diversion Header Mode**<br>sbc-diversion-mode<br>[IpProfile_SBCDiversionMode] | Defines the device’s handling of the SIP Diversion header for the SIP entity associated with the IP Profile.  
- **[0]** As Is = (Default) Diversion header is not handled.  
- **[1]** Add = History-Info header is converted to a Diversion header.  
- **[2]** Remove = Removes the Diversion header and the conversion to the History-Info header depends on the SBCHistoryInfoMode parameter.  
For more information on interworking of the History-Info and Diversion headers, see Interworking SIP Diversion and History-Info Headers on page 478.  
**Note:** If the Diversion header is used, you can specify the URI type (e.g., "tel:" ) to use in the header, using the SBCDiversionUnType parameter. |
| **History-Info Header Mode**<br>sbc-history-info-mode<br>[IpProfile_SBCHistoryInfoMode] | Defines the device’s handling of the SIP History-Info header for the SIP entity associated with the IP Profile.  
- **[0]** As Is = (Default) History-Info header is not handled.  
- **[1]** Add = Diversion header is converted to a History-Info header.  
- **[2]** Remove = History-Info header is removed from the SIP dialog and the conversion to the Diversion header depends on the SBCDiversionMode parameter.  
For more information on interworking of the History-Info and Diversion headers, see Interworking SIP Diversion and History-Info Headers on page 478. |
| **Session Expires Mode**<br>sbc-session-expires-mode<br>[IpProfile_SBCSessionExpiresMode] | Defines the required session expires mode for the SIP entity associated with the IP Profile.  
- **[0]** Transparent = (Default) The device does not interfere with the session expires negotiation.  
- **[1]** Observer = If the SIP Session-Expires header is
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| present, the device does not interfere, but maintains an independent timer for each leg to monitor the session. If the session is not refreshed on time, the device disconnects the call.  
- [2] Not Supported = The device does not allow a session timer with this SIP entity.  
- [3] Supported = The device enables the session timer with this SIP entity. If the incoming SIP message does not include any session timers, the device adds the session timer information to the sent message. You can configure the value of the Session-Expires and Min-SE headers, using the SBCSessionExpires and SBCMinSE parameters, respectively. |
| Remote Update Support  
sbc-rmt-update-suppp  
[IpProfile_SBCRemoteUpdateSupport] | Defines whether the SIP UPDATE message is supported by the SIP entity associated with the IP Profile.  
- [0] Not Supported = UPDATE message is not supported.  
- [1] Supported Only After Connect = UPDATE message is supported only after the call is connected.  
- [2] Supported = (Default) UPDATE message is supported during call setup and after call establishment. |
| Remote re-INVITE  
sbc-rmt-re-invite-suppp  
[IpProfile_SBCRemoteReinviteSupport] | Defines whether the destination UA of the re-INVITE request supports re-INVITE messages and if so, whether it supports re-INVITE with or without SDP.  
- [0] Not Supported = re-INVITE is not supported and the device does not forward re-INVITE requests. The device sends a SIP response to the re-INVITE request, which can either be a success or a failure, depending on whether the device can bridge the media between the endpoints.  
- [1] Supported only with SDP = re-INVITE is supported, but only with SDP. If the incoming re-INVITE arrives without SDP, the device creates an SDP and adds it to the outgoing re-INVITE.  
- [2] Supported = (Default) re-INVITE is supported with or without SDP. |
| Remote Delayed Offer Support  
sbc-rmt-delayed-offer  
[IpProfile_SBCRemoteDelayedOfferSupport] | Defines whether the remote endpoint supports delayed offer (i.e., initial INVITEs without an SDP offer).  
- [0] Not Supported = Initial INVITE requests without SDP are not supported.  
- [1] Supported = (Default) Initial INVITE requests without SDP are supported. **Note:** For the parameter to function, you need to assign extension coders to the IP Profile of the SIP entity that does not support delayed offer (using the IpProfile_SBCExtensionCodersGroupName parameter). |
| Remote Representation Mode  
sbc-rmt-rprsntnation  
[IpProfile_SBCRemoteRepresentationMode] | Enables interworking SIP in-dialog, Contact and Record-Route headers between SIP entities. The parameter defines the device's handling of in-dialog, Contact and Record-Route headers for messages sent to the SIP entity associated with the IP Profile. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| - [1] According to Operation Mode = (Default) Depends on the setting of the 'Operation Mode' parameter in the IP Group or SRDs table:  
  - B2BUA: Device operates as if the parameter is set to Replace Contact [0].  
  - Call State-full Proxy: Device operates as if the parameter is set to Add Routing Headers [1].  
- [0] Replace Contact = Device replaces the address in the Contact header, received in incoming messages from the other side, with its own address in the outgoing message sent to the SIP entity.  
- [1] Add Routing Headers = Device adds a Record-Route header for itself to outgoing messages (requests/responses) sent to the SIP entity in dialog-setup transactions. The Contact header remains unchanged.  
- [2] Transparent = Device doesn't change the Contact header and doesn't add a Record-Route header for itself. Instead, it relies on its own inherent mechanism to remain in the route of future requests in the dialog (for example, relying on the way the endpoints are set up or on TLS as the transport type). |

Keep Incoming Via Headers  
**sbc-keep-via-headers [IpProfile_SBCKeepVIAHeaders]**  
Enables interworking SIP Via headers between SIP entities. The parameter defines the device's handling of Via headers for messages sent to the SIP entity associated with the IP Profile.  
- [-1] According to Operation Mode = Depends on the setting of the 'Operation Mode' parameter in the IP Groups table or SRDs table:  
  - B2BUA: Device operates as if the parameter is set to Disable [0].  
  - Call State-full Proxy: Device operates as if the parameter is set to Enable [1].  
- [0] Disable = Device removes all Via headers received in the incoming SIP request from the other leg and adds a Via header identifying only itself, in the outgoing message sent to the SIP entity.  
- [1] Enable = Device retains the Via headers received in the incoming SIP request and adds itself as the top-most listed Via header in the outgoing message sent to the SIP entity. |

Keep Incoming Routing Headers  
**sbc-keep-routing-headers [IpProfile_SBCKeepRoutingHeaders]**  
Enables interworking SIP Record-Route headers between SIP entities. The parameter defines the device's handling of Record-Route headers for request/response messages sent to the SIP entity associated with the IP Profile.  
- [-1] According to Operation Mode = (Default) Depends on the setting of the 'Operation Mode' in the IP Group or SRDs table:  
  - B2BUA: Device operates as if the parameter is set to Disable [0].  
  - Call State-full Proxy: Device operates as if the parameter is set to Enable [1].  
- [0] Disable = Device removes the Record-Route header from the outgoing message sent to the SIP entity.  
- [1] Enable = Device adds itself as the top-most listed Record-Route header in the outgoing message sent to the SIP entity.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>headers received in requests and responses from the other side, in the outgoing SIP message sent to the SIP entity. The device creates a route set for that side of the dialog based on these headers, but doesn't send them to the SIP entity.</td>
<td></td>
</tr>
<tr>
<td>[1] Enable = Device retains the incoming Record-Route headers received in requests and non-failure responses from the other side, in the following scenarios:</td>
<td></td>
</tr>
<tr>
<td>✓ The message is part of a SIP dialog-setup transaction.</td>
<td></td>
</tr>
<tr>
<td>✓ The messages in the setup and previous transaction didn't include the Record-Route header, and therefore hadn't set the route set.</td>
<td></td>
</tr>
<tr>
<td>Note: Record-Routes are kept only for SIP INVITE, UPDATE, SUBSCRIBE and REFER messages.</td>
<td></td>
</tr>
<tr>
<td>Keep User-Agent Header</td>
<td>Enables interworking SIP User-Agent headers between SIP entities. The parameter defines the device's handling of User-Agent headers for response/request messages sent to the SIP entity associated with the IP Profile.</td>
</tr>
<tr>
<td>[0] Disable = Device removes the User-Agent/Server headers received in the incoming message from the other side, and adds its' own User-Agent header in the outgoing message sent to the SIP entity.</td>
<td></td>
</tr>
<tr>
<td>[1] Enable = Device retains the User-Agent/Server headers received in the incoming message and sends the headers as is in the outgoing message to the SIP entity.</td>
<td></td>
</tr>
<tr>
<td>Handle X-Detect</td>
<td>Enables the detection and notification of events (AMD, CPT, and fax), using the X-Detect SIP header.</td>
</tr>
<tr>
<td>[0] No (default)</td>
<td></td>
</tr>
<tr>
<td>[1] Yes</td>
<td></td>
</tr>
<tr>
<td>For more information, see Event Detection and Notification using X-Detect Header on page 196.</td>
<td></td>
</tr>
<tr>
<td>ISUP Body Handling</td>
<td>Defines the handling of ISUP data for interworking between SIP and SIP-I endpoints.</td>
</tr>
<tr>
<td>[0] Transparent = (Default) ISUP data is passed transparently (as is) between endpoints (SIP-I to SIP-I calls).</td>
<td></td>
</tr>
<tr>
<td>[1] Remove = ISUP body is removed from INVITE messages.</td>
<td></td>
</tr>
<tr>
<td>[2] Create = ISUP body is added to outgoing INVITE messages.</td>
<td></td>
</tr>
<tr>
<td>For more information on interworking SIP and SIP-I, see</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ISUP Variant</td>
<td>Defines the ISUP variant for interworking SIP and SIP-I endpoints.</td>
</tr>
</tbody>
</table>
| sbc-isup-variant              | • [0] itu92 = (Default) ITU 92 variant  
• [1] Spirou = SPIROU (ISUP France)                                                                                                                             |
| Max Call Duration             | Defines the maximum duration (in minutes) per SBC call that is associated with the IP Profile. If the duration is reached, the device terminates the call. The valid range is 0 to 35,791, where 0 is unlimited duration. The default is the value configured for the global parameter, SBCMaxCallDuration. |
| SBC Registration              |                                                                                                                                                                                                            |
| User Registration Time        | Defines the registration time (in seconds) that the device responds to SIP REGISTER requests from users belonging to the SIP entity associated with the IP Profile. The registration time is inserted in the Expires header in the outgoing response sent to the user. |
| sbc-usr-reg-time              | The Expires header determines the lifespan of the registration. For example, a value of 3600 means that the registration will timeout in one hour and at that point, the user will not be able to make or receive calls.            |
| [IpProfile_SBCUserRegistrationTime] | The valid range is 0 to 2,000,000. The default is 0. If configured to 0, the Expires header's value received in the user's REGISTER request remains unchanged. If no Expires header is received in the REGISTER message and the parameter is set to 0, the Expires header's value is set to 180 seconds, by default. |
| Note:                         | The corresponding global parameter is SBCUserRegistrationTime.                                                                                                                                             |
| NAT UDP Registration Time     | Defines the registration time (in seconds) that the device includes in register responses, in response to SIP REGISTER requests from users belonging to the SIP entity associated with the IP Profile. The parameter applies only to users that are located behind NAT and whose communication type is UDP. The registration time is inserted in the Expires header in the outgoing response sent to the user. |
| sbc-usr-udp-nat-reg-time      | The Expires header determines the lifespan of the registration. For example, a value of 3600 means that the registration will timeout in one hour, unless the user sends a refresh REGISTER before the timeout. Upon timeout, the device removes the user's details from the registration database, and the user will not be able to make or receive calls through the device. |
| [IpProfile_SBCUserBehindUdpNATRegistrationTime] | The valid value is 0 to 2,000,000. If configured to 0, the Expires header's value received in the user's REGISTER request remains unchanged. By default, no value is defined (-1). |
| Note:                         | If the parameter is not configured, the registration time is according to the global parameter SBCUserRegistrationTime or IP Profile parameter.                                                                |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IpProfile_SBCUserRegistrationTime.</td>
<td>Defines the registration time (in seconds) that the device includes in register responses, in response to SIP REGISTER requests from users belonging to the SIP entity associated with the IP Profile. The parameter applies only to users that are located behind NAT and whose communication type is TCP. The registration time is inserted in the Expires header in the outgoing response sent to the user. The Expires header determines the lifespan of the registration. For example, a value of 3600 means that the registration will timeout in one hour, unless the user sends a refresh REGISTER before the timeout. Upon timeout, the device removes the user’s details from the registration database, and the user will not be able to make or receive calls through the device. The valid value is 0 to 2,000,000. If configured to 0, the Expires header’s value received in the user’s REGISTER request remains unchanged. By default, no value is defined (-1). <strong>Note:</strong> If the parameter is not configured, the registration time is according to the global parameter SBCUserRegistrationTime or IP Profile parameter IpProfile_SBCUserRegistrationTime.</td>
</tr>
</tbody>
</table>

### SBC Forward and Transfer

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IpProfile_SBCRemoteReferBehavior</td>
<td>Defines the device's handling of SIP REFER requests for the IP entity (transferee - call party that is transferred to the transfer target) associated with the IP Profile.</td>
</tr>
<tr>
<td>[0] Regular = (Default) SIP Refer-To header value is unchanged and the device forwards the REFER message as is.</td>
<td></td>
</tr>
<tr>
<td>[1] Database URL = SIP Refer-To header value is changed so that the re-routed INVITE is sent through the device:</td>
<td></td>
</tr>
<tr>
<td>a. Before forwarding the REFER request, the device changes the host part to the device's IP address and adds a special prefix (&quot;T~&amp;R_&quot;) to the Contact user part.</td>
<td></td>
</tr>
<tr>
<td>b. The incoming INVITE is identified as a REFER-resultant INVITE according to this special prefix.</td>
<td></td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>d. The special prefix is removed before the resultant INVITE is sent to the destination.</td>
<td></td>
</tr>
<tr>
<td>[2] IP Group Name = Sets the host part in the REFER message to the name defined for the IP Group (in the IP Groups table).</td>
<td></td>
</tr>
<tr>
<td>[3] Handle Locally = Handles the incoming REFER</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>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' parameter must be set to REFER).</td>
<td></td>
</tr>
<tr>
<td>[4] Local Host = In the REFER message received from the transferor, the device replaces the Refer-To header value (URL) with the IP address of the device or with the 'Local Host Name' parameter value configured for the IP Group (transferee) to where the device forwards the REFER message. This ensures that the transferee sends the re-routed INVITE back to the device which then sends the call to the transfer target.</td>
<td></td>
</tr>
<tr>
<td>[5] Keep Host = The device forwards the REFER message without changing the host part in the SIP Refer-To header. This applies to all types of call transfers (e.g., blind and attendant transfer).</td>
<td></td>
</tr>
<tr>
<td>Note: The corresponding global parameter is SBCReferBehavior.</td>
<td></td>
</tr>
</tbody>
</table>

Remote Replaces Mode

sbc-rmt-replaces-behavior [IpProfile_SBCRemoteReplacesBehavior]

Enables the device to handle incoming INVITEs containing the Replaces header for the SIP entity (which does not support the header) associated with the IP Profile. The Replaces header is used to replace an existing SIP dialog with a new dialog such as in call transfer or call pickup.

- [0] Standard = (Default) The SIP entity supports INVITE messages containing Replaces headers. The device forwards the INVITE message containing the Replaces header to the SIP entity. The device may change the value of the Replaces header to reflect the call identifiers of the leg.
- [1] Handle Locally = The SIP entity does not support INVITE messages containing Replaces headers. The device terminates the received INVITE containing the Replaces header and establishes a new call between the SIP entity and the new call party. It then disconnects the call with the initial call party, by sending it a SIP BYE request.
- [2] Keep as is = The SIP entity supports INVITE messages containing Replaces headers. The device forwards the Replaces header as is in incoming REFER and outgoing INVITE messages from/to the SIP entity (i.e., Replaces header's value is unchanged).

For example, assume that the device establishes a call between A and B. If B initiates a call transfer to C, the device receives an INVITE with the Replaces header from C. If A supports the Replaces header, the device simply forwards the INVITE as is to A; a new call is established between A and C and the call between A and B is disconnected. However, if A does not support the Replaces header, the device uses this feature to terminate the INVITE with Replaces header and handles the transfer for A. The device does this by connecting A to C, and disconnecting the call between A and B, by sending a SIP BYE request to B. Note that if media transcoding is required, the device sends an INVITE to C on behalf of A.
### 22. Coders and Profiles

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Play RBT To Transferee**  
`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 the SIP entity associated with the IP Profile (which does not support such a tone generation during call transfer). The ringback tone indicates to the transferee of the ringing of the transfer target (to where the transferee is being transferred).  
- [0] No (Default)  
- [1] Yes  

Typically, the transferee hears a ringback tone only if the transfer target sends it early media. However, if the transferee is put on-hold before being transferred, no ringback tone is heard.  

When this feature is enabled, the device generates a ringback tone to the transferee during call transfer in the following scenarios:  
- Transfer target sends a SIP 180 (Ringing) to the device.  
- For non-blind transfer, if the call is transferred while the transfer target is ringing and no early media occurs.  
- The "Remote Early Media RTP Behavior parameter is set to Delayed (used in the Skype for Business 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 660. |
| **Remote 3xx Mode**  
`sbc-rmt-3xx-behavior**  
`[IpProfile_SBCRemote3xxBehavior] | Defines the device’s handling of SIP 3xx redirect responses for the SIP entity associated with the IP Profile. By default, the device’s handling of SIP 3xx responses is to send the Contact header unchanged. However, some SIP entities may support different versions of the SIP 3xx standard while others may not even support SIP 3xx.  

When enabled, the device handles SIP redirections between different subnets (e.g., between LAN and WAN sides). This is required when the new address provided by the redirector (Redirect sever) may not be reachable by the far-end user (FEU) located in another subnet. For example, a far-end user (FEU) in the WAN sends a SIP request via the device to a Redirect server in the LAN, and the Redirect server replies with a SIP 3xx response to a PBX in the LAN in the Contact header. If the device sends this response as is (i.e., with the original Contact header), the
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEU is unable to reach the new destination.</td>
<td></td>
</tr>
<tr>
<td>• [0] Transparent = (Default) The device forwards the received SIP 3xx response as is, without changing the Contact header (i.e., transparent handling).</td>
<td></td>
</tr>
<tr>
<td>• [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 (&quot;T~&amp;R_&quot;), 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.</td>
<td></td>
</tr>
<tr>
<td>• [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).</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td>• When the parameter is changed from 1 to 0, new 3xx Contact headers remain unchanged. However, requests with the special prefix continue using the device’s database to locate the new destination.</td>
<td></td>
</tr>
<tr>
<td>• 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:</td>
<td></td>
</tr>
<tr>
<td>✓ sip:10.10.10.10:5060;transport=tcp;param=a</td>
<td></td>
</tr>
<tr>
<td>✓ sip:10.10.10.10:5060;transport=tcp;param=b</td>
<td></td>
</tr>
<tr>
<td>• The database entry expires two hours after the last use.</td>
<td></td>
</tr>
<tr>
<td>• The maximum number of destinations (i.e., database entries) is 50.</td>
<td></td>
</tr>
<tr>
<td>• The corresponding global parameter is SBC3xxBehavior.</td>
<td></td>
</tr>
</tbody>
</table>

**SBC Hold**

**Remote Hold Format**

**[IPProfile_SBCRemoteHoldFormat]**

Defines the format of the SDP in the SIP re-INVITE (or UPDATE) for call hold that the device sends to the held party.

<p>| • [0] Transparent = (Default) Device forwards SDP as is. |
| • [1] Send Only = Device sends SDP with 'a=sendonly'. |
| • [2] Send Only Zero ip = Device sends SDP with 'a=sendonly' and 'c=0.0.0.0'. |
| • [3] Inactive = Device sends SDP with 'a=inactive'. |
| • [4] Inactive Zero ip = Device sends SDP with 'a=inactive' and 'c=0.0.0.0'. |
| • [5] Not Supported = This option can be used when the remote side does not support call hold. The device terminates call hold requests received on the leg interfacing with the initiator of the call hold, and replies to this initiator with a SIP 200 OK response. However, call retrieve (resume) requests received from the initiator are forwarded to the remote side. The device... |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>can play a held tone to the held party if the 'SBC Play Held Tone' parameter is set to Yes.</strong>&lt;br&gt;• [6] Hold and Retrieve Not Supported = This option can be used when the remote side does not support call hold and retrieve (resume). The device terminates call hold and call retrieve requests received on the leg interfacing with the initiator of the call hold/retrieve, and replies to this initiator with a SIP 200 OK response. Therefore, the device does not forward call hold and/or retrieve requests to the remote side.</td>
<td></td>
</tr>
<tr>
<td><strong>Reliable Held Tone Source</strong>&lt;br&gt;[IPProfile_ReliableHoldToneSource]</td>
<td>Enables the device to consider the received call-hold request (re-INVITE/UPDATE) with SDP containing 'a=sendonly', as genuine.&lt;br&gt;• [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.&lt;br&gt;• [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).&lt;br&gt;&lt;b&gt;Note:&lt;/b&gt; The device plays a held tone only if the 'SBC Play Held Tone' parameter is set to Yes.</td>
</tr>
<tr>
<td><strong>Play Held Tone</strong>&lt;br&gt;[IpProfile_SBCPlayHeldTone]</td>
<td>Enables the device to play a hold tone to the call party that is placed on hold. This is useful if the held party does not support play of a local hold tone, or for IP entities initiating call hold that do not support the generation of hold tones.&lt;br&gt;• [0] No (default)&lt;br&gt;• [1] Yes&lt;br&gt;&lt;b&gt;Note:&lt;/b&gt; If you configure the parameter to Yes, the device plays the tone only if the 'SBC Remote Hold Format' parameter is configured to one of the following: send-only, send only 0.0.0.0, not supported, or transparent (when the incoming SDP is 'sendonly').</td>
</tr>
<tr>
<td><strong>SBC Fax</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Fax Coders Group</strong>&lt;br&gt;[IpProfile_SBCFaxCodersGroupName]</td>
<td>Assigns a Coder Group which defines the supported fax coders for fax negotiation for the SIP entity associated with the IP Profile. To configure Coder Groups, see Configuring Coder Groups on page 407.&lt;br&gt;&lt;b&gt;Note:&lt;/b&gt; The parameter is applicable only if you set the IpProfile_SBCFaxBehavior parameter to a value other than [0].</td>
</tr>
<tr>
<td><strong>Fax Mode</strong>&lt;br&gt;[IpProfile_SBCFaxBehavior]</td>
<td>Enables the device to handle fax offer-answer negotiations for the SIP entity associated with the IP Profile.&lt;br&gt;• [0] As Is = (Default) Device forwards fax transparently, without interference.&lt;br&gt;• [1] Handle always = Handle fax according to fax settings in the IP Profile for all offer-answer transactions (including the initial INVITE).&lt;br&gt;• [2] Handle on re-INVITE = Handle fax according to fax...</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>User's Manual 444 Document #: LTRT-41877</td>
<td></td>
</tr>
<tr>
<td><strong>Fax Offer Mode</strong>&lt;br&gt;sbc-fax-offer-mode&lt;br&gt;[IpProfile_SBCFaxOfferMode]</td>
<td>Defines the coders included in the outgoing SDP offer (sent to the called &quot;fax&quot;) for the SIP entity associated with the IP Profile. &lt;ul&gt;&lt;li&gt;[0] All coders = (Default) Use only (and all) the coders of the selected Coder Group configured using the SBCFaxCodersGroupID parameter.&lt;/li&gt;&lt;li&gt;[1] Single coder = Use only one coder. If a coder in the incoming offer (from the calling &quot;fax&quot;) matches a coder in the SBCFaxCodersGroupID, the device uses this coder. If no match exists, the device uses the first coder listed in the Coders Group ID (SBCFaxCodersGroupID).&lt;br&gt;<strong>Note:</strong> The parameter is applicable only if you set the IpProfile_SBCFaxBehavior parameter to a value other than [0].&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
<tr>
<td><strong>Fax Answer Mode</strong>&lt;br&gt;sbc-fax-answer-mode&lt;br&gt;[IpProfile_SBCFaxAnswerMode]</td>
<td>Defines the coders included in the outgoing SDP answer (sent to the calling &quot;fax&quot;) for the SIP entity associated with the IP Profile. &lt;ul&gt;&lt;li&gt;[0] All coders = Use matched coders between the incoming offer coders (from the calling &quot;fax&quot;) and the coders of the selected Coder Group (configured using the SBCFaxCodersGroupID parameter).&lt;/li&gt;&lt;li&gt;[1] Single coder = (Default) Use only one coder. If the incoming answer (from the called &quot;fax&quot;) includes a coder that matches a coder match between the incoming offer coders (from the calling &quot;fax&quot;) and the coders of the selected Coder Group (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 &quot;fax&quot;) and the coders of the selected Coder Group.&lt;br&gt;<strong>Note:</strong> The parameter is applicable only if you set the IpProfile_SBCFaxBehavior parameter to a value other than [0].&lt;/li&gt;&lt;/ul&gt;</td>
</tr>
<tr>
<td><strong>Remote Renegotiate on Fax Detection</strong>&lt;br&gt;sbc-rmt-renegotiate-on-fax-detect&lt;br&gt;[IpProfile_SBCRemoteRenegotiateOnFaxDetection]</td>
<td>Enables local handling of fax detection and negotiation by the device on behalf of the SIP entity associated with the IP Profile. This applies to faxes sent immediately upon the establishment of a voice channel (i.e., after 200 OK). The device attempts to detect the fax (CNG tone) from the originating SIP entity within a user-defined interval (see the SBCFaxDetectionTimeout parameter) immediately after the voice call is established. Once fax is detected, the device can handle the subsequent fax negotiation by sending re-INVITE messages to both SIP entities. The device also negotiates the fax coders between the two SIP entities. The negotiated coders are according to the list of fax coders.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Fax Coders Group</strong></td>
<td>assigned to each SIP entity, using the IP Profile parameter 'Fax Coders Group'.</td>
</tr>
<tr>
<td>0</td>
<td>Transparent = (Default) Device does not interfere in the fax transaction and assumes that the SIP entity fully supports fax renegotiation upon fax detection.</td>
</tr>
<tr>
<td>1</td>
<td>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.</td>
</tr>
<tr>
<td>2</td>
<td>No = The SIP entity does not support fax renegotiation upon fax detection when it is the originating or terminating fax.</td>
</tr>
<tr>
<td><strong>Fax Rerouting Mode</strong></td>
<td>Enables the rerouting of incoming SBC calls that are identified as fax calls to a new IP destination.</td>
</tr>
<tr>
<td>0</td>
<td>Disable (default)</td>
</tr>
<tr>
<td>1</td>
<td>Rerouting without delay</td>
</tr>
<tr>
<td><strong>Media</strong></td>
<td>Defines the device's handling of calls when RTP packets (media) are not received within a user-defined timeout interval (configured by the BrokenConnectionEventTimeout parameter). The interval can be during call setup (configured by the NoRTPDetectionTimeout parameter) or mid-call when RTP flow suddenly stops (configured by the BrokenConnectionEventTimeout parameter).</td>
</tr>
<tr>
<td>0</td>
<td>Ignore = The call is maintained despite no media and is released when signaling ends the call (i.e., SIP BYE).</td>
</tr>
<tr>
<td>1</td>
<td>Disconnect = (Default) The device ends the call.</td>
</tr>
<tr>
<td>2</td>
<td>Reroute = The device ends the call and searches the IP-to-IP Routing table for a matching rule and if found, generates a new INVITE to the corresponding destination (i.e., alternative routing). You can configure a routing rule whose matching characteristics is explicitly for calls with broken RTP connections. This is done using the Call Trigger parameter, as described in Configuring SBC IP-to-IP Routing Rules on page 501.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The device can only detect a broken RTP connection if silence compression is disabled for the RTP session.</td>
</tr>
<tr>
<td></td>
<td>If during a call the source IP address (from where the RTP packets are received by the device) is changed without notifying the device, the device rejects these RTP packets. To overcome this, configure the DisconnectOnBrokenConnection parameter to 0. By this configuration, the device doesn't detect RTP packets arriving from the original source IP address and switches (after 300 msec) to the RTP packets arriving from the new source IP address.</td>
</tr>
<tr>
<td></td>
<td>The corresponding global parameter is DisconnectOnBrokenConnection.</td>
</tr>
</tbody>
</table>

**Media IP Version Preference**

**media-ip-version-preference**

Defines the preferred RTP media IP addressing version for outgoing SIP calls (according to RFC 4091 and RFC 4092). The RFCs concern Alternative Network Address Types (ANAT) semantics in the SDP to offer groups of network addresses (IPv4 and IPv6) and the IP address version preference to establish the media stream. The IP address is indicated in the "c=" field (Connection) of the SDP.

- [0] Only IPv4 = (Default) SDP offer includes only IPv4 media IP addresses.
- [1] Only IPv6 = SDP offer includes only IPv6 media IP addresses.
- [2] Prefer IPv4 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first (preferred) media is IPv4.
- [3] Prefer IPv6 = SDP offer includes IPv4 and IPv6 media IP addresses, but the first (preferred) media is IPv6.

To indicate ANAT support, the device uses the SIP Allow header or to enforce ANAT it uses the Require header:

```plaintext
Require: sdp-anat
```

In the outgoing SDP, each 'm=' field is associated with an ANAT group. This is done using the 'a=mid:' and 'a=group:ANAT' fields. Each 'm=' field appears under a unique 'a=mid:' number, for example:

```plaintext
a=mid:1
m=audio 63288 RTP/AVP 0 8 18 101
c=IN IP6 3000::290:8fff:fe40:3e21
```

The 'a=group:ANAT' field shows the 'm=' fields belonging to it, using the number of the 'a=mid:' field. In addition, the ANAT group with the preferred 'm=' fields appears first. For example, the preferred group includes 'm=' fields under 'a=mid:1' and 'a=mid3':

```plaintext
a=group:ANAT 1 3
a=group:ANAT 2 4
```

If you configure the parameter to a "prefer" option, the outgoing SDP offer contains two medias which are the same except for the "c=" field. The first media is the preferred address type (and this type is also on the session...
Parameter | Description
---|---
level "c=" field), while the second media has its "c=" field with the other address type. Both medias are grouped by ANAT. For example, if the incoming SDP contains two medias, one secured and the other non-secured, the device sends the outgoing SDP with four medias:
- Two secured medias grouped in the first ANAT group, one with IPv4 and the other with IPv6. The first is the preferred type.
- Two non-secured medias grouped in the second ANAT group, one with IPv4 and the other with IPv6. The first is the preferred type.

Note:
- The parameter is applicable only when the device offers an SDP.
- The IP addressing version is determined according to the first SDP "m=" field.
- The feature is applicable to any type of media (e.g., audio and video) that has an IP address.
- The corresponding global parameter is MediaIPVersionPreference.

RTP Redundancy Depth
rtp-redundancy-depth
[IpProfile_RTPRedundancyDepth] Enables the device to generate RFC 2198 redundant packets. This can be used for packet loss where the missing information (audio) can be reconstructed at the receiver’s end from the redundant data that arrives in subsequent packets. This is required, for example, in wireless networks where a high percentage (up to 50%) of packet loss can be experienced.
- [0] 0 = (Default) Disable.
- [1] 1 = Enable - previous voice payload packet is added to current packet.

Note:
- When enabled, you can configure the payload type, using the RFC2198PayloadType parameter.
- The corresponding global parameter is RTPRedundancyDepth.

Gateway Answering Machine
AMD Sensitivity Parameter Suite
amd-sensitivity-parameter-suit
[IpProfile_AMDSensitivityParameterSuit] Defines the AMD Parameter Suite to use for the Answering Machine Detection (AMD) feature.
- [0] 0 = (Default) Parameter Suite 0 based on North American English with standard detection sensitivity resolution (8 sensitivity levels, from 0 to 7). This AMD Parameter Suite is provided by the AMD Sensitivity file, which is shipped pre-installed on the device.
- [1] 1 = Parameter Suite based 1 on North American English with high detection sensitivity resolution (16 sensitivity levels, from 0 to 15). This AMD Parameter Suite is provided by the AMD Sensitivity file, which is shipped pre-installed on the device.
- [2] 2 to [7]7 = Optional Parameter Suites that you can create based on any language (16 sensitivity levels,
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMD Sensitivity Level</td>
<td>Defines the AMD detection sensitivity level of the selected AMD Parameter Suite (using the 'AMD Sensitivity Parameter Suite' parameter).</td>
</tr>
<tr>
<td>amd-sensitivity-level</td>
<td>For Parameter Suite 0, the valid range is 0 to 7, where 0 is for best detection of an answering machine and 7 for best detection of a live call. For any Parameter Suite other than 0, the valid range is 0 to 15, where 0 is for best detection of an answering machine and 15 for best detection of a live call.</td>
</tr>
<tr>
<td>[IpProfile_AMDSensitivityLevel]</td>
<td>Note: The corresponding global parameter is AMDSensitivityLevel.</td>
</tr>
<tr>
<td>AMD Max Greeting Time</td>
<td>Defines the maximum duration (in 5-msec units) that the device can take to detect a greeting message.</td>
</tr>
<tr>
<td>amd-max-greeting-time</td>
<td>The valid range value is 0 to 51132767. The default is 300.</td>
</tr>
<tr>
<td>[IpProfile_AMDMaxGreetingTime]</td>
<td>Note: The corresponding global parameter is AMDMaxGreetingTime.</td>
</tr>
<tr>
<td>AMD Max Post Silence Greeting Time</td>
<td>Defines the maximum duration of silence from after the greeting time is over (configured by AMDMaxGreetingTime) until the device's AMD decision.</td>
</tr>
<tr>
<td>amd-max-post-silence-greeting-time</td>
<td>Note: The corresponding global parameter is AMDMaxPostGreetingSilenceTime.</td>
</tr>
<tr>
<td>[IpProfile_AMDMaxPostSilenceGreetingTime]</td>
<td></td>
</tr>
<tr>
<td>Local Tones</td>
<td></td>
</tr>
<tr>
<td>Local RingBack Tone Index</td>
<td>Defines the ringback tone that you want to play from the PRT file.</td>
</tr>
<tr>
<td>local-ringback-tone-index</td>
<td>To associate a user-defined tone, configure the parameter with the tone's index number (0-79) as appears in the PRT file. By default (value of -1), the device plays a default ringback tone.</td>
</tr>
<tr>
<td>[IPProfile_LocalRingbackTone]</td>
<td>To play user-defined tones, you need to record your tones and then install them on the device using a loadable Prerecorded Tones (PRT) file. For more information, see Prerecorded Tones File on page 660.</td>
</tr>
<tr>
<td>Local Held Tone Index</td>
<td>Defines the held tone that you want to play from the PRT file.</td>
</tr>
<tr>
<td>local-held-tone-index</td>
<td>To associate a user-defined tone, configure the parameter with the tone's index number (0-79) as appears in the PRT file. By default (value of -1), the device plays a default held tone.</td>
</tr>
<tr>
<td>[IPProfile_LocalHeldTone]</td>
<td>To play user-defined tones, you need to record your tones</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td>and then install them on the device using a loadable Prerecorded Tones (PRT) file. For more information, see Prerecorded Tones File on page 660.</td>
</tr>
</tbody>
</table>
This page is intentionally left blank.
23 SBC Overview

This section provides an overview of the device’s SBC application.

Note:

- For guidelines on how to deploy your SBC device, refer to the SBC Design Guide document.
- The SBC feature is available only if the device is installed with a License Key that includes this feature. For installing a License Key, see "License Key" on page 669.
- 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 965.

23.1 Feature List

The SBC application supports the following main features:

- **NAT traversal:** The device supports NAT traversal, allowing, for example, communication with ITSPs with globally unique IP addresses and with far-end users located behind NAT on the WAN. The device supports this by:
  - Continually registering far-end users with its users registration database.
  - Maintaining remote NAT binding state by frequent registrations and thereby, off-loading far-end registrations from the LAN IP PBX.
  - Using Symmetric RTP (RFC 4961) to overcome bearer NAT traversal.

- **VoIP firewall and security for signaling and media:**
  - **SIP signaling:**
    ♦ Deep and stateful inspection of all SIP signaling packets.
    ♦ SIP dialog initiations may be rejected based on values of incoming SIP INVITE message and other Layer-3 characteristics.
    ♦ Packets not belonging to an authorized SIP dialog are discarded.
  - **RTP:**
    ♦ Opening pinholes (ports) in the device’s firewall based on SDP offer-answer negotiations.
    ♦ Deep packet inspection of all RTP packets.
    ♦ Late rogue detection - if a SIP session was gracefully terminated and someone tries to "ride on it" with rogue traffic from the already terminated RTP and SIP context, the VoIP Firewall prevents this from occurring.
    ♦ Disconnects call (after user-defined time) if RTP connection is broken.
    ♦ Black/White lists for both Layer-3 firewall and SIP classification.

- **Stateful Proxy Operation Mode:** The device can act as a Stateful Proxy by enabling SIP messages to traverse it transparently (with minimal interference) between the inbound and outbound legs.

- **B2BUA and Topology Hiding:** The device intrinsically supports topology hiding, limiting the amount of topology information displayed to external parties. For example, IP addresses of ITSPs’ equipment (e.g. proxies, gateways, and application servers) can be hidden from outside parties. The device's topology hiding is provided by implementing back-to-back user agent (B2BUA) leg routing:
- Strips all incoming SIP Via header fields and creates a new Via value for the outgoing message.
- Each leg has its own Route/Record Route set.
- User-defined manipulation of SIP To, From, and Request-URI host names.
- Generates a new SIP Call-ID header value (different between legs).
- Changes the SIP Contact header and sets it to the device’s address.
- Layer-3 topology hiding by modifying source IP address in the SIP IP header.

SIP normalization: The device supports SIP normalization, whereby the SBC application can overcome interoperability problems between SIP user agents. This is achieved by the following:
- Manipulation of SIP URI user and host parts.
- Connection to ITSP SIP trunks on behalf of an IP-PBX - the device can register and utilize user and password to authenticate for the IP-PBX.

Survivability:
- Routing calls to alternative routes such as the PSTN.
- Routing calls between user agents in the local network using a dynamic database (built according to registrations of SIP user agents).

Routing:
- IP-to-IP routing translations of SIP, UDP, TCP, TLS (when extensive transcoding is not required).
- Load balancing and redundancy of SIP servers.
- Routing according to Request-URI\Specific IP address\Proxy\FQDN.
- Alternative routing.
- Routing between different Layer-3 networks (e.g., LAN and WAN).

Load balancing/redundancy of SIP servers.
- ITSP accounts.
- SIP URI user and host name manipulations.
- Coder transcoding.

23.2 B2BUA and Stateful Proxy Operating Modes

The device can operate in one or both of the following SBC modes:

- **Back-to-Back User Agent (B2BUA):** Maintains independent sessions toward the endpoints, processing an incoming request as a user agent server (UAS) on the inbound leg, and processing the outgoing request as a user agent client (UAC) on the outbound leg. SIP messages are modified regarding headers between the legs and all the device’s interworking features may be applied.

- **Stateful Proxy Server:** SIP messages traverse the device transparently (with minimal interference) between the inbound and outbound legs, for connecting SIP endpoints.

By default, the device’s B2BUA mode changes SIP dialog identifiers and topology data in SIP messages traversing through it:

- **Call identifiers:** Replaces the From-header tag and Call-ID header so that they are different for each leg (inbound and outbound).

- **Routing headers:**
  - Removes all Via headers in incoming requests and sends the outgoing message with its own Via header.
  - Doesn't forward any Record-Route headers from the inbound to outbound leg, and vice versa.
23. SBC Overview

- Replaces the address of the Contact header in the incoming message with its own address in the outgoing message.
- Replaces the User-Agent/Server header value in the outgoing message, and replaces the original value with itself in the incoming message.

In contrast, when the device operates in Stateful Proxy mode, the device by default forwards SIP messages transparently (unchanged) between SIP endpoints (from inbound to outbound legs). The device retains the SIP dialog identifiers and topology headers received in the incoming message and sends them as is in the outgoing message. The device handles the above mentioned headers transparently (i.e., they remain unchanged) or according to configuration (enabling partial transparency), and only adds itself as the top-most Via header and optionally, to the Record-Route list. To configure the handling of these headers for partial transparency, use the following IP Profile parameters (see "Configuring IP Profiles" on page 417):

- IpProfile_SBCRemoteRepresentationMode: Contact and Record-Route headers
- IpProfile_SBCKeepViaHeaders: Via headers
- IpProfile_SBCKeepUserAgentHeader: User-Agent headers
- IpProfile_SBCKeepRoutingHeaders: Record-Route headers
- IpProfile_SBCRemoteMultipleEarlyDialogs: To-header tags

Thus, the Stateful Proxy mode provides full SIP transparency (no topology hiding) or asymmetric topology hiding. Below is an example of a SIP dialog-initiating request when operating in Stateful Proxy mode for full transparency, showing all the incoming SIP headers retained in the outgoing INVITE message.

Figure 23-1: Example of SIP Message Handling in Stateful Proxy Mode

<table>
<thead>
<tr>
<th>Incoming INVITE</th>
<th>Outgoing INVITE</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVITE sip:<a href="mailto:bob@domain.com">bob@domain.com</a> SIP/2.0</td>
<td>INVITE sip:<a href="mailto:bob@domain.com">bob@domain.com</a> SIP/2.0</td>
</tr>
<tr>
<td>To: Bob <a href="">sip:bob@domain.com</a></td>
<td>To: Bob <a href="">sip:bob@domain.com</a></td>
</tr>
<tr>
<td>From: Alice</td>
<td>From: Alice</td>
</tr>
<tr>
<td><a href="">sip:alice@caller.com</a>;tag=100</td>
<td><a href="">sip:alice@caller.com</a>;tag=100</td>
</tr>
<tr>
<td>Call-ID: <a href="mailto:callid1@caller.com">callid1@caller.com</a></td>
<td>Call-ID: <a href="mailto:callid1@caller.com">callid1@caller.com</a></td>
</tr>
<tr>
<td>Contact: <a href="">sip:alice@pcl.caller.com</a></td>
<td>Contact: <a href="">sip:alice@pcl.caller.com</a></td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP pc1.com;branch=branchn2</td>
<td>Via: SIP/2.0/UDP pc1.com;branch=branchn3</td>
</tr>
<tr>
<td>Via: SIP/2.0/UDP pc1.com;branch=branchn2</td>
<td>Via: SIP/2.0/UDP pc1.com;branch=branchn1</td>
</tr>
<tr>
<td>Record-Route: &lt;pc2.com;lr&gt;</td>
<td>Record-Route: &lt;Proxy-IP;lr&gt;</td>
</tr>
<tr>
<td>Record-Route: &lt;pc1.com;lr&gt;</td>
<td>Record-Route: &lt;pc2.com;lr&gt;</td>
</tr>
<tr>
<td>CSeq: 666 INVITE</td>
<td>CSeq: 666 INVITE</td>
</tr>
<tr>
<td>User-Agent: IPFv3.1</td>
<td>User-Agent: IPFv3.1</td>
</tr>
<tr>
<td>Max-Forwards: 70</td>
<td>Max-Forwards: 70</td>
</tr>
<tr>
<td>Content-Type: application/sdp</td>
<td>Content-Type: application/sdp</td>
</tr>
<tr>
<td>Content-Length: 142</td>
<td>Content-Length: 142</td>
</tr>
<tr>
<td>v=0</td>
<td>v=0</td>
</tr>
</tbody>
</table>

Some of the reasons for implementing Stateful Proxy mode include:

- B2BUA typically hides certain SIP headers for topology hiding. In specific setups, some SIP servers require the inclusion of these headers to know the history of the SIP request. In such setups, the requirement may be asymmetric topology hiding, whereby SIP traffic toward the SIP server must expose these headers whereas SIP traffic toward the users must not expose these headers.
- B2BUA changes the call identifiers between the inbound and outbound SBC legs and therefore, call parties may indicate call identifiers that are not relayed to the other leg. Some SIP functionalities are achieved by conveying the SIP call identifiers either in SIP specific headers (e.g., Replaces) or in the message bodies (e.g. Dialog Info in an XML body).
In some setups, the SIP client authenticates using a hash that is performed on one or more of the headers that B2BUA changes (removes). Therefore, implementing B2BUA would cause authentication to fail.

For facilitating debugging procedures, some administrators require that the value in the Call-ID header remains unchanged between the inbound and outbound SBC legs. As B2BUA changes the Call-ID header, such debugging requirements would fail.

The operating mode can be configured per the following configuration entities:

- SRDs in the SRDs table (see "Configuring SRDs" on page 327)
- IP Groups in the IP Groups table (see "Configuring IP Groups" on page 345)

If the operation mode is configured in both tables, the operation mode of the IP Group is applied. Once configured, the device uses default settings in the IP Profiles table for handling the SIP headers, as mentioned previously. However, you can change the default settings to enable partial transparency.

**Note:**

- The To-header tag remains the same for inbound and outbound legs of the dialog, regardless of operation mode.
- If the Operation Mode of the SRD/IP Group of one leg of the dialog is set to 'Call Stateful Proxy', the device also operates in this mode on the other leg with regards to the dialog identifiers (Call-ID header, tags, CSeq header).
- It is recommended to implement the B2BUA mode, unless one of the reasons mentioned previously is required. B2BUA supports all the device's feature-rich offerings, while Stateful Proxy may offer only limited support. The following features are not supported when in Stateful Proxy mode:
  - Alternative routing
  - Call forking
  - Terminating REFER/3xx
- If Stateful Proxy mode is enabled and any one of the unsupported features is enabled, the device disables the Stateful Proxy mode and operates in B2BUA mode.
- You can configure the device to operate in both B2BUA and Stateful Proxy modes for the same users. This is typically implemented when users need to communicate with different SIP entities (IP Groups). For example, B2BUA mode for calls destined to a SIP Trunk and Stateful Proxy mode for calls destined to an IP PBX. The configuration is done using IP Groups and SRDs.
- If Stateful Proxy mode is used only due to the debugging benefits, it is recommended to configure the device to only forward the Call-ID header unchanged.
23.3 Call Processing of SIP Dialog Requests

The device processes incoming SIP dialog requests (SIP methods) such as INVITE, SUBSCRIBE, OPTIONS, REFER, INFO, UNSOLICITED NOTIFY, MESSAGE, and REGISTER. The process is summarized in the following figure and subsequently described:

![SBC Call Processing Diagram]

The SIP dialog-initiating process consists of the following stages:

1. **Determining Source and Destination URL**: The SIP protocol has more than one URL in a dialog-initiating request that may represent the source and destination URLs. The device obtains the source and destination URLs from certain SIP headers. Once the URLs are determined, the user and host parts of the URLs can be used as matching rule characteristics for classification, message manipulation, and call routing.
2. **Determining SIP Interface:** The device checks the SIP Interface on which the SIP dialog is received. The SIP Interface defines the local SIP "listening" port and IP network interface. For more information, see "Configuring SIP Interfaces" on page 337.

3. **Applying SIP Message Manipulation:** Depending on configuration, the device can apply a SIP message manipulation rule (assigned to the SIP Interface) on the incoming SIP message. A SIP Message Manipulation rule defines a matching characteristics \( (\text{condition}) \) of the incoming SIP message and the corresponding manipulation operation (e.g., remove the P-Asserted-Identity header), which can apply to almost any aspect of the message (add, remove or modify SIP headers and parameters). For more information, see "Configuring SIP Message Manipulation" on page 393.

4. **Classifying to an IP Group:** Classification identifies the incoming SIP dialog request as belonging to a specific IP Group (i.e., from where the SIP dialog request originated). The classification process is based on the SRD to which the dialog belongs (the SRD is determined according to the SIP Interface). For more information, see "Configuring Classification Rules" on page 493.

5. **Applying Inbound Manipulation:** Depending on configuration, the device can apply an Inbound Manipulation rule to the incoming dialog. This manipulates the user part of the SIP URI for source (e.g., in the SIP From header) and destination (e.g., in the Request-URI line). The manipulation rule is associated with the incoming dialog, by configuring the rule with incoming matching characteristics such as source IP Group and destination host name. The manipulation rules are also assigned a Routing Policy, which in turn, is assigned to IP-to-IP routing rules. As most deployments require only one Routing Policy, the default Routing Policy is automatically assigned to manipulation and routing rules. For more information, see "Configuring IP-to-IP Inbound Manipulations" on page 531.

6. **SBC IP-to-IP Routing:** The device searches the IP-to-IP Routing table for a routing rule that matches the characteristics of the incoming call. If found, the device routes the call to the configured destination which can be, for example, an IP Group, the Request-URI if the user is registered with the device, and a specified IP address. For more information, see "Configuring SBC IP-to-IP Routing Rules" on page 501.

7. **Applying Inbound SIP Message Manipulation:** Depending on configuration, the device can apply a SIP message manipulation rule (assigned to the IP Group) on the incoming dialog. For more information, see Stage 3.

8. **Applying Outbound Manipulation:** Depending on configuration, the device can apply an Outbound Manipulation rule to the outbound dialog. This manipulates the user part of the Request-URI for source (e.g., in the SIP From header) or destination (e.g., in the SIP To header) or calling name in the outbound SIP dialog. The
manipulation rule is associated with the dialog, by configuring the rule with incoming matching characteristics such as source IP Group and destination host name. The manipulation rules are also assigned a Routing Policy, which in turn, is assigned to IP-to-IP routing rules. As most deployments require only one Routing Policy, the default Routing Policy is automatically assigned to manipulation rules and routing rules. For more information, see "Configuring IP-to-IP Outbound Manipulations" on page 535.

9. **Applying Outbound SIP Message Manipulation:** Depending on configuration, the device can apply a SIP message manipulation rule (assigned to the IP Group) on the outbound dialog. For more information, see Stage 3.

10. The call is sent to the configured destination.

23.4 **User Registration**

The device provides a registration database for registering users. Only users belonging to a User-type IP Group can register with the device. User-type IP Groups represent a group of SIP user agents that share the following characteristics:

- Perform registrations and share the same serving proxy/registrar
- Same SIP and media behavior
- Same IP Profile
- Same SIP handling configuration
- Same Call Admission Control (CAC)

Typically, the device is configured as the user's outbound proxy, routing requests (using the IP-to-IP Routing table) from the user's User-type IP Group to the serving proxy, and vice versa. Survivability can be achieved using the alternative routing feature.

The device forwards registration requests (REGISTER messages) from a Server-type IP Group, but does not save the registration binding in its registration database.

23.4.1 **Initial Registration Request Processing**

A summary of the device's handling of registration requests (REGISTER messages) is as follows:

- The URL in the To header of the REGISTER message constitutes the primary Address of Record (AOR) for registration (according to standard). The device can save other AORs in its registration database as well. When the device searches for a user in its registration database, any of the user's AORs can result in a match.
- The device's Classification process for initial REGISTER messages is slightly different than for other SIP messages. Unlike other requests, initial REGISTER requests can't be classified according to the registration database.
- If registration succeeds (replied with 200 OK by the destination server), the device adds a record to its registration database, which identifies the specific contact of the specific user (AOR). The device uses this record to route subsequent SIP requests to the specific user (in normal or survivability modes).
- The records in the device's registration database include the Contact header. The device adds every REGISTER request to the registration database before manipulation, allowing correct user identification in the Classification process for the next received request.
- You can configure Call Admission Control (CAC) rules for incoming and outgoing REGISTER messages. For example, you can limit REGISTER requests from a specific IP Group or SRD. Note that this applies only to concurrent REGISTER dialogs and not concurrent registrations in the device's registration database.
The device provides a dynamic registration database that it updates according to registration requests traversing it. Each database entry for a user represents a binding between an AOR (obtained from the SIP To header), optional additional AORs, and one or more contacts (obtained from the SIP Contact headers). Database bindings are added upon successful registration responses from the proxy server (SIP 200 OK). The device removes database bindings in the following cases:

- Successful de-registration responses (REGISTER with Expires header that equals zero).
- Registration failure responses.
- Timeout of the Expires header value (in scenarios where the UA did not send a refresh registration request).

Note:

- The same contact cannot belong to more than one AOR.
- Contacts with identical URIs and different ports and transport types are not supported (same key is created).
- Multiple contacts in a single REGISTER message is not supported.
- One database is shared between all User-type IP Groups.

23.4.2 Classification and Routing of Registered Users

The device can classify incoming SIP dialog requests (e.g., INVITE) from registered users to an IP Group, by searching for the sender's details in the registration database. The device uses the AOR from the From header and the URL in the Contact header of the request to locate a matching registration binding. The found registration binding contains information regarding the registered user, including the IP Group to which it belongs. (Upon initial registration, the Classification table is used to classify the user to a User-type IP Group and this information is then added with the user in the registration database.)

The destination of a dialog request can be a registered user and the device thus uses its registration database to route the call. This can be achieved by various ways such as configuring a rule in the IP-to-IP Routing table where the destination is a User-type IP Group or any matching user registered in the database ('Destination Type' is configured to All Users). The device searches the registration database for a user that matches the incoming Request-URI (listed in chronological order):

1. Unique Contact generated by the device and sent in the initial registration request to the serving proxy.
2. AOR. The AOR is originally obtained from the incoming REGISTER request and must either match both user part and host part of the Request-URI, or only user part.
3. Contact. The Contact is originally obtained from the incoming REGISTER request.

If registrations are destined to the database (using the above rules), the device does not attempt to find a database match, but instead replies with a SIP 200 OK (used for Survivability). Once a match is found, the request is routed either to the contact received in the initial registration or (if the device identifies that the user agent is behind a NAT) to the source IP address of the initial registration.

You can configure (using the SBCDBRoutingSearchMode parameter) for which part of the destination Request-URI in the INVITE message the device must search in the registration database:

- Only by entire Request-URI (user@host), for example, "4709@joe.company.com".
- By entire Request-URI, but if not found, by the user part of the Request-URI, for example, "4709".
When an incoming INVITE is received for routing to a user and the user is located in the registration database, the device sends the call to the user's corresponding contact address specified in the database.

**Note:** If the Request-URI contains the "tel:" URI or "user=phone" parameter, the device searches only for the user part.

### 23.4.3 General Registration Request Processing

The device's general handling of registration requests (REGISTER messages) for unregistered users is as follows:

- The device routes REGISTER requests according to the IP-to-IP Routing table. If the destination is a User-type IP Group, the device does not forward the registration; instead, it accepts (replies with a SIP 200 OK response) or rejects (replies with a SIP 4xx) the request according to the user's IP Group configuration.
- Alternative routing can be configured for REGISTER requests, in the IP-to-IP Routing table.
- By default, the Expires header has the same value in incoming and outgoing REGISTER messages. However, you can modify the Expires value using the following parameters: SBCUserRegistrationTime, SBCProxyRegistrationTime, SBCRandomizeExpires, and SBCSurvivabilityRegistrationTime. You can also modify the Expires value of REGISTER requests received from users located behind NAT, using the IP Profile parameters IpProfile_SBCUserBehindUdpNATRegistrationTime and IpProfile_SBCUserBehindTcpNATRegistrationTime.
- By default, the Contact header in outgoing REGISTER message is different than the Contact header in the incoming REGISTER. The user part of the Contact is populated with a unique contact generated by the device and associated with the specific registration. The IP address in the host part is changed to the address of the device. Alternatively, the original user can be retained in the Contact header and used in the outgoing REGISTER request (using the SBCKeepContactUserinRegister parameter).

### 23.4.4 Registration Refreshes

Registration refreshes are incoming REGISTER requests from users that are registered in the device's registration database. The device sends these refreshes to the serving proxy only if the serving proxy's Expires time is about to expire; otherwise, the device responds with a 200 OK to the user, without routing the REGISTER. Each such refreshes also refresh the internal timer set on the device for this specific registration.

The device automatically notifies SIP proxy / registrar servers of users that are registered in its registration database and whose registration timeout has expired. When a user's registration timer expires, the device removes the user's record from the database and sends an un-register notification (REGISTER message with the Expires header set to 0) to the proxy/registrar. This occurs only if a REGISTER message is sent to an IP Group destination type (in the IP-to-IP Routing table).

You can also apply a graceful period to unregistered requests, using the 'User Registration Grace Time' parameter (SBCUserRegistrationGraceTime):

- You can configure the device to add extra time (grace period) to the expiration timer of registered users in the database. If you configure this grace period, the device keeps the user in the database (and does not send an un-register to the registrar server), allowing the user to send a "late" re-registration to the device. The device removes the
user from the database only when this additional time expires.

- The graceful period is also used before removing a user from the registration database when the device receives a successful unregister response (200 OK) from the registrar/proxy server. This is useful in scenarios, for example, in which users (SIP user agents) such as IP Phones erroneously send unregister requests. Instead of immediately removing the user from the registration database upon receipt of a successful unregister response, the device waits until it receives a successful unregister response from the registrar server, waits the user-defined graceful time and if no register refresh request is received from the user agent, removes the contact (or AOR) from the database.

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

### 23.4.5 Registration Restriction Control

The device provides flexibility in controlling user registrations:

- **Limiting Number of Registrations:** You can limit the number of users that can register with the device per IP Group, SIP Interface, and/or SRD, in the IP Group, SIP Interface and SRDs tables respectively. By default, no limitation exists.

- **Blocking Incoming Calls from Unregistered Users:** You can block incoming calls (INVITE requests) from unregistered users belonging to User-type IP Groups. By default, calls from unregistered users are not blocked. This is configured per SIP Interface or SRD. When the call is rejected, the device sends a SIP 500 (Server Internal Error) response to the remote end.

### 23.4.6 Deleting Registered Users

You can remove registered users from the device's registration database through CLI:

- To delete a specific registered user:
  
  ```
  # clear voip register db sbc user <AOR of user – user part or user@host>
  ```

  For example:

  ```
  # clear voip register db sbc user John@10.33.2.22
  # clear voip register db sbc user John
  ```

- To delete all registered users belonging to a specific IP Group:

  ```
  # clear voip register db sbc ip-group <ID or name>
  ```

### 23.5 Media Handling

Media behavior includes anything related to the establishment, management and termination of media sessions within the SIP protocol. Media sessions are created using the SIP offer-answer mechanism. If successful, the result is a bi-directional media (RTP) flow (e.g. audio, fax, modem, DTMF). Each offer-answer may create multiple media sessions of different types (e.g. audio and fax). In a SIP dialog, multiple offer-answer transactions may occur and each may change the media session characteristics (e.g. IP address, port, coders, media types, and RTP mode). The media capabilities exchanged in an offer-answer transaction include the following:

- Media types (e.g., audio, secure audio, video, fax, and text)
- IP addresses and ports of the media flow
23. SBC Overview

- Media flow mode (send receive, receive only, send only, inactive)
- Media coders (coders and their characteristics used in each media flow)
- Other (standard or proprietary) media and session characteristics

Typically, the device does not change the negotiated media capabilities (mainly performed by the remote user agents). However, it does examine and may take an active role in the SDP offer-answer mechanism. This is done mainly to anchor the media to the device (default) and also to change the negotiated media type, if configured. Some of the media handling features, which are described later in this section, include the following:

- Media anchoring (default)
- Direct media
- Audio coders restrictions
- Audio coders transcoding
- RTP-SRTP transcoding
- DTMF translations
- Fax translations and detection
- Early media and ringback tone handling
- Call hold translations and held tone generation
- NAT traversal
- RTP broken connections
- Media firewall

  - RTP pin holes - only RTP packets related to a successful offer-answer negotiation traverse the device: When the device initializes, there are no RTP pin holes opened. This means that each RTP/RTCP packets destined to the device are discarded. Once an offer-answer transaction ends successfully, an RTP pin hole is opened and RTP/RTCP flows between the two remote user agents. Once a pin hole is opened, the payload type and RTP header version is validated for each packet. RTP pin holes close if one of the associated SIP dialogs is closed (may also be due to broken connection).

  - Late rogue detection - once a dialog is disconnected, the related pin holes also disconnect.

  - Deep Packet inspection of the RTP that flows through the opened pin holes.

23.5.1 Media Anchoring

By default, the device anchors the media (RTP) traffic. In other words, the media between SIP endpoints traverses the device. You can change this default mode by enabling direct media between SIP endpoints. Media anchoring may be required, for example, to resolve NAT problems, enforce media security policies, perform media transcoding, and media monitoring.

To enforce RTP traffic to flow through the device, the device modifies all IP address fields in the SDP:

- Origin: IP address, session and version id
- Session connection attribute ('c=' field)
- Media connection attribute ('c=' field)
- Media port number
- RTCP media attribute IP address and port

The device uses different local ports (e.g., for RTP, RTCP and fax) for each leg (inbound and outbound). The local ports are allocated from the Media Realm associated with each
The Media Realm assigned to the leg's IP Group (in the IP Groups table) is used. If not assigned to the IP Group, the Media Realm assigned to the leg's SIP Interface (in the SIP Interfaces table) is used. The following figure provides an example of SDP handling for a call between a LAN IP Phone 10.2.2.6 and a remote IP Phone 212.179.1.13 on the WAN.

**Figure 23-3: SDP Offer/Answer Example**

### 23.5.2 Direct Media

You can configure the device to allow the media (RTP/SRTP) session to flow directly between the SIP endpoints, without traversing the device. This is referred to as No Media Anchoring (also known as Anti-Tromboning or Direct Media). SIP signaling continues to traverse the device, with minimal intermediation and involvement, to enable certain SBC capabilities such as routing. By default, the device employs media anchoring, whereby the media session traverses the device, as described in "Media Anchoring" on page 463.

Direct media offers the following benefits:

- Saves network bandwidth
- Reduces the device's CPU usage (as there is no media handling)
- Avoids interference in SDP negotiation and header manipulation on RTP/SRTP
Direct media is typically implemented for calls between users located in the same LAN or domain, and where NAT traversal is not required and other media handling features such as media transcoding is not required. The following figure provides an example of direct media between LAN IP phones, while SIP signaling continues to traverse the device between LAN IP phones and the hosted WAN IP-PBX.

Figure 23-4: Direct Media where only Signaling Traverses Device

- To enable direct media:
  - For all calls: Use the global parameter, SBCDirectMedia (overrides all other direct media configuration).
  - For specific calls:
    - SIP Interface: You can enable direct media per SIP Interface (in the SIP Interfaces table), whereby calls (source and destination) associated with this same SIP Interface are handled as direct media calls. The SIP Interface can also enable direct media for users located behind the same NAT. For more information, see "Configuring SIP Interfaces" on page 337.
    - Direct Media Tag: You can enable direct media between users that are configured with the same Direct Media tag value. The tag is configured using the IP Profiles table’s IPProfile_SBCDirectMediaTag parameter (see "Configuring IP Profiles" on page 417).

The device employs direct media between endpoints under the following configuration conditions (listed in chronological order):

1. Direct media is enabled by the global parameter (SBCDirectMedia).
2. IP Groups of the endpoints are associated with IP Profiles whose 'Direct Media Tag' parameter has the same value (non-empty value).
3. IP Groups of the endpoints have the 'SBC Operation Mode' parameter set to Microsoft Server (direct media is required in the Skype for Business environment). For more information, see "Configuring IP Groups" on page 345.
4. IP Groups of the endpoints use the same SIP Interface and the SIP Interface's 'SBC
Direct Media’ parameter is set to **Enable** (SIPInterface_SBCDirectMedia = 1).

5. IP Groups of the endpoints use the same SIP Interface and the SIP Interface’s ‘SBC Direct Media’ parameter is set to Enable When Single NAT (SIPInterface_SBCDirectMedia = 2), and the endpoints are located behind the same NAT.

**Note:**

- If you enable direct media by the SBCDirectMedia parameter, direct media is applied to all calls even if direct media is disabled per SIP Interface.
- If you configure direct media for all calls (using the SBCDirectMedia parameter), the device does not open voice channels nor allocate media ports for the calls, as the media always bypasses the device. In contrast, if you configure direct media for specific calls, the device allocates ports for these calls. The reason is that the ports may be required for mid-call services (e.g., early media, call forwarding, call transfer, and playing on-hold tones) handled by the server (IP PBX), which traverse the device. Therefore, make sure that you have allocated sufficient media ports (Media Realm) for such calls.
- Direct media cannot operate with the following features:
  - Manipulation of SDP data (offer-answer transaction) such as ports, IP address, coders
  - Force transcoding
  - Extension Coders
  - Extension of RFC 2833 / out-of-band DTMF / in-band DTMF
  - Extension of SRTP/RTP
- All restriction features (Allowed Coders, restrict SRTP/RTP, restrict RFC 2833) can operate with direct media. Restricted coders are removed from the SDP offer message.
- For two users belonging to the same SIP Interface that is enabled for direct media and one of the users is defined as a foreign user (example, “follow me service”) located in the WAN while the other is located in the LAN: calls between these two users cannot be established until direct media is disabled for the SIP Interface. The reason for this is that the device does not interfere in the SIP signaling. In other words, parameters such as IP addresses are not manipulated for calls between LAN and WAN (although required).

### 23.5.3 Restricting Audio Coders

You can configure a list of permitted (allowed) voice coders that can be used for a specific SIP entity (leg). In other words, you can enforce the use of specific coders. If the SDP offer in the incoming SIP message does not contain any coder that is configured as an allowed coder, the device rejects the calls (unless transcoding is implemented whereby Extension coders are added to the SDP, as described in Coder Transcoding on page 467). If the SDP offer contains some coders that are configured as allowed coders, the device manipulates the SDP offer by removing the coders that are not configured as allowed coders, before routing the SIP message to its destination. The device also re-orders (prioritizes) the coder list in the SDP according to the listed order of configured allowed coders.

For example, assume the following:

- The SDP offer in the incoming SIP message contains the G.729, G.711, and G.723 coders.
- The allowed coders configured for the SIP entity include G.711 and G.729.
The device removes the G.723 coder from the SDP offer, re-orders the coder list so that G.711 is listed first, and sends the SIP message containing only the G.711 and G.729 coders in the SDP.

The allowed coders are configured in the Allowed Audio Coders Groups table. For more information, see "Configuring Allowed Audio Coder Groups" on page 412.

**Note:** If you assign the SIP entity an Allowed Audio Coders Group for coder restriction and a Coders Group for extension coders (i.e., voice transcoding), the allowed coders take precedence over the extension coders. In other words, if an extension coder is not listed as an allowed coder, the device does not add the extension coder to the SDP offer.

### 23.5.4 Coder Transcoding

By default, the device forwards media packets transparently (i.e., no media negotiation) between SIP endpoints. However, when there are no common coders between two SIP entities that need to establish voice communication (i.e., the SDP answer from one SIP entity doesn't include any coder included in the SDP offer previously sent by the other), you can configure the device to perform audio coder transcoding between the inbound and outbound legs in order to enable media flow between them.

Transcoding may also be performed in scenarios where the same coder has been chosen between the legs, but where coder transrating is required. For example, the coders may use different coder settings such as rate and packetization time (G.729 at 20 ms to G.729 at 30 ms).

The coders that the device adds to the SDP offer on the outbound leg is referred to as extension coders. The extension coders are configured using Coder Groups (see "Configuring Coder Groups" on page 407), which you need to then assign to the IP Profile associated with the SIP entity.

The figure below illustrates transcoding between two SIP entities (IP Groups) where one uses G.711 (LAN IP phone) and the other G.729 (WAN IP phone). The initial SDP offer received on the inbound leg from the LAN IP phone includes coder G.711 as the supported coder. In the outgoing SDP offer on the outbound leg to the WAN IP phone, the device adds extension coder G.729 to the SDP, which is supported by the WAN IP phone. The subsequent incoming SDP answer from the WAN IP phone includes the G.729 coder as the chosen coder. Since this coder was not included in the original incoming SDP offer from the LAN IP phone, the device performs G.729-G.711 transcoding between the inbound and outbound legs.
Note:

- If you assign a SIP entity an Allowed Audio Coders Group for coder restriction (allowed coders) and a Coders Group for extension coders, the allowed coders take precedence over the extension coders. In other words, if an extension coder is not listed as an allowed coder, the device does not add the extension coder to the SDP offer.

- If none of the coders in the incoming SDP offer on the inbound leg appear in the associated Allowed Audio Coders Group for coder restriction, the device rejects the call (sends a SIP 488 to the SIP entity that initiated the SDP offer).

- If none of the coders (including extension coders) in the outgoing SDP offer on the outbound leg appear in the associated Allowed Audio Coders Group for coder restriction, the device rejects the call (sends a SIP 488 to the SIP entity that initiated the SDP offer).

- For coder transcoding, the following prerequisites must be met (otherwise, the extension coders are not added to the SDP offer):
  - The device must support at least one of the coders listed in the incoming SDP offer.
  - The device must have available DSPs for both legs (inbound and outbound).
  - The incoming SDP offer must have at least one media line that is audio ('m=audio').

- The device adds the extension coders below the coder list received in the original SDP offer. This increases the chance of media flow without requiring transcoding.

- The device does not add extension coders that also appear in the original SDP offer.

As an example for using allowed and extension coders, assume the following:

**Inbound leg:**

- Incoming SDP offer includes the G.729, G.711, and G.723 coders.

```
m=audio 6050 RTP/AVP 18 0 8 4 96
a=rtpmap:18 G729/8000
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:4 G723/8000
a=fmtp:4 annexa=no
a=rtpmap:96 telephone-event/8000
a=fmtp:96 0-15
a=ptime:20
a=sendrecv
```

The SDP "m=audio 6010 RTP/AVP 18 0 8 4 96" line shows the coder priority, where "18" (G.729) is highest and "4" (G.723) is lowest.

- Allowed Audio Coders Group for coder restriction includes the G.711 and G.729 coders (listed in order of appearance).

**Outbound leg:**

- Allowed Audio Coders Group for coder restriction includes the G.723, G.726, and G.729 coders (listed in order of appearance).

- Allowed Audio Coders Group for coder extension (transcoding) includes the G.726 coder.

1. On the inbound leg for the incoming SDP offer: The device allows and keeps the coders in the SDP that also appear in the Allowed Audio Coders Group for coder restriction (i.e., G.711 and G.729). It changes the order of listed coders in the SDP so that G.711 is listed first. The device removes the coders (i.e., G.723) from the SDP that do not appear in the Allowed Audio Coders Group for coder restriction.
2. On the outbound leg for the outgoing SDP offer: The SDP offer now includes only the G.711 and G.729 coders due to the coder restriction process on the incoming SDP offer (see Step 1).

   a. The device adds the extension coder to the SDP offer and therefore, the SDP offer now includes the G.711, G.729 and G.726 coders.

   m=audio 6050 RTP/AVP 0 8 18 96
   a=rtpmap:0 PCMU/8000
   a=rtpmap:8 PCMA/8000
   a=rtpmap:18 G729/8000
   a=rtpmap:96 G726-32/8000
   a=fmtp:4 annexa=no
   a=rtpmap:96 telephone-event/8000
   a=fmtp:96 0-15
   a=ptime:20
   a=sendrecv

   b. The device applies coder restriction to the SDP offer. As the Allowed Audio Coders Group for coder restriction includes the G.723, G.726, and G.729 coders, the device allows and keeps the G.729 and G.726, but removes the G.711 coder as it does not appear in the Allowed Audio Coders Group for coder restriction.

   m=audio 6050 RTP/AVP 18 96
   a=rtpmap:18 G729/8000
   a=rtpmap:96 G726-32/8000
   a=fmtp:4 annexa=no
   a=rtpmap:96 telephone-event/8000
   a=fmtp:96 0-15
   a=ptime:20
   a=sendrecv

3. The device includes only the G.729 and G.726 coders in the SDP offer that it sends from the outgoing leg to the outbound SIP entity. The G.729 is listed first as the Allowed Audio Coders Group for coder restriction takes precedence over the extension coder.

   To configure coder transcoding:

   1. In the Coder Groups table, configure a Coders Group for extension coders. For more information, see "Configuring Coder Groups" on page 407.

   2. In the IP Profiles table, configure the IP Profile associated with the SIP entity:

      a. Assign the Coders Group to the IP Profile, using the 'Extension Coders Group' parameter (SBCExtensionCodersGroupName).

      b. Enable extension coders by configuring the 'Allowed Coders Mode' parameter to Restriction or Restriction and Preference.
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.
- The transcoding mode can be configured globally, using the TranscodingMode parameter or for specific calls, using the IP Profiles table.

23.5.5 Transcoding Mode

By default, the device performs transcoding only if required. This refers to all types of transcoding (interworking) that require DSPs such as voice coder transcoding, DTMF negotiations, and fax negotiations. Transcoding is required, for example, when two SIP entities use different coders. In such a scenario, you would need to configure transcoding (i.e., extension coders), the device performs coder transcoding between the legs (inbound and outbound). If the SIP entities use the same coder, the device does not perform transcoding.

Alternatively, you can configure the device to always perform transcoding, regardless whether it is required or not. This is referred to as forced transcoding. For example, if the SIP entities use the same coder, the device performs transcoding of the same coder (e.g., G.711 and G.711) between the two legs.

The transcoding mode can be configured globally (TranscodingMode parameter) or per SIP entity using IP Profiles (IpProfile_TranscodingMode parameter).

Note: If the transcoding mode is configured to Force (i.e., always performs transcoding) for an IP Profile associated with a specific SIP entity, the device also applies forced transcoding for the SIP entity communicating with this SIP entity, regardless of its IP Profile settings.

23.5.6 Prioritizing Coder List in SDP Offer

In addition to restricting the use of coders using Allowed Audio Coders Groups (see "Configuring Allowed Audio Coder Groups" on page 412), you can also prioritize the coders listed in the SDP offer. This feature is referred to as Coder Preference and applies to both SBC legs:

- **Incoming SDP offer:** The device arranges the coder list in the incoming SDP offer according to the order of appearance of the Allowed Audio Coders Group that is associated with the incoming dialog. The coders listed higher up in the group take preference over ones listed lower down. To configure this, configure the 'Allowed Coders Mode' parameter (IpProfile_SBCAllowedCodersMode) in the associated IP Profile to Preference or Restriction and Preference. If you configure the parameter to Preference, the coders in the SDP offer that also appear in the Allowed Audio Coders Group are listed first in the SDP offer, and the coders in the SDP offer that do not appear in the Allowed Audio Coders Group are listed after the Allowed coders in the SDP offer. Therefore, this setting does not restrict coder use to Allowed coders, but uses (prefers) the Allowed coders whenever possible.

- **Outgoing SDP offer:** If only Allowed coders are used, the device arranges the coders in the SDP offer as described above. However, if Extension coders are also used, the coder list is arranged according to the SBCPreferencesMode parameter. Depending on the parameter's settings, the
Extension coders are added after the Allowed coders according to their order in the Allowed Audio Coders Group, or the Allowed and Extension coders are arranged according to their position in the Allowed Audio Coders Group.

23.5.7 SRTP-RTP and SRTP-SRTP Transcoding

The device supports transcoding between SRTP and RTP. The device can also enforce specific SBC legs to use SRTP and/or RTP. The device's handling of SRTP/RTP is configured using the IP Profile parameter, SBCMediaSecurityBehaviour, which provides the following options:

- SBC passes the media as is, regardless of whether it's RTP or SRTP (default).
- SBC legs negotiate only SRTP media lines (m=); RTP media lines are removed from the incoming SDP offer-answer.
- SBC legs negotiate only RTP media lines; SRTP media lines are removed from the incoming offer-answer.
- Each SDP offer-answer is extended (if not already) to two media lines for RTP and SRTP.

If after SDP offer-answer negotiation, one SBC leg uses RTP while the other uses SRTP, the device performs RTP-SRTP transcoding. To translate between RTP and SRTP, the following prerequisites must be met:

- At least one supported SDP "crypto" attribute.
- The EnableMediaSecurity parameter must be set to 1.

Transcoding where both legs are configured for SRTP is typically required to trans-encrypt and trans-decrypt. This is relevant when the MKI and Symmetric MKI parameters are enabled. In other words, both sides need to both encrypt and decrypt the outgoing and incoming SRTP packets, respectively.

DSP resources are not required for RTP-SRTP transcoding.

23.5.8 Multiple RTP Media Streams per Call Session

The device's SBC application supports multiple RTP media streams per SBC call session. It supports the negotiation of up to five media streams ('m=' line) in the SDP offer/answer model per session. The media can include a combination of any of the following types:

- Audio, indicated in the SDP as 'm=audio'
- Video, indicated in the SDP as 'm=video'
- Text, indicated in the SDP as 'm=text'
- Fax, indicated in the SDP as 'm=image'
- Binary Floor Control Protocol (BFCP), indicated in the SDP as 'm=application <port> UDP/BFCP'

Therefore, the device supports transcoding of various attributes in the SDP offer-answer (e.g., codec, port, and packetization time) per media type. If the device is unable to perform transcoding (e.g., does not support the coder), it relays the SBC dialog transparently.

The device transparently forwards Binary Floor Control Protocol (BFCP) signaling over UDP between IP entities (RFC 4582). BFCP is a signaling protocol used by some third-party conferencing servers to share content (such as video conferencing, presentations or documents) between conference participants (SIP clients supporting BFCP). The SDP offer/answer exchange model is used to establish (negotiate) BFCP streams between clients. The BFCP stream is identified in the SDP as 'm=application <port> UDP/BFCP' and a dedicated UDP port is used for the BFCP streams.
23.5.9 Interworking Miscellaneous Media Handling

This section describes various interworking features relating to media handling.

23.5.9.1 Interworking DTMF Methods

The device supports interworking between various DTMF methods such as RFC 2833, In-Band DTMF’s, and SIP INFO (Cisco\Nortel\Korea). By default, the device allows the remote user agents to negotiate (in case of RFC 2833) and passes DTMF without intervention. However, if two user agents (UA) support different DTMF methods, the device can interwork these different DTMF methods at each leg.

This DTMF interworking feature is enabled using IP Profiles (ini file parameter IPProfile):

- **SBCRFC2833Behavior** - affects the RFC 2833 SDP offer-answer negotiation:
  - [0]: (default) the device does not intervene in the RFC 2833 negotiation.
  - [1]: each outgoing offer-answer includes RFC 2833 in the offered SDP (the device adds RFC 2833 only if the incoming offer does not include RFC 2833).
  - [2]: the device removes RFC 2833 from the incoming offer.

- **SBCAlternativeDTMFMethod** – the device's first priority for DTMF method at each leg is RFC 2833. Therefore, if a specific leg negotiates RFC 2833 successfully, then the chosen DTMF method for this leg is RFC 2833. For legs where RFC 2833 is not negotiated successfully, the device uses the parameter to determine the DTMF method for the leg.

The chosen DTMF method determines (for each leg) which DTMF method is used for sending DTMF’s. If the device interworks between different DTMF methods and one of the methods is In-band\RFC 2833, detection and generation of DTMF methods requires DSP allocation.

23.5.9.2 Interworking RTP Redundancy

The device supports interworking of RTP redundancy (according to RFC 2198) between SIP entities. Employing IP Profiles, you can configure RTP redundancy handling per SIP entity:

- Generate RFC 2198 redundant packets (IpProfile_RTPRedundancyDepth parameter).
- Determine RTP redundancy support in the RTP redundancy negotiation in SDP offer/answer (IpProfile_SBCRTPRedundancyBehavior parameter). If not supported, the device discards RTP redundancy packets (if present) received from or sent to the SIP entity.

For more information, see the above parameters in "Configuring IP Profiles" on page 417.

23.5.9.3 Interworking RTP-RTCP Multiplexing

The device supports interworking of RTP-RTCP multiplexing onto a single, local UDP port (according to RFC 5761) between SIP entities. Employing IP Profiles, you can configure RTP multiplexing per SIP entity, using the IPProfile_SBCRTCPMux parameter (see "Configuring IP Profiles" on page 417).

23.5.9.4 Interworking RTCP Attribute in SDP

The device supports interworking the RTCP attribute 'a=rtcp' in the SDP between SIP entities. Employing IP Profiles, you can configure RTCP attribute handling (add, remove or transparent) per SIP entity, using the IpProfile_SBCSDPHandleRTCPAttribute parameter (see "Configuring IP Profiles" on page 417).
23.5.9.5 Interworking Crypto Lifetime Field
The device supports interworking the lifetime field in the 'a=crypto' attribute of the SDP, between SIP entities. Employing IP Profiles, you can configure the lifetime field handling (remove or retain) per SIP entity, using the IpProfile_SBCRemoveCryptoLifetimeInSDP parameter (see "Configuring IP Profiles" on page 417).

23.5.9.6 Interworking Media Security Protocols
The device supports interworking media security protocols for SRTP, between SIP entities. Employing IP Profiles, you can configure the security protocol (SDES, DTLS or both) per SIP entity, using the IPProfile_SBCMediaSecurityMethod parameter (see "Configuring IP Profiles" on page 417). For more information on SDES and DTLS, see "Configuring Media (SRTP) Security" on page 204.

23.5.9.7 Interworking ICE Lite for NAT Traversal
The device supports interworking ICE for NAT traversal, between SIP entities. Employing IP Profiles, you can enable ICE Lite per SIP entity, using the IPProfile_SBCIceMode parameter (see "Configuring IP Profiles" on page 417).

23.6 Fax Negotiation and Transcoding
The device can allow fax transmissions to traverse transparently without transcoding or it can handle the fax as follows:
- Allow interoperability between different fax machines, supporting fax transcoding if required.
- Restrict usage of specific fax coders to save bandwidth, enhance performance, or comply with supported coders. These coders include G.711 (A-Law or Mu-Law), VBD (G.711 A-Law or G.711 Mu-Law), and T38.

Fax configuration is done in the Coder Groups table and IP Profiles table. The Coder Groups table defines the supported coders, which is assigned to the IP Profile associated with the SIP entity. The IP Profiles table also defines the negotiation method used between the incoming and outgoing fax legs, using the following fax-related parameters:
- IPProfile_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.
- IPProfile_SBCFaxCodersGroupName: defines the supported fax coders (from the Coder Groups table).
- IPProfile_SBCFaxOfferMode: determines the fax coders sent in the outgoing SDP offer.
- IPProfile_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 417.
23.7 Limiting SBC Call Duration

You can configure the maximum allowed call duration (in minutes) per SBC call. If an established call reaches this user-defined limit, the device terminates the call. The feature ensures that calls are properly terminated, allowing available resources for new calls. The following procedure describes how to configure the feature for all calls (globally). To configure the feature per specific calls, use IP Profiles (IpProfile_SBCMaxCallDuration).

➢ To configure maximum call duration:

1. Open the SBC General Settings page (Setup menu > Signaling & Media tab > SBC folder > SBC General Settings).
2. In the ‘Max Call Duration’ field (SBCMaxCallDuration), enter the maximum call duration per SBC call:

   ![Figure 23-5: Configuring Maximum Call Duration](image)

3. Click Apply.

23.8 SBC Authentication

The device can authenticate SIP servers and SBC users (clients). The different authentication methods are described in the subsequent subsections.

23.8.1 SIP Authentication Server Functionality

The device can function as an Authentication server for authenticating received SIP message requests, based on HTTP authentication Digest with MD5. Alternatively, such requests can be authenticated by an external, third-party server.

When functioning as an Authentication server, the device can authenticate the following SIP entities:

■ SIP servers: This is applicable to Server-type IP Groups. This provides protection from rogue SIP servers, preventing unauthorized usage of device resources and functionality. To authenticate remote servers, the device challenges the server with a user-defined username and password that is shared with the remote server. When the device receives an INVITE request from the remote server, it challenges the server by replying with a SIP 401 Unauthorized response containing the WWW-Authenticate header. The remote server then re-sends the INVITE containing an Authorization header with authentication information based on this username-password combination to confirm its identity. The device uses the username and password to authenticate the message prior to processing it.

■ SIP clients: These are clients belonging to a User-type IP Group. This support prevents unauthorized usage of the device’s resources by rogue SIP clients. When the device receives an INVITE or REGISTER request from a client (e.g., SIP phone) for SIP message authorization, the device processes the authorization as follows:
23. SBC Overview

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 Groups 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 663). 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 Groups table (see "Configuring IP Groups" on page 345).

23.8.2 User Authentication based on RADIUS

The device can authenticate SIP clients (users) using a remote RADIUS server. The device supports the RADIUS extension for digest authentication of SIP clients, according to draft-sterman-aaa-sip-01. Based on this standard, the device generates the nonce (in contrast to RFC 5090, where it is done by the RADIUS server).

RADIUS based on draft-sterman-aaa-sip-01 operates as follows:

1. The device receives a SIP request without an Authorization header from the SIP client.
2. The device generates the nonce and sends it to the client in a SIP 407 (Proxy Authentication Required) response.
3. The SIP client sends the SIP request with the Authorization header to the device.
4. The device sends an Access-Request message to the RADIUS server.
5. The RADIUS server verifies the client's credentials and sends an Access-Accept (or Access-Reject) response to the device.
6. The device accepts the SIP client's request (sends a SIP 200 OK or forwards the authenticated request) or rejects it (sends another SIP 407 to the SIP client).

To configure this feature, set the SBCServerAuthMode ini file parameter to 2.

23.9 Interworking SIP Signaling

The device supports interworking of SIP signaling messages to ensure interoperability between communicating SIP UAs or entities. This is critical in network environments where the UAs on opposing SBC legs have different SIP signaling support. For example, some UAs may support different versions of a SIP method while others may not 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.

### 23.9.1 Interworking SIP 3xx Redirect Responses

The device supports interworking of SIP 3xx redirect responses. By default, the device's handling of SIP 3xx responses is to send the Contact header unchanged. However, some SIP UAs may support different versions of the SIP 3xx standard while others may not even support SIP 3xx.

The handling of SIP 3xx can be configured for all calls, using the global parameter SBC3xxBehavior. To configure different SIP 3xx handling options for different UAs (i.e., per IP Group), use the IP Profiles table parameter, 'SBC Remote 3xx Mode'.

#### 23.9.1.1 Resultant INVITE Traversing Device

The device can handle SIP 3xx responses so that the new INVITE message sent as a result of the 3xx traverses the device. The reasons for enforcing resultant INVITEs to traverse the device may vary:

- The user that receives the 3xx is unable to route to the 3xx contact (i.e., the user is on the LAN and the new contact is on the WAN). In such a scenario, the device enables the user to reach the WAN contact and overcome NAT problems.
- Enforce certain SBC policies (e.g., call admission control, header manipulation, and transcoding) on the resultant INVITE.

The device enforces this by modifying each Contact in the 3xx response as follows:

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

The SBC handling for the 3xx resultant INVITE is as follows:

1. The incoming INVITE is identified as a 3xx resultant INVITE according to the special prefix.
2. The device automatically replaces the SBC host part (in the Request-URI) with the host from the 3xx Contact.
3. The prefix ("T~&R_") remains in the user part for the classification, manipulation, and routing mechanisms.
4. The classification, manipulation, and routing processes are done exactly like any other INVITE handling. The special prefix can be used for specific routing rules for 3xx resultant INVITEs.
5. The prefix is removed before the resultant INVITE is sent to the destination.

**Figure 23-6: 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).

### 23.9.1.2 Local Handling of SIP 3xx

The device can handle SIP 3xx responses on behalf of the dialog-initiating UA and retry the request (e.g., INVITE) using one or more alternative URIs included in the 3xx response. The new request includes SIP headers from the initial request such as Diversion, History-Info, P-Asserted-Id, and Priority. The source and destination URIs can be manipulated using the regular manipulation mechanism.

The device sends the new request to the alternative destination according to the IP-to-IP Routing table rules. (where the ‘Call Trigger’ field is set to 3xx). It is also possible to specify the IP Group that sent the 3xx request as matching criteria for the re-routing rule in this table (*ReRoute IP Group ID* field).
23.9.2 Interworking SIP Diversion and History-Info Headers

This device can be configured to interwork between the SIP Diversion and History-Info headers. This is important, for example, to networks that support the Diversion header but not the History-Info header, or vice versa. Therefore, mapping between these headers is crucial for preserving the information in the SIP dialog regarding how and why (e.g., call redirection) the call arrived at a certain SIP UA. If the Diversion header is used, you can specify the URI type (e.g., "tel:" ) to use in the header, using the SBCDiversionUriType parameter.

This feature is configured in the IP Profiles 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>
<thead>
<tr>
<th>Parameter Value</th>
<th>SIP Header Present in Received SIP Message</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Diversion</td>
</tr>
<tr>
<td>HistoryInfoMode = Add</td>
<td>Diversion converted to</td>
</tr>
<tr>
<td>DiversionMode = Remove</td>
<td>History-Info</td>
</tr>
<tr>
<td></td>
<td>Diversion removed.</td>
</tr>
<tr>
<td>HistoryInfoMode = Remove</td>
<td>Not present.</td>
</tr>
<tr>
<td>DiversionMode = Add</td>
<td></td>
</tr>
<tr>
<td>HistoryInfoMode = Disable</td>
<td>Diversion converted to</td>
</tr>
<tr>
<td>DiversionMode = Add</td>
<td>History-Info</td>
</tr>
<tr>
<td>HistoryInfoMode = Add</td>
<td>Not present.</td>
</tr>
<tr>
<td>DiversionMode = Add</td>
<td></td>
</tr>
<tr>
<td>HistoryInfoMode = Add</td>
<td>Diversion converted to</td>
</tr>
<tr>
<td>DiversionMode = Add</td>
<td>History-Info</td>
</tr>
<tr>
<td>HistoryInfoMode = Remove</td>
<td>Diversion removed.</td>
</tr>
<tr>
<td>DiversionMode = Remove</td>
<td></td>
</tr>
</tbody>
</table>

23.9.3 Interworking SIP REFER Messages

The device supports interworking of SIP REFER messages. SIP UAs may support different versions of the REFER standard while others may not even support REFER. This feature supports the following:

- Attended, unattended, and semi-attended call transfers
- Sending INVITE, REFER-notifications, BYE, PRACK and Session Timer on behalf of peer PBXs
Advanced routing rules for the new, initiated INVITE
Forwarding early media after REFER while attempting to avoid transcoding (by sending session update)
Interoperate with environments 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. To configure different REFER handling options for different UAs (i.e., IP Groups), use the IP Profiles table parameter, ‘Remote REFER Mode’.

Local handling of REFER: This option is used for UAs that do not support REFER. Upon receipt of a REFER request, instead of forwarding it to the IP Group, the device handles it locally. It generates a new INVITE to the alternative destination according to the rules in the IP-to-IP Routing table (where the 'Call Trigger' field is set to REFER). It is also possible to specify the IP Group that sent the REFER request, as matching criteria for the re-routing rule in this table ('ReRoute IP Group ID' field).

Transparent handling: The device forwards the REFER with the Refer-To header unchanged.
Re-routing through SBC: The device changes the Refer-To header so that the re-routed INVITE is sent through the SBC application.
IP Group Name: The device sets the host part in the REFER message to the name configured for the IP Group in the IP Groups table.

23.9.4 Interworking SIP PRACK Messages

The device supports interworking of SIP Provisional Response ACKnowledgement (PRACK) messages (18x). While some UAs may not support PRACK (RFC 3262) others may require it. The device can be configured to resolve this interoperable issue and enable sessions between such endpoints. SIP PRACK handling is configured using the IP Profile parameter, ‘SBC Prack Mode’:
Optional: PRACK is optional for these UAs. If required, the device performs the PRACK process on behalf of the destination UA.
Mandatory: PRACK is required for these UAs. Calls from UAs that do not support PRACK are rejected. Calls destined to these UAs are also required to support PRACK.
Transparent (default): The device does not intervene with the PRACK process and forwards the request as is.

23.9.5 Interworking SIP Session Timer

The device supports interworking of the SIP signaling keep-alive mechanism. The SIP standard provides a signaling keep-alive mechanism using re-INVITE and UPDATE messages. In certain setups, keep-alive may be required by some SIP UAs while for others it may not be supported. The device can resolve this mismatch by performing the keep-alive process on behalf of SIP UAs that do not support it.
To configure the handling of session expires, use the IP Profile parameter, ‘SBC Session Expires Mode’.

23.9.6 Interworking SIP Early Media

The device supports early media. Early media is when the media flow starts before the SIP call is established (i.e., before the 200 OK response). This occurs when the first SDP offer-
answer transaction completes. The offer-answer options can be included in the following SIP messages:

- Offer in first INVITE, answer on 180, and no or same answer in the 200 OK
- Offer in first INVITE, answer on 180, and a different answer in the 200 OK (not standard)
- INVITE without SDP, offer in 180, and answer in PRACK
- PRACK and UPDATE transactions can also be used for initiating subsequent offer-answer transactions before the INVITE 200 OK response.
- In a SIP dialog lifetime, media characteristics after originally determined by the first offer-answer transaction can be changed by using subsequent offer-answer transactions. These transactions may be carried either in UPDATE or re-INVITE transactions. The media handling is similar to the original offer-answer handling. If the offer is rejected by the remote party, no media changes occur (e.g., INVITE without SDP, then 200 OK and ACK, offer-answer within an offer-answer, and Hold re-INVITE with IP address of 0.0.0.0 - IP address is unchanged).

The device supports various interworking modes for early media between SIP UAs (i.e., IP Groups):

- **Early Media Enabling**: The device supports the interworking of early media between SIP UAs that support early media and those that do not support receipt of early media. Early media can arrive in provisional responses to an INVITE request. The device forwards the request of early media for IP Groups that support this capability; otherwise, the device terminates it. Provisional responses whose SDP are suppressed are changed to a SIP 180 response. This feature is also supported for delayed offers. This is configured using the IP Profile parameter, 'SBC Remote Early Media Support'. The device refers to the parameter also for features that require early media such as playing ringback tone.

- **Early Media Response Type**: The device supports the interworking of different SIP provisional response types between UAs for forwarding the early media to the caller. This can support all early media response types (default), SIP 180 only, or SIP 183 only, and is configured by the IP Profile parameter, 'SBC Remote Early Media Response Type'.

- **Multiple 18x**: The device supports the interworking of different support for multiple 18x responses (including 180 Ringing, 181 Call is Being Forwarded, 182 Call Queued, and 183 Session Progress) that are forwarded to the caller. The UA can be configured as supporting only receipt of the first 18x response (i.e., the device forwards only this response to the caller), or receipt of multiple 18x responses (default). This is configured by the IP Profile parameter, 'SBC Remote Multiple 18x Support'.
**Early Media RTP:** The device supports the interworking with remote clients that send 18x responses with early media and whose subsequent RTP is delayed, and with remote clients that do not support this and require RTP to immediately follow the 18x response. Some clients do not support 18x with early media, while others require 18x with early media (i.e., they cannot play ringback tone locally). These various interworking capabilities are configured by the IP Profile parameters, 'Remote Early Media RTP Detection Mode', 'SBC Remote Supports RFC 3960', and 'SBC Remote Can Play Ringback'. See the flowcharts below for the device's handling of such scenarios:

**Figure 23-7: SBC Early Media RTP 18x without SDP**

- SBC receives SIP 18x without SDP from Destination
- Source UA can play local Ringback tone?
- Yes
- No
- Source UA supports Early Media?
- Yes
- SBC sends Source 18x with SDP and plays ringback tone to Destination
- No
- SBC sends Source UA 18x without SDP (forwards as is)
### 23.9.7 Interworking SIP re-INVITE Messages

The device supports interworking of SIP re-INVITE messages. This enables communication between endpoints that generate re-INVITE requests and those that do not support the receipt of re-INVITEs. The device does not forward re-INVITE requests to IP Groups that do not support it. Instead, it sends a SIP response to the re-INVITE request, which can either be a success or a failure, depending on whether the device can bridge the media between the endpoints. The device can also handle re-INVITEs with or without an SDP body, enabling communication between endpoints that do not support re-INVITE requests without SDP, and those that require SDP. The device generates an SDP offer and adds it to the incoming re-INVITE request if it does not contain an SDP and only then forwards it to the destination endpoint. This interworking support is configured by the IP Profile parameter, 'SBC Remote Reinvite Support'.

### 23.9.8 Interworking SIP UPDATE Messages

The device supports interworking of the SIP UPDATED message. This enables communication between UAs that generate UPDATE requests and those that do not support the receipt of UPDATE requests. The device does not forward UPDATE requests to IP Groups that do not support it. Instead, it sends a SIP response to the UPDATE request which can either be a success or a failure, depending on whether the device can bridge the media between the endpoints. The handling of UPDATE messages is configured by the IP Profile parameter 'SBC Remote Update Support'.
23.9.9 Interworking SIP re-INVITE to UPDATE

The device enables communication between endpoints (IP Groups) that do not support re-INVITE requests but support the UPDATE method, and vice versa. The device translates the re-INVITE request to the UPDATE request, and vice versa. Note that if a re-INVITE request arrives without SDP, the device generates the SDP and adds it to the outgoing UPDATE request. To enable this feature, each IP Group needs to be configured with its unique capabilities by associating it with a relevant IP Profile. For example, an IP Group that supports UPDATE requests but not re-INVITEs would be configured as follows:

- SBCRemoteUpdateSupport = 2 (Supported)
- SBCRemoteReinviteSupport = 0 (Not Supported)

If a re-INVITE request needs to be forwarded to this IP Group, it is translated to an UPDATE request.

23.9.10 Interworking Delayed Offer

The device supports interworking of INVITE messages with and without SDP between SIP entities. The device enables sessions between endpoints (IP Groups) that send INVITEs without SDP (i.e., delayed media) and those that do not support the receipt of INVITEs without SDP. The device creates an SDP and adds it to INVITEs that arrive without SDP. This intervention in the SDP offer-answer process may require transcoding. Delayed offer is also supported when early media is present.

Employing IP Profiles, you can configure this interworking feature per SIP entity, using the 'SBC Remote Delayed Offer Support' parameter (see "Configuring IP Profiles" on page 417).

Note: For SIP entities that do not support delayed offer, you must assign extension coders to its IP Profile (using the 'Extension Coders' parameter).

23.9.11 Interworking Call Hold

The device supports the interworking of call hold / retrieve requests between SIP entities supporting different call hold capabilities:

- Interworking SDP call hold formats. This is configured by the IP Profile parameter, 'SBC Remote Hold Format'.
- Interworking the play of the held tone for IP entities that cannot play held tones locally. This is configured by the IP Profile parameter, 'SBC Play Held Tone'.
- Interworking generation of held tone where the device generates the tone to the held party instead of the call hold initiator. This is configured by the IP Profile parameter, 'SBC Reliable Held Tone Source'.

To configure IP Profiles, see "Configuring IP Profiles" on page 417.

23.9.12 Interworking SIP Via Headers

The device supports the interworking of SIP Via headers between SIP entities. For the outgoing message sent to a SIP entity, the device can remove or retain all the Via headers received in the incoming SIP request from the other side. Employing IP Profiles, you can
configure this interworking feature per SIP entity, using the IpProfile_SBCKeepVIAHeaders parameter (see "Configuring IP Profiles" on page 417).

### 23.9.13 Interworking SIP User-Agent Headers

The device supports the interworking of SIP User-Agent headers between SIP entities. For the outgoing message sent to a SIP entity, the device can remove or retain all the User-Agent headers received in the incoming SIP request/response from the other side. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the IpProfile_SBCKeepUserAgentHeader parameter (see "Configuring IP Profiles" on page 417).

### 23.9.14 Interworking SIP Record-Route Headers

The device supports the interworking of SIP Record-Route headers between IP entities. For the outgoing message sent to a SIP entity, the device can remove or retain all the Record-Route headers received in the incoming SIP request/response from the other side. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the IpProfile_SBCKeepRoutingHeaders parameter (see "Configuring IP Profiles" on page 417).

### 23.9.15 Interworking SIP To-Header Tags in Multiple SDP Answers

The device supports the interworking of SIP To-header tags in call forking responses (i.e., multiple SDP answers) between IP entities. The device can either use the same To-header tag value for all SDP answers sent to the SIP entity, or send each SDP answer with its original tag. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the IpProfile_SBCRemoteMultipleEarlyDialogs parameter (see "Configuring IP Profiles" on page 417).

### 23.9.16 Interworking In-dialog SIP Contact and Record-Route Headers

The device supports the interworking of in-dialog, SIP Contact and Record-Route headers between SIP entities. Employing IP Profiles, you can configure this interworking feature per SIP entity, using the IpProfile_SBCRemoteRepresentationMode parameter (see "Configuring IP Profiles" on page 417).
24 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 License Key that includes this feature. For installing a License Key, see "License Key" on page 669.

➢ To enable the SBC application:

1. Open the Applications Enabling page (Setup menu > Signaling & Media tab > Core Entities folder > Applications Enabling).
2. From the 'SBC Application' drop-down list, select Enable:

   ![Enable SBC Application]

3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.
This page is intentionally left blank.
25 Configuring General SBC Settings

This section describes configuration of various SBC features.

25.1 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.

To enable the feature:

- Configure the 'SBC Dialog-Info Interworking' (EnableSBCDialogInfoInterworking) parameter to Enable.

When the 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
<?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="CCDORRTDRKIKWFVFBRWYM" direction="initiator">
  <state event="replaced">terminated</state>
</dialog>
<dialog id="sfhjsjk12" call-id="67402270@10.132.10.150"
  local-tag="1c137249965"
  remote-tag="CCDORRTDRKIKWFVFBRWYM" direction="receiver">
  <state reason="replaced">confirmed</state>
  <replaces
call-id="67402270@10.132.10.150"
  local-tag="1c137249965"
  remote-tag="CCDORRTDRKIKWFVFBRWYM"/>
  <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>
26 Configuring Admission Control

The Admission Control table lets you configure up to 1,500 Call Admission Control rules (CAC). CAC rules define the maximum number of concurrent calls (SIP dialogs) permitted per IP Group, SIP Interface or SRD, and per user (identified by its registered contact). CAC rules also define a guaranteed (reserved) number of concurrent calls. Thus, CAC rules can be useful for implementing Service Level Agreements (SLA) policies.

CAC rules can be applied per SIP request type and SIP dialog direction (inbound and/or outbound). These relate to requests that initiate SIP dialogs and not the subsequent requests that can be of different type and direction. The SIP dialog-initiating request types can include INVITE, REGISTER, and/or SUBSCRIBE messages, or it can be configured to include the total number of all dialogs.

This feature also provides support for SIP-dialog rate control, using the “token bucket” mechanism. The token bucket is a control mechanism that dictates the rate of SIP-dialog setups based on the presence of tokens in the bucket – a logical container that holds aggregate SIP dialogs to be accepted or transmitted. Tokens in the bucket are removed (“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 an SRD and each of its associated IP Groups, by configuring multiple CAC rules. In such a setup, the SRD's reserved call capacity must be greater or equal to the summation of the reserved call capacity of all these IP Groups. In other words, the SRD serves as the "parent" reserved call capacity. If the SRD's reserved call capacity is greater, the extra call capacity can be used as a shared pool between the IP Groups for unreserved calls when they exceed their reserved capacity. For example, assume that the reserved capacities for an SRD and its associated IP Groups are as follows:

- SRD reserved call capacity: 40
- IP Group ID 1 reserved call capacity: 10
- IP Group ID 2 reserved call capacity: 20

In this setup, the SRD offers a shared pool for unreserved call capacity of 10 [i.e., 40 – (10 + 20)]. If IP Group ID 1 needs to handle 15 calls, it is guaranteed 10 calls and the remaining 5 is provided from the SRD's shared pool. If the SRD's shared pool is currently empty and resources for new calls are required, the quota is taken from the device's total capacity, if available. For example, if IP Group ID 1 needs to handle 21 calls, it's guaranteed 10, the SRD's shared pool provides another 10, and the last call is provided from the device's total call capacity support (e.g., of 200).

Requests that reach the user-defined call limit (maximum concurrent calls and/or call rate) are sent to an alternative route if configured in the IP-to-IP Routing table. If no alternative
routing rule exists, the device rejects the SIP request with a SIP 480 "Temporarily Unavailable" response.

**Note:** The device applies the CAC rule for the incoming leg immediately after the Classification process. If the call/request is rejected at this stage, no routing is performed. The enforcement for the outgoing leg is performed within each alternative route iteration. This is accessed from two places: one during initial classification/routing, and another during alternative routing process.

The following procedure describes how to configure CAC rules through the Web interface. You can also configure it through ini file (SBCAdmissionControl) or CLI (configure voip > sbc sbc-admission-control).

➢ **To configure a CAC rule:**

1. Open the Admission Control table (Setup menu > Signaling & Media tab > SBC folder > Admission Control).
2. Click New; the following dialog box appears:

   ![Figure 26-1: Admission Control Table - Add Dialog Box](image)

3. Configure an Admission Control rule according to the parameters described in the table below.
4. Click Apply.

**Table 26-1: Admission Control Table Parameter Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRD</td>
<td>Assigns an SRD to the rule. By default, no value is defined. For all SRDs, configure the parameter to Any. <strong>Note:</strong> The parameter is applicable only if 'Limit Type' is configured to SRD.</td>
</tr>
<tr>
<td>srd-name</td>
<td></td>
</tr>
<tr>
<td>[SBCAdmissionControl_SRDNam e]</td>
<td></td>
</tr>
</tbody>
</table>

**General**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>[SBCAdmissionControl_Index]</td>
<td></td>
</tr>
</tbody>
</table>
### 26. Configuring Admission Control

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. By default, no value is defined.</td>
</tr>
<tr>
<td>Limit Type</td>
<td>Defines the entity to which the rule applies.</td>
</tr>
<tr>
<td></td>
<td>- [0] IP Group (default)</td>
</tr>
<tr>
<td></td>
<td>- [1] SRD</td>
</tr>
<tr>
<td></td>
<td>- [2] SIP Interface</td>
</tr>
<tr>
<td>IP Group</td>
<td>Assigns an IP Group to the rule if the rule is applied to an IP Group. By default, no value is defined. For all IP Groups, configure the parameter to Any. Note: The parameter is applicable only if 'Limit Type' is configured to IP Group.</td>
</tr>
<tr>
<td>SIP Interface</td>
<td>Assigns a SIP Interface to the rule if the rule is applied to a SIP Interface. By default, no value is defined. For all SIP Interfaces, configure the parameter to Any. Note: The parameter is applicable only if 'Limit Type' is configured to SIP Interface.</td>
</tr>
<tr>
<td>Request Type</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>- [0] All = (Default) Includes the total number of all dialogs.</td>
</tr>
<tr>
<td></td>
<td>- [1] INVITE</td>
</tr>
<tr>
<td></td>
<td>- [2] SUBSCRIBE</td>
</tr>
<tr>
<td></td>
<td>- [3] Other = All SIP request types except INVITEs and SUBSCRIBEs (e.g., REGISTER).</td>
</tr>
<tr>
<td>Request Direction</td>
<td>Defines the call direction of the SIP request to which the rule applies.</td>
</tr>
<tr>
<td></td>
<td>- [0] Both = (Default) Rule applies to inbound and outbound SIP dialogs.</td>
</tr>
<tr>
<td></td>
<td>- [1] Inbound = Rule applies only to inbound SIP dialogs.</td>
</tr>
<tr>
<td></td>
<td>- [2] Outbound = Rule applies only to outbound SIP dialogs.</td>
</tr>
<tr>
<td>Limit</td>
<td>Defines the guaranteed (minimum) call capacity. The default is 0 (i.e., no reserved capacity). Note:</td>
</tr>
<tr>
<td></td>
<td>- Reserved call capacity is applicable only to IP Groups and SRDs (i.e., 'Limit Type' parameter configured to IP Group or SRD). If you configure the 'Limit Type' parameter to SIP Interface, leave the 'Reserved Capacity' parameter at its default (i.e., 0).</td>
</tr>
<tr>
<td></td>
<td>- Reserved call capacity is applicable only to INVITE and SUBSCRIBE messages.</td>
</tr>
<tr>
<td></td>
<td>- Reserved call capacity must be less than the maximum</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>capacity (limit) configured</td>
<td>for the CAC rule (see the 'Limit' parameter below).</td>
</tr>
<tr>
<td></td>
<td>• The total reserved call capacity configured for all CAC rules must be within the device's total call capacity support.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Limit</td>
<td>Defines the maximum number of concurrent SIP dialogs per IP Group, SIP Interface or SRD. You can also use the following special values:</td>
</tr>
<tr>
<td></td>
<td>• [0] 0 = Block all these dialogs.</td>
</tr>
<tr>
<td></td>
<td>• [-1] -1 = (Default) Unlimited.</td>
</tr>
<tr>
<td>Limit Per User</td>
<td>Defines the maximum number of concurrent SIP dialogs per user belonging to the specified IP Group, SIP Interface or SRD. You can also use the following special values:</td>
</tr>
<tr>
<td></td>
<td>• [-1] -1 = (Default) Unlimited.</td>
</tr>
<tr>
<td></td>
<td>• [0] 0 = Block all these dialogs.</td>
</tr>
<tr>
<td>Rate</td>
<td>Defines the maximum number of SIP dialogs per IP Group, SIP Interface or SRD that can be handled per second</td>
</tr>
<tr>
<td></td>
<td>The default is 0 (i.e., unlimited rate).</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• You must first configure the 'Maximum Burst' parameter (see below) before configuring the 'Rate' parameter.</td>
</tr>
<tr>
<td></td>
<td>• The token bucket feature is per IP Group, SIP Interface, SRD, SIP request type, and SIP request direction.</td>
</tr>
<tr>
<td>Maximum Burst</td>
<td>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, 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 configured in the 'Rate' field), the device accepts the first 100 SIP dialogs and rejects the last one. The device sends a SIP 480 “Temporarily Unavailable” response when it rejects requests. Dropped requests are not counted in the bucket. The default is 0 (i.e., unlimited SIP dialogs). <strong>Note:</strong> The token bucket feature is per IP Group, SIP Interface, SRD, SIP request type, and SIP request direction.</td>
</tr>
<tr>
<td>max-burst</td>
<td></td>
</tr>
</tbody>
</table>
# 27 Routing SBC

This section describes the configuration of the call routing entities for the SBC application.

## 27.1 Configuring Classification Rules

The Classification table lets you configure up to 1,500 Classification rules. A Classification rule classifies incoming SIP dialog-initiating requests (e.g., INVITE messages) to a "source" IP Group. The source IP Group is the SIP entity that sent the SIP dialog request. Once classified, the device uses the IP Group to process the call (manipulation and routing).

You can also use the Classification table for employing SIP-level access control for successfully classified calls, by configuring Classification rules with whitelist and blacklist settings. If a Classification rule is configured as a whitelist ("Allow"), the device accepts the SIP dialog and processes the call. If the Classification rule is configured as a blacklist ("Deny"), the device rejects the SIP dialog.

Configuration of Classification rules includes two areas:

- **Match**: Defines the matching characteristics of the incoming IP call (e.g., source SIP Interface and IP address). Classification is primarily based on the SIP Interface (as the matching characteristics) on which the incoming dialog is received. As Classification rules must first be assigned with an SRD, the SIP Interface is one that belongs to the SRD. Therefore, Classification rules are configured per SRD, where multiple SIP Interfaces can be used as matching characteristics. However, as multiple SRDs are relevant only for multi-tenant deployments, for most deployments only a single SRD is required. As the device provides a default SRD ("Default_SRD"), when only one SRD is required, the device automatically assigns it to the Classification rule.

- **Action**: Defines the action that is done if the incoming call matches the characteristics of the rule (i.e., classifies the call to the specified IP Group).

The device searches the table from **top to bottom** for the **first** rule that matches the characteristics of the incoming call. If it finds a matching rule, it classifies the call to the IP Group configured for that rule.

**Note:** Configure stricter classification rules higher up in the table than less strict rules to ensure incoming dialogs are classified to the desired IP Group. **Strict** refers to the number of matching characteristics configured for the rule. For example, a rule configured with source host name and destination host name as matching characteristics is stricter than a rule configured with only source host name. If the rule configured with only source host name appears higher up in the table, the device ("erroneously") uses the rule to classify incoming dialogs matching this source host name (even if they also match the rule appearing lower down in the table configured with the destination host name as well).

If the device doesn't find a matching rule (i.e., classification fails), the device rejects or allows the call depending on the following configuration:

- **To configure the action for unclassified calls:**
  1. Open the SBC General Settings (Setup menu > Signaling & Media tab > SBC folder > SBC General Settings).
2. From the 'Unclassified Calls' drop-down list, select **Reject** to reject unclassified calls or **Allow** to accept unclassified calls:

Figure 27-1: Configuring Action for Classification Failure

![Unclassified Calls drop-down list](image)

3. Click **Apply**.

If you configure the parameter to **Allow**, the incoming SIP dialog is assigned to an IP Group as follows:

1. The device determines on which SIP listening port (e.g., 5061) the incoming SIP dialog request was received and the SIP Interface configured with this port (in the SIP Interfaces table).
2. The device determines the SRD associated with this SIP Interface (in the SIP Interfaces table) and then classifies the SIP dialog to the first IP Group in the IP Groups table that is associated with the SRD. For example, if IP Groups 3 and 4 belong to the same SRD, the device classifies the call to IP Group 3.

**Note:** If classification of a SIP request fails and you configure the device to reject unclassified calls, the device can send a specific SIP response code per SIP Interface. To configure this, use the 'Classification Failure Response Type' parameter in the SIP Interfaces table (see "Configuring SIP Interfaces" on page 337).

The Classification table is used to classify incoming SIP dialog requests **only if** the following classification stages fail:

1. **Classification Stage 1 - Based on User Registration Database:** The device searches its users registration database to check whether the incoming SIP dialog arrived from a registered user. The device searches the database for a user that matches the address-of-record (AOR) and Contact of the incoming SIP message:
   - Compares the SIP Contact header to the contact value in the database.
   - Compares the URL in the SIP P-Asserted-Identity/From header to the registered AOR in the database.

   If the device finds a matching registered user, it classifies the user to the IP Group associated with the user in the database. If this classification stage fails, the device proceeds to classification based on Proxy Set.

2. **Classification Stage 2 - Based on Proxy Set:** If the database search fails, the device performs classification based on Proxy Set. This classification is applicable only to Server-type IP Groups and is done only if classification based on Proxy Set is enabled (see the 'Classify By Proxy Set' parameter in the IP Groups table in "Configuring IP Groups" on page 345). The device checks whether the incoming INVITE's IP address (if host name, then according to the dynamically resolved IP address list) is configured for a Proxy Set (in the Proxy Sets table). If such a Proxy Set exists, the device classifies the INVITE to the IP Group that is associated with the Proxy Set. The Proxy Set is assigned to the IP Group in the IP Groups table.

   If more than one Proxy Set is configured with the same IP address and port combination to the same IP Group, the device may classify and route the SIP dialog to an incorrect IP Group. In such a scenario, a warning is generated in the Syslog message. Therefore, when classification is by Proxy Set, pay attention to the configured IP addresses and the 'Classification Input' parameter of your Proxy Sets. When more than one Proxy Set is configured with the same IP address, the device selects the matching Proxy Set in the following precedence order:
a. Selects the Proxy Set whose IP address, port, and transport type match the source of the incoming dialog.

b. If no match is found for a), it selects the Proxy Set whose IP address and transport type match the source of the incoming dialog (if the 'Classification Input' parameter is configured to IP Address Only).

c. If no match is found for b), it selects the Proxy Set whose IP address match the source of the incoming dialog (if the 'Classification Input' parameter is configured to IP Address Only).

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

Note:

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

- If multiple IP Groups are associated with the same Proxy Set, use Classification rules to classify the incoming dialogs to the IP Groups (do not use the Classify by Proxy Set feature).

- The device saves incoming SIP REGISTER messages in its registration database. If the REGISTER message is received from a User-type IP Group, the device sends the message to the configured destination.
The flowchart below illustrates the classification process:

Figure 27-2: Classification Process (Identifying IP Group or Rejecting Call)

The following procedure describes how to configure Classification rules through the Web interface. You can also configure it through ini file (Classification) or CLI (configure voip > sbc classification).

➢ To configure a Classification rule:
   1. Open the Classification table (Setup menu > Signaling & Media tab > SBC folder > Classification Table).
2. Click **New**; the following dialog box appears:

**Figure 27-3: Classification Table - Add Dialog Box**

3. Configure the Classification rule according to the parameters described in the table below.

4. Click **Apply**.

**Table 27-1: Classification Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRD srd-name [Classification_SRDN]</td>
<td>Assigns an SRD to the rule as a matching characteristic for the incoming SIP dialog. If only one SRD is configured in the SRDs table, the SRD is assigned to the rule by default. If multiple SRDs are configured in the SRDs table, no value is assigned. To configure SRDs, see &quot;Configuring SRDs&quot; on page 327. <strong>Note:</strong> The parameter is mandatory.</td>
</tr>
<tr>
<td>Match</td>
<td></td>
</tr>
<tr>
<td>Index [Classification_Index]</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name classification-name [Classification_ClassificationName]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. By default, no name is defined. <strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>Source SIP Interface src-sip-interface-name [Classification_SrcSIPInterfaceName]</td>
<td>Assigns a SIP Interface to the rule as a matching characteristic for the incoming SIP dialog. The default is <strong>Any</strong> (i.e., all SIP Interfaces belonging to the SRD assigned to the rule). <strong>Note:</strong> The SIP Interface must belong to the SRD assigned to the</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Source IP Address</td>
<td>Defines a source IP address as a matching characteristic for the incoming SIP dialog. The valid value is an IP address in dotted-decimal notation. In addition, the following wildcards can be used:</td>
</tr>
<tr>
<td>src-ip-address</td>
<td>• &quot;x&quot; wildcard: represents single digits. For example, 10.8.8.xx represents all addresses between 10.8.8.10 and 10.8.8.99.</td>
</tr>
<tr>
<td>[Classification_SrcAddress]</td>
<td>• Asterisk (<em>) wildcard: represents any number between 0 and 255. For example, 10.8.8.</em> represents all addresses between 10.8.8.0 and 10.8.8.255. By default, no value is defined (i.e., any source IP address is accepted).</td>
</tr>
<tr>
<td>Source Transport Type</td>
<td>Defines the source transport type as a matching characteristic for the incoming SIP dialog.</td>
</tr>
<tr>
<td>src-transport-type</td>
<td>• [-1] Any = (Default) All transport types</td>
</tr>
<tr>
<td>[Classification_SrcTransportType]</td>
<td>• [0] UDP</td>
</tr>
<tr>
<td></td>
<td>• [1] TCP</td>
</tr>
<tr>
<td></td>
<td>• [2] TLS</td>
</tr>
<tr>
<td>Source Port</td>
<td>Defines the source port number as a matching characteristic for the incoming SIP dialog. By default, no value is defined.</td>
</tr>
<tr>
<td>src-port</td>
<td>[Classification_SrcPort]</td>
</tr>
<tr>
<td>Source Username Prefix</td>
<td>Defines the prefix of the source URI user part as a matching characteristic for the incoming SIP dialog. The URI is typically located in the SIP From header. However, you can configure the SIP header from where the device obtains the source URI, in the IP Groups table (&quot;Source URI Input&quot; parameter). For more information on how the device obtains the URI, see &quot;SIP Dialog Initiation Process&quot; on page 457. 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 &quot;Dialing Plan Notation for Routing and Manipulation&quot; on page 827. Note: For REGISTER requests, the source URI is obtained from the To header.</td>
</tr>
<tr>
<td>src-user-name-prefix</td>
<td>[Classification_SrcUsernamePrefix]</td>
</tr>
<tr>
<td>Source Host</td>
<td>Defines the prefix of the source URI host name as a matching characteristic for the incoming SIP dialog. The URI is typically located in the SIP From header. However, you can configure the SIP header from where the device obtains the source URI, in the IP Groups table (&quot;Source URI Input&quot; parameter). For more information on how the device obtains this URI, see &quot;Call Processing of SIP Dialog Requests&quot; on page 457.</td>
</tr>
<tr>
<td>src-host</td>
<td>[Classification_SrcHost]</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dst-user-name-prefix</td>
<td>The default is the asterisk (*) symbol, which represents any source host prefix. <strong>Note:</strong> For REGISTER requests, the source URI is obtained from the To header.</td>
</tr>
<tr>
<td>dst-host</td>
<td>Defines the prefix of the destination Request-URI host name as a matching characteristic for the incoming SIP dialog. The default is the asterisk (*) symbol, which represents any destination host prefix.</td>
</tr>
<tr>
<td>message-condition-name</td>
<td>Assigns a Message Condition rule to the Classification rule as a matching characteristic for the incoming SIP dialog. By default, no value is defined. To configure Message Condition rules, see &quot;Configuring Message Condition Rules&quot; on page 501.</td>
</tr>
</tbody>
</table>
| action-type | Defines a whitelist or blacklist for the matched incoming SIP dialog. 
- **[0]** Deny = Blocks incoming SIP dialogs that match the characteristics of the rule (blacklist).
- **[1]** Allow = (Default) Allows incoming SIP dialogs that match the characteristics of the rule (whitelist) and assigns it to the associated IP Group. |
| dest-routing-policy | Assigns a Routing Policy to the matched incoming SIP dialog. The assigned Routing Policy overrides the Routing Policy assigned to the SRD (in the SRDs table). The option to assign Routing Policies to Classification rules is useful in deployments requiring different routing and manipulation rules for specific calls pertaining to the same SRD. In such scenarios, you need to configure multiple Classification rules for the same SRD, where for some rules no Routing Policy is assigned (i.e., the SRD's assigned Routing Policy is used) while for others a different Routing Policy is specified to override the SRD's assigned Routing Policy. By default, no value is defined. To configure Routing Policies, see "Configuring SBC Routing Policy Rules" on page 521. |
| src-ip-group-name | Assigns an IP Group to the matched incoming SIP dialog. By default, no value is defined. To configure IP Groups, see "Configuring IP Groups" on page 345. **Note:** The IP Group must be associated with the assigned SRD (see the ‘SRD’ parameter in the table). |
| ip-profile-id | Assigns an IP Profile to the matched incoming SIP dialog. The assigned IP Profile overrides the IP Profile assigned to the IP Group (in the IP Groups table) to which the SIP dialog is classified. |
Parameter | Description
---|---
<Classification_IpProfileName> | Therefore, assigning an IP Profile during classification allows you to assign different IP Profiles to specific users (calls) that belong to the same IP Group (User or Server type).

For example, you can configure two Classification rules to classify incoming calls to the same IP Group. However, one Classification rule is a regular rule that doesn't specify any IP Profile (IP Profile assigned to IP Group is used), while the second rule is configured with an additional matching characteristic for the source hostname prefix (e.g., "abcd.com") and with an additional action that assigns a different IP Profile.

By default, no value is defined.

**Note:** For User-type IP Groups, if a user is already registered with the device (from a previous, initial classification process), the device classifies subsequent INVITE requests from the user according to the device's users database instead of the Classification table. In such a scenario, the same IP Profile that was previously assigned to the user by the Classification table is also used (in other words, the device's users database stores the associated IP Profile).

### 27.1.1 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. The 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=z9hG4bKVEBTDAHSUYRTXEDGEJY
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>,<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:

<table>
<thead>
<tr>
<th>Index</th>
<th>Source Username Prefix</th>
<th>Destination Username Prefix</th>
<th>Destination Host</th>
<th>Source IP Group</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>333</td>
<td>-</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1111</td>
<td>2000</td>
<td>10.10.10.10</td>
<td>2</td>
</tr>
</tbody>
</table>

2. In the IP Groups table, add the following IP Groups:

<table>
<thead>
<tr>
<th>Index</th>
<th>Source URI Input</th>
<th>Destination URI Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
In the example, a match exists only for Classification Rule #1. This is because the source (1111) and destination (2000) username prefixes match those in the INVITE’s P-Called-Party-ID header (i.e., "<sip:1111@10.33.38.1>") and Route header (i.e., "<sip:2000@10.10.10.10>"), respectively. These SIP headers were determined in IP Group 2.

27.2 Configuring SBC IP-to-IP Routing

The IP-to-IP Routing table lets you configure up to 9,000 SBC IP-to-IP routing rules.

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

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

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

**Note**: Configure stricter rules higher up in the table than less strict rules to ensure the desired rule is used to route the call. _Strict_ refers to the number of matching characteristics configured for the rule. For example, a rule configured with source host name and source IP Group as matching characteristics is stricter than a rule configured with only source host name. If the rule configured with only source host name appears higher up in the table, the device ("erroneously") uses the rule to route calls matching this source host name (even if they also match the rule appearing lower down in the table configured with the source IP Group as well).

You can route incoming SIP dialog messages (e.g., INVITE) to any of the following IP destinations:

- According to registered user Contact listed in the device's registration database (only for User-type IP Groups).
- IP Group - the destination is the address configured for the Proxy Set associated with the IP Group.
- IP Group Set - the destination can be based on multiple IP Groups for load balancing, where each call may be sent to a different IP Group within the IP Group Set depending on the IP Group Set's definition.
- Routing tag - the device sends the call to an IP Group (or IP Group Set) based on a destination Dial Plan tag that corresponds to the destination (called) prefix number.
- IP address in dotted-decimal notation or FQDN. Routing to a host name can be resolved using NAPTR/SRV/A-Record.
- Request-URI of incoming SIP dialog-initiating requests.
- Any registered user in the registration database. If the Request-URI of the incoming INVITE exists in the database, the call is sent to the corresponding contact address specified in the database.
According to result of an ENUM query.

Hunt Group - used for call survivability of call centers (see "Configuring Call Survivability for Call Centers" on page 577).

According to result of LDAP query (for more information on LDAP-based routing, see "Routing Based on LDAP Active Directory Queries" on page 235).

Third-party routing server, which determines the destination (next hop) of the call (IP Group). The IP Group represents the next device in the routing path to the final destination. For more information, see "Centralized Third-Party Routing Server" on page 275.

Back to the sender of the incoming message, where the reply can be a SIP response code or a 3xx redirect response (with an optional Contact field to where the sender must re-send the message).

The following figure summarizes the destination types:

Figure 27-4: IP-to-IP Routing Destination Types

To configure and apply an IP-to-IP Routing rule, the rule must be associated with a Routing Policy. The Routing Policy associates the routing rule with an SRD(s). Therefore, the Routing Policy lets you configure routing rules for calls belonging to specific SRD(s). However, as multiple Routing Policies are relevant only for multi-tenant deployments (if needed), for most deployments, only a single Routing Policy is required. As the device provides a default Routing Policy ("Default_SBCRoutingPolicy"), when only one Routing Policy is required, the device automatically assigns the default Routing Policy to the routing rule. If you are implementing LDAP-based routing (with or without Call Setup Rules) and/or Least Cost Routing (LCR), you need to configure these settings for the Routing Policy.
27. Routing SBC

(regardless of the number of Routing Policies employed). For more information on Routing Policies, see “Configuring SBC Routing Policy Rules” on page 521.

The IP-to-IP Routing table also provides the following features:

- **Alternative Routing:** 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 Alternative Routing Reasons table (see "Configuring SIP Response Codes for Alternative Routing Reasons" on page 519).
    - SIP 408 Timeout or no response (after timeout).
  - The DNS resolution includes IP addresses that the device has yet to try (for the current call).

Messages are re-routed with the same SIP Call-ID and CSeq header fields (increased by 1).

**Note:** If the Proxy Set (see Configuring Proxy Sets on page 359) associated with the destination of the call is configured with multiple IP addresses, the device first attempts to route the call to one of these IP addresses, starting with the first listed address. Only when the call cannot be routed to any of the Proxy Set’s IP addresses does the device search the IP-to-IP Routing table for an alternative routing rule for the call.

- **Re-routing 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).

- **Load Balancing:** You can implement load balancing of calls, belonging to the same source, between a set of destination IP Groups known as an **IP Group Set**. The IP Group Set can include up to five IP Groups (Server-type and/or Gateway-type only) and the chosen IP Group depends on the configured load-balancing policy (e.g., Round Robin). To configure the feature, you need to first configure an IP Group Set (see Configuring IP Group Sets on page 525), and then assign it to a routing rule with 'Destination Type' configured to **IP Group Set**.

- **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. To configure Cost Groups, see "Least Cost Routing" on page 263. If two routing rules have identical costs, then the rule appearing higher up in the table (i.e., first-matched rule) is used. If a selected route is unavailable, the device uses the next least-cost routing rule. However, even if a matched rule is not assigned a Cost Group, the device can select it as the preferred route over other matched routing rules that are assigned Cost Groups, according to the default LCR settings configured for the assigned Routing Policy (see "Configuring SBC Routing Policy Rules" on page 521).

- **Call Forking:** The IP-to-IP Routing table can be configured to route an incoming IP call to multiple destinations (call forking). The incoming call can be routed to multiple destinations of any type such as an IP Group or IP address. The device forks the call by sending simultaneous INVITE messages to all the specified destinations. It handles
the multiple SIP dialogs until one of the calls is answered and then terminates the other SIP dialogs.

Call forking is configured by creating a Forking group. A Forking group consists of a main routing rule (‘Alternative Route Options’ set to Route Row) whose ‘Group Policy’ is set to Forking, and one or more associated routing rules (‘Alternative Route Options’ set to Group Member Ignore Inputs or Group Member Consider Inputs). The group members must be configured in contiguous table rows to the main routing rule. If an incoming call matches the input characteristics of the main routing rule, the device routes the call to its destination and all those of the group members.

An alternative routing rule can also be configured for the Forking group. The alternative route is used if the call fails for the Forking group (i.e., main route and all its group members). The alternative routing rule must be configured in the table row immediately below the last member of the Forking group. The 'Alternative Route Options' of this alternative route must be set to Alt Route Ignore Inputs or Alt Route Consider Inputs. The alternative route can also be configured with its own forking group members, where if the device uses the alternative route, the call is also sent to its group members. In this case, instead of setting the alternative route’s 'Group Policy' to None, you must set it to Forking. The group members of the alternative route must be configured in the rows immediately below it.

The LCR feature can also be employed with call forking. The device calculates a maximum call cost for each Forking group and routes the call to the Forking group with the lowest cost. Thus, even if the call can successfully be routed to the main routing rule, a different routing rule can be chosen (even an alternative route, if configured) based on LCR. If routing to one Forking group fails, the device tries to route the call to the Forking group with the next lowest cost (main or alternative route), and so on. The prerequisite for this functionality is that the incoming call must successfully match the input characteristics of the main routing rule.

- **Dial Plan Tags for Representing Source / Destination Numbers**: If your deployment includes calls of many different called (source URI user name) and/or calling (destination URI user name) numbers that need to be routed to the same destination, you can employ user-defined tags to represent these numbers. Thus, instead of configuring many routing rules, you can configure only one routing rule using the tag as the source and destination number matching characteristics, and a destination for the calls. For more information, see Using Dial Plan Tags for Matching Routing Rules on page 541.

- **Dial Plan Tags for Determining Destination IP Group**: Instead of configuring multiple routing rules, you can configure a single routing rule with a specific "destination" Dial Plan tag. The device uses the tag to determine the destination IP Group. For more information, see Using Dial Plan Tags for Routing Destinations on page 551.

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

The following procedure describes how to configure IP-to-IP routing rules through the Web interface. You can also configure it through ini file (IP2IPRouting) or CLI (configure voip > sbc routing ip2ip-routing).

- **To configure an IP-to-IP routing rule**:
  1. Open the IP-to-IP Routing table (Setup menu > Signaling & Media tab > SBC folder > Routing > IP-to-IP Routing).
2. Click **New**; the following dialog box appears:

![Figure 27-5: IP-to-IP Routing Table - Add Dialog Box]

3. Configure an IP-to-IP routing rule according to the parameters described in the table below.

4. Click **Apply**.

Table 27-2: IP-to-IP Routing Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Routing Policy</strong></td>
<td>Assigns a Routing Policy to the rule. The Routing Policy associates the rule with an SRD(s). The Routing Policy also defines default LCR settings as well as the LDAP servers used if the routing rule is based on LDAP routing (and Call Setup Rules). If only one Routing Policy is configured in the Routing Policies table, the Routing Policy is automatically assigned. If multiple Routing Policies are configured, no value is assigned. To configure Routing Policies, see “Configuring SBC Routing Policy Rules” on page 521. <strong>Note</strong>: The parameter is mandatory.</td>
</tr>
<tr>
<td><code>sbc-routing-policy-name</code></td>
<td><code>[IP2IPRouting_RoutingPolicyName]</code></td>
</tr>
<tr>
<td><code>General</code></td>
<td></td>
</tr>
<tr>
<td><code>Index</code></td>
<td>Defines an index number for the new table row. <strong>Note</strong>: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td><code>Name</code></td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. By default, no value is defined.</td>
</tr>
<tr>
<td><code>Alternative Route Options</code></td>
<td>Determines whether this routing rule is the main routing rule or an alternative routing rule (to the rule defined directly above it in the routing rules table).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>[IP2IPRouting_AltRouteOptions]</td>
<td>The table).</td>
</tr>
<tr>
<td></td>
<td>• [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.</td>
</tr>
<tr>
<td></td>
<td>• [1] Alternative Route Ignore Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route regardless of the incoming SIP dialog's input characteristics.</td>
</tr>
<tr>
<td></td>
<td>• [2] Alternative Route Consider Inputs = If the call cannot be routed to the main route (Route Row), the call is routed to this alternative route only if the incoming SIP dialog matches this routing rule's input characteristics.</td>
</tr>
<tr>
<td></td>
<td>• [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.</td>
</tr>
<tr>
<td></td>
<td>• [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.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>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 519). 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 Alternative Routing Reasons table.</td>
</tr>
<tr>
<td></td>
<td>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).</td>
</tr>
</tbody>
</table>

**Match**

| Source IP Group src-ip-group-name [IP2IPRouting_SrcIPGroupName] | Defines the IP Group from where the IP call is received (i.e., the IP Group that sent the SIP dialog). Typically, the IP Group of an incoming SIP dialog is determined (or classified) using the Classification table (see “Configuring Classification Rules” on page 493). The default is Any (i.e., any IP Group). Note: The selectable IP Group for the parameter depends on the assigned Routing Policy (in the ‘Routing Policy’ parameter in this table). For more information, see “Configuring SBC Routing Policy Rules” on page 521. |

| Request Type | Defines the SIP dialog request type (SIP Method) of the |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| request-type [IP2IPRouting_RequestType] | incoming SIP dialog.  
- [0] All (default)  
- [1] INVITE  
- [2] REGISTER  
- [3] SUBSCRIBE  
- [4] INVITE and REGISTER  
- [5] INVITE and SUBSCRIBE  
- [6] OPTIONS |

| Source Username Prefix [IP2IPRouting_SrcUsernamePrefix] | Defines the prefix of the user part of the incoming SIP dialog's source URI (usually the From URI). You can use special notations for denoting the prefix. To denote calls without a user part in the URI, use the $ sign. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 827.  
The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty.  
**Note:** If you need to route calls of many different source URI user names to the same destination, you can use tags (see 'Source Tags' parameter below) instead of this parameter. |

| Source Host [IP2IPRouting_SrcHost] | Defines the host part of the incoming SIP dialog's source URI (usually the From URI).  
The default is the asterisk (*) symbol (i.e., any host name). If this rule is not required, leave this field empty. |

| Source Tags [IP2IPRouting_SrcTags] | Assigns a tag to denote source URI user names corresponding to the tag configured in the associated Dial Plan.  
The valid value is a string of up to 20 characters. The tag is case insensitive.  
To configure tags, see "Configuring Dial Plans" on page 541.  
**Note:**  
- Make sure that you assign the Dial Plan in which you have configured the tag, to the related IP Group or SRD.  
- Instead of using tags and configuring the parameter, you can use the 'Source Username Prefix' parameter to specify a specific URI source user or all source users. |

| Destination Username Prefix [IP2IPRouting_DestUsernamePrefix] | Defines the prefix of the incoming SIP dialog's destination URI (usually the Request URI) user part. You can use special notations for denoting the prefix. To denote calls without a user part in the URI, use the $ sign. For available notations, see "Dialing Plan Notation for Routing and Manipulation" on page 827.  
The default is the asterisk (*) symbol (i.e., any prefix). If this rule is not required, leave this field empty.  
**Note:** If you need to route calls of many different destination URI user names to the same destination, you can use tags (see 'Source Tags' parameter below) instead of this parameter. |

| Destination Host [IP2IPRouting_DestHost] | Defines the host part of the incoming SIP dialog’s destination URI (usually the Request-URI).  
The default is the asterisk (*) symbol (i.e., any destination host). |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Destination Tags</strong>&lt;br&gt;<strong>dest-tags</strong>&lt;br&gt;[IP2IPRouting_DestTags]</td>
<td>Assigns a prefix tag to denote destination URI user names corresponding to the tag configured in the associated Dial Plan. The valid value is a string of up to 20 characters. The tag is case insensitive. To configure prefix tags, see &quot;Configuring Dial Plans&quot; on page 541. <strong>Note:</strong>&lt;br&gt;• Make sure that you assign the Dial Plan in which you have configured the prefix tag, to the related IP Group or SRD. • Instead of using tags and configuring the parameter, you can use the 'Destination Username Prefix' parameter to specify a specific URI destination user or all destinations users.</td>
</tr>
<tr>
<td><strong>Message Condition</strong>&lt;br&gt;<strong>message-condition-name</strong>&lt;br&gt;[IP2IPRouting_MessageConditionName]</td>
<td>Assigns a SIP Message Condition rule to the IP-to-IP Routing rule. To configure Message Condition rules, see &quot;Configuring Message Condition Rules&quot; on page 501.</td>
</tr>
<tr>
<td><strong>Call Trigger</strong>&lt;br&gt;<strong>trigger</strong>&lt;br&gt;[IP2IPRouting_Trigger]</td>
<td>Defines the reason (i.e., trigger) for re-routing (i.e., alternative routing) the SIP request:&lt;br&gt;• [0] Any = (Default) This routing rule is used for all scenarios (re-routes and non-re-routes).&lt;br&gt;• [1] 3xx = Re-routes the request if it was triggered as a result of a SIP 3xx response.&lt;br&gt;• [2] REFER = Re-routes the INVITE if it was triggered as a result of a REFER request.&lt;br&gt;• [3] 3xx or REFER = Applies to options [1] and [2].&lt;br&gt;• [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.&lt;br&gt;• [5] Broken Connection = If the device detects a broken RTP connection during the call and the Broken RTP Connection feature is enabled ([IpProfile_DisconnectOnBrokenConnection parameter is configured to [2]]), you can use this option as an explicit matching characteristics to route the call to an alternative destination. Therefore, for alternative routing upon broken RTP detection, position the routing rule configured with this option above the regular routing rule associated with the call. Such a configuration setup ensures that the device uses this alternative routing rule only when RTP broken connection is detected.&lt;br&gt;• [6] Fax Rerouting = Reroutes the INVITE to a fax destination (different IP Group) if it is identified as a fax call. For more information, see Configuring Rerouting of Calls to Fax Destinations on page 513.</td>
</tr>
</tbody>
</table>
| **ReRoute IP Group**<br>**re-route-ip-group-id**<br>[IP2IPRouting_ReRouteIPGroupName] | Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. This parameter is typically used for re-routing requests (e.g., INVITEs) when interworking is required for SIP 3xx redirect responses or REFER messages. For more information, see "Interworking SIP..."
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3xx Redirect Responses</td>
<td>on page 476 and &quot;Interworking SIP REFER Messages&quot; on page 478, respectively. The parameter functions together with the 'Call Trigger' parameter (in the table). The default is Any (i.e., any IP Group). Note: The selectable IP Group for the parameter depends on the assigned Routing Policy (in the 'Routing Policy' parameter in this table). For more information, see &quot;Configuring SBC Routing Policy Rules&quot; on page 521.</td>
</tr>
</tbody>
</table>

**Action**

**Destination Type**

**dst-type [IP2IPRouting_DestType]**

Determines the destination type to which the outgoing SIP dialog is sent.

- **[0]** IP Group = (Default) The SIP dialog is sent to the IP Group as defined in the 'Destination IP Group' (IP2IPRouting_DestIPGroupName) parameter. For more information on the actual address, see the 'Destination IP Group' parameter.
- **[1]** Dest Address = The SIP dialog is sent to the address configured in the following parameters: 'Destination Address', 'Destination Port' and 'Destination Transport Type'.
- **[2]** Request URI = The SIP dialog is sent to the address indicated in the incoming Request-URI. If the parameters 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these parameters take precedence.
- **[3]** ENUM = An ENUM query is sent to include the destination address. If the parameters 'Destination Port' and 'Destination Transport Type' are configured, the incoming Request-URI parameters are overridden and these parameters take precedence.
- **[4]** Hunt Group = Used for call center survivability. For more information, see "Configuring Call Survivability for Call Centers" on page 577.
- **[5]** Dial Plan = (For Backward Compatibility Only - see Note below) 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. Make sure that the Routing Policy assigned to the routing rule is configured with the
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Server Group</td>
<td>for defining the LDAP server(s) to query.</td>
</tr>
<tr>
<td>[9] Routing Server</td>
<td>Device sends a request to a third-party routing server for an appropriate destination (next hop) for the matching call.</td>
</tr>
<tr>
<td>[10] All Users</td>
<td>Device checks whether the Request-URI (i.e., destination user) in the incoming INVITE is registered in its' users' database, and if yes, it sends the INVITE to the address of the corresponding contact specified in the database. If the Request-URI is not registered, the call is rejected.</td>
</tr>
<tr>
<td>[11] IP Group Set</td>
<td>The device employs load balancing and routes the call to one of the IP Groups in the IP Group Set, assigned using the 'IP Group Set' parameter (below).</td>
</tr>
<tr>
<td>[12] Destination Tag</td>
<td>The device routes the call to an IP Group determined by Dial Plan tags. The tag is specified in the 'Routing Tag Name' parameter (below). For more information, see Using Dial Plan Tags for Routing Destinations on page 551.</td>
</tr>
<tr>
<td>[13] Internal</td>
<td>Instead of sending the incoming SIP dialog to another destination, the device replies to the sender of the dialog with a SIP response code or a redirection response, configured by the 'Internal Action' (IP2IPRouting_InternalAction) parameter in this table (see below).</td>
</tr>
</tbody>
</table>

**Note:** Use option [5] Dial Plan only for backward compatibility purposes; otherwise, use prefix tags as described in "Configuring Dial Plans" on page 541.

<table>
<thead>
<tr>
<th>Destination IP Group</th>
<th>dst-ip-group-name</th>
<th>Defines the IP Group to where you want to route the call. The actual destination of the SIP dialog message depends on the IP Group type (as defined in the 'Type' parameter):</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[IP2IPRouting_DestIPGroupName]</td>
<td>• Server-type IP Group: The SIP dialog is sent to the IP address configured for the Proxy Set that is associated with the IP Group.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• User-type IP Group: The device checks if the SIP dialog is from a registered user, by searching for a match between the Request-URI of the received SIP dialog and an AOR registration record in the device's database. If found, the device sends the SIP dialog to the IP address specified in the database for the registered contact.</td>
</tr>
<tr>
<td>By default, no value is defined.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
- The parameter is applicable only if the 'Destination Type' parameter is configured to IP Group.
- The selectable IP Group for the parameter depends on the assigned Routing Policy (in the 'Routing Policy' parameter in this table). For more information, see "Configuring SBC Routing Policy Rules" on page 521.

<table>
<thead>
<tr>
<th>Destination SIP Interface</th>
<th>dest-sip-interface-name</th>
<th>Defines the destination SIP Interface to where the call is sent. By default, no value is defined.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[IP2IPRouting_DestSIPInterfaceName]</td>
<td>To configure SIP Interfaces, see &quot;Configuring SIP Interfaces&quot; on page 337.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>The parameter is applicable only if the 'Destination Type' parameter is configured to any value other than IP Group. If the 'Destination Type' parameter is configured to IP Group, the following SIP Interface is used:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ Server-type IP Groups: SIP Interface that is assigned to the Proxy Set associated with the IP Group.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ User-type IP Groups: SIP Interface is determined during user registration with the device.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• For multi-tenancy, if the assigned Routing Policy is not shared (i.e., the Routing Policy is associated with an Isolated SRD), the SIP Interface must be one that is associated with the Routing Policy or with a shared Routing Policy (i.e., the Routing Policy is associated with one or more Shared SRDs). If the Routing Policy is shared, the SIP Interface can be one that is associated with any SRD or Routing Policy (but it's recommended that it belong to the same SRD/Routing Policy to avoid &quot;bleeding&quot;).</td>
<td></td>
</tr>
<tr>
<td><strong>Destination Address</strong></td>
<td>Defines the destination address to where the call is sent. The address can be an IP address or a domain name (e.g., domain.com).</td>
<td></td>
</tr>
<tr>
<td>dst-address</td>
<td>If ENUM-based routing is used (i.e., the 'Destination Type' parameter is set to ENUM) the parameter defines the IP address or domain name (FQDN) of the ENUM service, for example, e164.arpa, e164.customer.net or NRENum.net. The device sends the ENUM query containing the destination phone number to an external DNS server, configured in the IP Interfaces table. The ENUM reply includes a SIP URI (user@host) which is used as the destination Request-URI in this routing table. The valid value is a string of up to 50 characters (IP address or FQDN). By default, no value is defined. <strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td>[IP2IPRouting_DestAddress]</td>
<td>• The parameter is applicable only if the 'Destination Type' parameter is set to Dest Address [1] or ENUM [3]; otherwise, the parameter is ignored.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• When using domain names, enter a DNS server IP address or alternatively, define these names in the Internal DNS table (see &quot;Configuring the Internal SRV Table&quot; on page 160).</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• To terminate SIP OPTIONS messages at the device (i.e., to handle them locally), set the parameter to &quot;internal&quot;.</td>
<td></td>
</tr>
<tr>
<td><strong>Destination Port</strong></td>
<td>Defines the destination port to where the call is sent.</td>
<td></td>
</tr>
<tr>
<td>dst-port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[IP2IPRouting_DestPort]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Destination Transport Type</strong></td>
<td>Defines the transport layer type for sending the call:</td>
<td></td>
</tr>
<tr>
<td>dst-transport-type</td>
<td>• [-1] = (Default) Not configured - the transport type is determined by the SIPTransportType global parameter.</td>
<td></td>
</tr>
<tr>
<td>[IP2IPRouting_DestTransportType]</td>
<td>• [0] UDP</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• [1] TCP</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>IP Group Set</strong>&lt;br&gt;ipgroupset-name&lt;br&gt;[IP2IPRouting_IPGroupSetName]</td>
<td>Assigns an IP Group Set to the routing rule. The device routes the call to one of the IP Groups in the IP Group Set according to the load-balancing policy configured for the IP Group Set. For more information, see Configuring IP Group Sets on page 525. <strong>Note:</strong> The parameter is applicable only if you configure the 'Destination Type' parameter to <strong>IP Group Set</strong> (above).</td>
<td></td>
</tr>
<tr>
<td><strong>Call Setup Rules Set ID</strong>&lt;br&gt;call-setup-rules-set-id&lt;br&gt;[IP2IPRouting_CallSetupRulesSetId]</td>
<td>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. To configure Call Setup rules, see &quot;Configuring Call Setup Rules&quot; on page 405.</td>
<td></td>
</tr>
</tbody>
</table>
| **Group Policy**<br>group-policy<br>[IP2IPRouting_GroupPolicy] | Defines whether the routing rule includes call forking.  
  - **[0]** None = (Default) Call uses only this route (even if Forking Group members are configured in the rows below it).  
  - **[1]** Forking = Call uses this route and the routes of Forking Group members, if configured (in the rows below it).  
  **Note:** Each Forking Group can contain up to 20 members. In other words, up to 20 routing rules can be configured for the same Forking Group. |
| **Cost Group**<br>cost-group<br>[IP2IPRouting_CostGroup] | Assigns a Cost Group to the routing rule for determining the cost of the call. By default, no value is defined. To configure Cost Groups, see "Configuring Cost Groups" on page 265. **Note:**  
  - To implement LCR and its Cost Groups, you must enable LCR for the Routing Policy assigned to the routing rule (see "Configuring SBC Routing Policy Rules" on page 521). If LCR is disabled, the device ignores the parameter.  
  - The Routing Policy also determines whether matched routing rules that are not assigned Cost Groups are considered as a higher or lower cost route compared to matching routing rules that are assigned Cost Groups. For example, if the 'Default Call Cost' parameter in the Routing Policy is configured to **Lowest Cost**, even if the device locates matching routing rules that are assigned Cost Groups, the first-matched routing rule without an assigned Cost Group is considered as the lowest cost route and thus, chosen as the preferred route. |
| **Routing Tag Name**<br>routing-tag-name<br>[IP2IPRouting_RoutingTagName] | Defines the destination Dial Plan tag, which is used to determine the destination IP Group. The default value is "default", meaning that the device uses the first tag name in the Dial Plan rule that is configured without a value. For example, if the Dial Plan rule is configured with tags "Country=England;City=London;Essex", the default tag is "Essex". For more information, see Using Dial Plan Tags for Routing |
### 27. Routing SBC

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Destinations** | Destinations on page page 551. **Note:**  
  - The parameter is applicable only if the 'Destination Type' parameter is configured to **Destination Tag** (see above).  
  - Only one tag can be configured for the parameter.  
  - Only the tag name must be configured (not the value, if exists). For example, if the tag is configured in the Dial Plan rule as "Country=England", configure the parameter to "Country" only. |

**Internal Action**

| Internal Action | Defines a SIP response code (e.g., 200 OK) or a redirection response (with an optional Contact field indicating to where the sender must re-send the message) that the device sends to the sender of the incoming SIP dialog (instead of sending the call to another destination). The parameter is applicable only when the 'Destination Type' parameter in this table is configured to Internal (see above). The valid value syntax is (case-insensitive):  
  - For SIP response codes:  
    ```
    Reply(response='<code>')
    ```  
    The following example sends a SIP 200:  
    ```
    Reply(response='200')
    ```  
  - For redirection responses:  
    ```
    Redirect (response='<code>', contact='sip:'+...)
    Redirect (contact='...', response = '<code>')
    Redirect (contact = 'sip:user@host')
    ```  
    **Examples:**  
    - The device responds to the dialog with a SIP 300 redirect response that includes a contact value:  
      ```
      Redirect (response='300', contact='sip:102@host')
      ```  
    - The device redirects the call from the sender to a SIP Recording Server (SRS):  
      ```
      Redirect(response='302', contact='sip:'+header .to.url.user+'@siprecording.com')
      ```  
  **Note:**  
  - The parameter can be used for normal and alternative routing.  
  - The response code for redirect messages can only be 3xx. |

#### 27.2.1 Configuring Rerouting of Calls to Fax Destinations

You can configure the device to reroute incoming SBC calls identified as fax calls to a new IP destination. The device identifies a fax call if it detects, within a user-defined interval, a calling (CNG) tone on the originator side (incoming IP leg). If the device detects a fax call, it terminates the call and reroutes it using a new INVITE to the new fax destination (new IP Group). If the initial INVITE that was used to establish the voice call was already sent, the device sends a CANCEL (if not connected yet) or a BYE (if already connected) to release...
the call (with the internal disconnect reason RELEASE_BECAUSE_FAX_REROUTING, translated to Q.850 reason GWAPP_NORMAL_UNSPECIFIED 31).

Note:
- You must configure the originating fax to use the G.711 coder.
- If the remote side replies with T.38 or G.711 VBD, fax rerouting is not done.
- If both fax rerouting and fax re-INVITE are configured, only fax rerouting is done.

The following procedure describes how to configure fax rerouting, using an example consisting of three IP Groups.

➢ To configure fax rerouting:

1. Configure the fax detection interval:
   a. Open the Fax/Modem/CID Settings page (Setup menu > Signaling & Media tab > Media folder > Fax/Modem/CID Settings).
   b. In the 'Fax Detection Timeout' field (SBCFaxDetectionTimeout), enter the duration (in seconds) for which the device attempts to detect fax (CNG tone):

2. Configure three IP Groups in the IP Groups table (see Configuring IP Groups on page 345):
   - IP Group #0 "HQ": This is the source IP Group, sending voice calls and fax calls.
   - IP Group #1 "Voice": This is the destination for voice calls sent from IP Group #0.
   - IP Group #2 "Fax": This is the destination for fax calls sent from IP Group #0.

3. For the fax destination (IP Group #2), do the following:
   a. Configure a Coder Group with T.38, which allows fax transmission over IP (see Configuring Coder Groups on page 407).
   b. Configure an IP Profile (see Configuring IP Profiles on page 417):
      - Assign it the Coder Group.
      - Configure the 'Fax Rerouting Mode' parameter to Rerouting without delay:

   c. Assign the IP Profile to IP Group#2.

   - IP-to-IP Routing Rule #0 - routes voice calls from IP Group #0 to IP Group #1:

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP Group</td>
<td>HQ (IP Group #0)</td>
</tr>
<tr>
<td>Call Trigger</td>
<td>Initial Only</td>
</tr>
<tr>
<td>ReRoute IP Group</td>
<td>Voice (IP Group #1)</td>
</tr>
</tbody>
</table>

   - Destination Type | IP Group |
   - Destination IP Group | Voice (IP Group #1) |

   - IP-to-IP Routing Rule #1 - routes fax calls from IP Group #0 to IP Group #2:
### 27.2.2 Configuring Specific UDP Ports using Tag-based Routing

You can configure the device to use a specific local UDP port for each SIP entity (e.g., PBX) communicating with a common proxy server (e.g., ITSP). The figure below illustrates an example scenario of such an implementation, whereby the device uses a specific local UDP port (e.g., 6001, 6002, and 6003) for each IP PBX, on the leg interfacing with the proxy server:

**Figure 27-6: Example of Assigning Specific Ports using Tags**

For each IP PBX, the device sends SIP messages to the proxy server using the specific local, UDP port on the leg interfacing with the proxy server. For SIP messages received from the proxy server, the device routes the messages to the appropriate IP PBX according to the local UDP port on which the message was received. On the leg interfacing with the IP PBXs, the device uses the same local UDP port (e.g., 5060) for all IP PBXs (send and receive).

To configure this feature, you need to configure the SIP Interface of the proxy server with a special UDP port range, and use tag-based routing with Call Setup Rules to specify the exact UDP port you want assigned to each SIP entity (IP PBX), from the SIP Interface port range. The following procedure describes how to configure the device to use a specific local UDP port per SIP entity on the leg interfacing with a proxy server that is common to all the SIP entities. To facilitate understanding, the procedure is based on the previous example.

- **To configure specific UDP ports for SIP entities communicating with common proxy server:**
  1. Open the SIP Interfaces table (see Configuring SIP Interfaces on page 337), and then configure the following SIP Interfaces:
     - SIP Interface for leg interfacing with IP PBXs (local UDP port 5060 is used):

#### Table 27-2: SIP Interface configuration for specific UDP ports

<table>
<thead>
<tr>
<th>Match</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source IP Group</td>
<td>HQ (IP Group #0)</td>
</tr>
<tr>
<td>Call Trigger</td>
<td>Fax Rerouting</td>
</tr>
<tr>
<td>Destination Type</td>
<td>IP Group</td>
</tr>
<tr>
<td>Destination IP Group</td>
<td>Fax (IP Group #2)</td>
</tr>
</tbody>
</table>

- Open the SIP Interfaces table (see Configuring SIP Interfaces on page 337), and then configure the following SIP Interfaces:
  - SIP Interface for leg interfacing with IP PBXs (local UDP port 5060 is used):
• SIP Interface for leg interfacing with proxy server (specific local UDP ports are later taken from this port range):

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>2</td>
</tr>
<tr>
<td>Name</td>
<td>ITSP</td>
</tr>
<tr>
<td>Network Interface</td>
<td>LAN</td>
</tr>
<tr>
<td>UDP Port</td>
<td>5060</td>
</tr>
<tr>
<td>Additional UDP Ports</td>
<td>6000-7000</td>
</tr>
</tbody>
</table>

Note: For guidelines on configuring the 'Additional UDP Ports’ parameter (SIPInterface_AdditionalUDPPorts), see Configuring SIP Interfaces on page 337.

2. Open the IP Groups table (see Configuring IP Groups on page 345), and then configure the following IP Groups:

• IP Group for the first IP PBX ("Type" and "Port" tags are later used to identify the IP PBX and assign it a local UDP port 6001 on the leg interfacing with the proxy server):

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>1</td>
</tr>
<tr>
<td>Name</td>
<td>PBX-1</td>
</tr>
<tr>
<td>Type</td>
<td>Server</td>
</tr>
<tr>
<td>SBC Advanced</td>
<td></td>
</tr>
<tr>
<td>Tags</td>
<td>Type=PBX;Port=6001</td>
</tr>
</tbody>
</table>

• IP Group for the second IP PBX ("Type" and "Port" tags are later used to identify the IP PBX and assign it a local UDP port 6002 on the leg interfacing with the proxy server):

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>2</td>
</tr>
<tr>
<td>Name</td>
<td>PBX-2</td>
</tr>
<tr>
<td>Type</td>
<td>Server</td>
</tr>
<tr>
<td>SBC Advanced</td>
<td></td>
</tr>
<tr>
<td>Tags</td>
<td>Type=PBX;Port=6002</td>
</tr>
</tbody>
</table>
27. Routing SBC

• IP Group for the third IP PBX ("Type" and "Port" tags are later used to identify the IP PBX and assign it a local UDP port 6003 on the leg interfacing with the proxy server):

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>3</td>
</tr>
<tr>
<td>Name</td>
<td>PBX-3</td>
</tr>
<tr>
<td>Type</td>
<td>Server</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SBC Advanced</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tags</td>
<td>Type=PBX;Port=6003</td>
</tr>
</tbody>
</table>

• IP Group for the proxy server ("Type" tag is later used to identify proxy server):

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>4</td>
</tr>
<tr>
<td>Name</td>
<td>ITSP</td>
</tr>
<tr>
<td>Type</td>
<td>Server</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SBC Advanced</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tags</td>
<td>Type=ITSP</td>
</tr>
</tbody>
</table>

3. Open the Call Setup Rules table (see Configuring Call Setup Rules on page 384), and then configure the following Call Setup rules:

• Uses the value of the "Type" tag name, configured in the IP Group’s 'Tags' parameter, as the source tag:

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>1</td>
</tr>
<tr>
<td>Rule Set ID</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Subject</td>
<td>srctags.Type</td>
</tr>
<tr>
<td>Action Type</td>
<td>Modify</td>
</tr>
<tr>
<td>Action Value</td>
<td>param.ipg.src.tags.Type</td>
</tr>
</tbody>
</table>

• If the source tag name "Type" equals "PBX" (i.e., SIP message from an IP Group belonging to one of the IP PBXs), then use the value of the "Port" tag name, configured in the 'Tags' parameter of the classified IP Group, as the local UDP port on the leg interfacing with the proxy server for messages sent to the proxy server:

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>2</td>
</tr>
<tr>
<td>Rule Set ID</td>
<td>1</td>
</tr>
<tr>
<td>Condition</td>
<td>srctags.Type=='PBX'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th></th>
</tr>
</thead>
</table>
• If the source tag name "Type" equals "ITSP" (i.e., SIP message from the ITSP), then use the value (port number) of the local port on which the incoming message from the proxy server is received by the device, as the value of the destination tag name "Port". In other words, the value could either be "6001", "6002", or "6003". This value is then used by the IP-to-IP Routing table to determine to which IP PBX to send the message. For example, if the destination tag value is "6001", the device identifies the destination as "PBX-1".

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>3</td>
</tr>
<tr>
<td>Rule Set ID</td>
<td>1</td>
</tr>
<tr>
<td>Condition</td>
<td>srctags.Type=='ITSP'</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Subject</td>
<td>dsttags.Port</td>
</tr>
<tr>
<td>Action Type</td>
<td>Modify</td>
</tr>
<tr>
<td>Action Value</td>
<td>message.incoming.local-port</td>
</tr>
</tbody>
</table>

4. Open the IP-to-IP Routing table (see Configuring SBC IP-to-IP Routing Rules on page 501), and then configure the following IP-to-IP Routing rules:

• Routes calls from the IP PBXs (identified by the source tag name-value "Type=PBX") to the ITSP (identified as an IP Group):

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>1</td>
</tr>
<tr>
<td>Name</td>
<td>PBX-to-ITSP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Match</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Tag</td>
<td>Type=PBX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Type</td>
<td>IP Group</td>
</tr>
<tr>
<td>Destination IP Group</td>
<td>ITSP</td>
</tr>
</tbody>
</table>

• Routes calls from the ITSP (identified by the source tag name-value "Type=ITSP") to the IP PBXs (identified by the specific port assigned to the IP PBX by the value of the destination tag name "Port"):  

<table>
<thead>
<tr>
<th>General</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>2</td>
</tr>
<tr>
<td>Name</td>
<td>ITSP-to-PBX</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Match</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Tag</td>
<td>Type=ITSP</td>
</tr>
</tbody>
</table>
27.3 Configuring SIP Response Codes for Alternative Routing Reasons

The 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 (e.g., SIP 406), the device does alternative routing as follows: If the destination Proxy Set is configured with multiple IP addresses (see Configuring Proxy Sets on page 359), the device first attempts to route the call to one of these IP addresses, starting with the first listed address. If unsuccessful, the device then searches for an alternative routing rule in the IP-to-IP Routing table (see ‘Configuring SBC IP-to-IP Routing Rules’ on page 501).

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 the response code in the Alternative Routing Reasons table.

You can also configure alternative routing for the following proprietary response codes, if configured in the table, that are issued by the device itself:

- **805 IP Profile Call Limit**: The device generates the response code when Call Admission Control (CAC) limits (e.g., 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 489). 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 the response code when the call is terminated due to crossed user-defined thresholds of QoE metrics such as MOS, packet delay, and packet loss (see "Configuring Quality of Experience Profiles" on page 307) and/or media bandwidth (see "Configuring Bandwidth Profiles" on page 312). 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 Alternative Routing Reasons table and 3) configuring an alternative routing rule.

The device also generates the response code when it rejects a call based on Quality of Service rules due to crossed Voice Quality and Bandwidth thresholds (see "Configuring Quality of Service Rules" on page 316). If the response code is configured in the table and the device rejects a call due to threshold crossing, it searches in the IP-to-IP Routing table for an alternative routing rule.

- **818 Signalling Limits Exceeded**: The device generates the response code when it rejects a call based on Quality of Service rules due to crossed ASR, NER or ACD thresholds (see "Configuring Quality of Service Rules" on page 316). If the response code is configured in the table and the device rejects a call due to threshold crossing, it searches in the IP-to-IP Routing table for an alternative routing rule.
Note:
- If the device receives a SIP 408 response, an ICMP message, or no response, alternative routing is still performed even if the code is not configured in the Alternative Routing Reasons table.
- SIP requests belonging to an SRD or IP Group that have reached the call limit (maximum concurrent calls and/or call rate) as configured in the Call Admission table are sent to an alternative route if configured in the IP-to-IP Routing table for the SRD or IP Group. If no alternative routing rule is located, the device automatically rejects the SIP request with a SIP 480 (Temporarily Unavailable) response.

The following procedure describes how to configure the Alternative Routing Reasons table through the Web interface. You can also configure it through ini file (SBCAlternativeRoutingReasons) or CLI (configure voip > sbc routing sbc-alt-routing-reasons).

➢ To configure SIP reason codes for alternative IP routing:

1. Open the Alternative Routing Reasons table (Setup menu > Signaling & Media tab > SBC folder > Routing > Alternative Reasons).
2. Click New; the following dialog box appears:

   Figure 27-7: Alternative Routing Reasons Table - Dialog Box

3. Configure a SIP response code for alternative routing according to the parameters described in the table below.
4. Click Apply.

Table 27-3: Alternative Routing Reasons Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[SBCAlternativeRoutingReasons_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Release Cause</td>
<td>Defines a SIP response code for triggering the device's alternative routing mechanism.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>

### 27.4 Configuring SBC Routing Policy Rules

The Routing Policies table lets you configure up to 600 Routing Policy rules. A Routing Policy determines the routing and manipulation (inbound and outbound) rules per SRD in a multiple SRD configuration topology. The Routing Policy also configures the following:

- Enables Least Cost Routing (LCR), and configures default call cost (highest or lowest) and average call duration for routing rules that are not assigned LCR Cost Groups. The default call cost determines whether matched routing rules that are not assigned Cost Groups are considered as a higher or lower cost route compared to other matching routing rules that are assigned Cost Groups. If you disable LCR, the device ignores the Cost Groups assigned to the routing rules in the IP-to-IP Routing table.

- Assigns LDAP servers (LDAP Server Group) for LDAP-based routing. IP-to-IP routing rules configured for LDAP or CSR (Call Setup Rules) queries use the LDAP server(s) that is assigned to the routing rule’s associated Routing Policy. You can configure a Routing Policy per SRD or alternatively, configure a single Routing Policy that is shared between all SRDs.

The implementation of Routing Policies is intended for the following deployments only:

- Deployments requiring LCR and/or LDAP-based routing.

- Multi-tenant deployments that require multiple, logical routing tables where each tenant has its own dedicated ("separated") routing (and manipulation) table. In such scenarios, each SRD (tenant) is configured as an Isolated SRD and assigned its own unique Routing Policy, implementing an almost isolated, non-bleeding routing configuration topology.

For all other deployment scenarios, the Routing Policy is irrelevant and the handling of the configuration entity is not required as a default Routing Policy ("Default_SBCRoutingPolicy" at Index 0) is provided. When only one Routing Policy is required, the device automatically associates the default Routing Policy with newly added configuration entities that can be associated with the Routing Policy (as mentioned later in this section, except for Classification rules). This facilitates configuration, eliminating the need to handle the Routing Policy configuration entity (except if you need to enable LCR and/or assign an LDAP server to the Routing Policy). In such a setup, where only one Routing Policy is used, single routing and manipulation tables are employed for all SRDs.

**Note:** If possible, it is recommended to use only one Routing Policy for all SRDs (tenants), unless deployment requires otherwise (i.e., a dedicated Routing Policy per SRD).
Once configured, you need to associate the Routing Policy with an SRD(s) in the SRDs table. To determine the routing and manipulation rules for the SRD, you need to assign the Routing Policy to routing and manipulation rules. The figure below shows the configuration entities to which Routing Policies can be assigned:

![Configuration Entities Diagram]

Typically, assigning a Routing Policy to a Classification rule is not required, as when an incoming call is classified it uses the Routing Policy associated with the SRD to which it belongs. However, if a Routing Policy is assigned to a Classification rule, it overrides the Routing Policy assigned to the SRD. The option to assign Routing Policies to Classification rules is useful in deployments requiring different routing and manipulation rules for specific calls pertaining to the same SRD. In such scenarios, you need to configure multiple Classification rules for the same SRD, where for some rules no Routing Policy is assigned (i.e., the SRD's assigned Routing Policy is used) while for others a different Routing Policy is specified to override the SRD's assigned Routing Policy.

In multi-tenant environments employing multiple SRDs and Routing Policies, the IP Groups that can be used in routing rules (in the IP-to-IP Routing table) are as follows:

- If the Routing Policy is assigned to only one SRD and the SRD is an Isolated SRD, the routing rules of the Routing Policy can be configured with IP Groups belonging to the Isolated SRD and IP Groups belonging to all Shared SRDs.

- If the Routing Policy is assigned to a Shared SRD, the routing rules of the Routing Policy can be configured with any IP Group (i.e., belonging to Shared and Isolated SRDs). In effect, the Routing Policy can include routing rules for call routing between Isolated SRDs.

- If the Routing Policy is assigned to multiple SRDs (Shared and/or Isolated), the routing rules of the Routing Policy can be configured with IP Groups belonging to all Shared SRDs as well as IP Groups belonging to Isolated SRDs that are assigned the Routing Policy.

To facilitate the configuration of routing rules in the IP-to-IP Routing table through the Web interface, only the permitted IP Groups (according to the above) are displayed as optional values.

The general flow for processing the call for multi-tenant deployments and Routing Policies is as follows:

1. Using the Classification table, the device classifies the incoming call to an IP Group, based on the SIP Interface on which the call is received. Based on the SIP Interface, the device associates the call to the SRD that is assigned to the SIP Interface.

2. Once the call has been successfully classified to an IP Group, the Routing Policy assigned to the associated SRD is used. However, if a Routing Policy is configured in the Classification table, it overrides the Routing Policy assigned to the SRD.

3. The regular manipulation (inbound and outbound) and routing processes are done according to the associated Routing Policy.
The following procedure describes how to configure Routing Policies rules through the Web interface. You can also configure it through ini file (SBCRoutingPolicy) or CLI (configure voip > sbc routing sbc-routing-policy).

➢ **To configure a Routing Policy rule:**

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

   ![Figure 27-8: Routing Policies Table - Add Dialog Box]

3. Configure the Routing Policy rule according to the parameters described in the table below.
4. Click **Apply**.

   ![Table 27-4: Routing Policies table Parameter Descriptions]

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td>Note: Each row must be configured with a unique index.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>[SBCRoutingPolicy_Name]</td>
<td>The valid value is a string of up to 40 characters. By default, no name is defined. If you don’t configure a name, the device automatically assigns a name in the following format: &quot;SBCRoutingPolicy_&lt;Index&gt;&quot;, for example, &quot;SBCRoutingPolicy_2&quot;. <strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>LDAP Servers Group Name</td>
<td>Assigns an LDAP Server Group to the Routing Policy. Routing rules in the IP-to-IP Routing table that are associated with the Routing Policy and that are configured with LDAP and/or Call Setup Rules, use the LDAP server(s) configured for this LDAP Server Group. By default, no value is defined. For more information on LDAP Server Groups, see &quot;Configuring LDAP Server Groups&quot; on page 238. <strong>Note:</strong> The default Routing Policy is assigned the default LDAP Server Group (&quot;DefaultCTRLServersGroup&quot;).</td>
</tr>
</tbody>
</table>

**Least Cost Routing**

<table>
<thead>
<tr>
<th>LCR Feature</th>
<th>Enables the Least Cost Routing (LCR) feature for the Routing Policy.</th>
</tr>
</thead>
</table>
| lcr-enable | • [0] Disable (default)  
• [1] Enable |
| [SBCRoutingPolicy_LCREnable] | For more information on LCR, see "Least Cost Routing" on page 263. |

<table>
<thead>
<tr>
<th>Default Call Cost</th>
<th>Defines whether routing rules in the IP-to-IP Routing table that are not assigned a Cost Group are considered a higher cost or lower cost route compared to other matched routing rules that are assigned Cost Groups.</th>
</tr>
</thead>
</table>
| lcr-default-cost | • [0] Lowest Cost = (Default) The device considers a matched routing rule (belonging to the Routing Policy) that is not assigned a Cost Group as the lowest cost route. Therefore, it uses the routing rule.  
• [1] Highest Cost = The device considers a matched routing rule (belonging to the Routing Policy) that is not assigned a Cost Group as the highest cost route. Therefore, it is only used if the other matched routing rules that are assigned Cost Groups are unavailable. **Note:** If multiple matched routing rules without an assigned Cost Group exist, the device selects the first matched rule in the table. |
| [SBCRoutingPolicy_LCRDefaultCost] | |

| LCR Call Duration | Defines the average call duration (in minutes) and is used to calculate the variable portion of the call cost. This is useful, for example, when the average call duration spans over multiple time bands. The LCR is calculated as follows: cost = call connect cost + (minute cost * average call duration). The valid value is 0-65533. The default is 1. For example, assume the following Cost Groups:  
• "Weekend A": call connection cost is 1 and charge per minute is 6. Therefore, a call of 1 minute cost 7 units.  
• "Weekend B": call connection cost is 6 and charge per minute is 1. Therefore, a call of 1 minute cost 7 units. |
| lcr-call-length | [SBCRoutingPolicy_LCRAverageCal iLength] |
### 27.5 Configuring IP Group Sets

The IP Group Set table lets you configure up to 750 IP Group Sets. An IP Group Set is a group of IP Groups used for load balancing of calls, belonging to the same source, to a call destination (i.e., IP Group). Each IP Group Set can include up to five IP Groups (Server-type and/or Gateway-type only). The chosen destination IP Group for each call depends on the configured load-balancing policy, which can be Round Robin, Random Weights, or Homing (for more information, see the table's description, later in this section).

Alternative routing within the IP Group Set is also supported, whereby if a chosen destination IP Group responds with a reject response that is configured as a reason for alternative routing (see Configuring SIP Response Codes for Alternative Routing Reasons on page 519) or doesn't respond at all (i.e., keep-alive with its' associated Proxy Set fails), the device attempts to send the call to another IP Group in the IP Group Set (according to the load-balancing policy). For enabling Proxy Set keep-alive, see Configuring Proxy Sets on page 359.

An example of round-robin load-balancing and alternative routing: The first call is sent to IP Group #1 in the IP Group Set, the second call to IP Group #2, and the third call to IP Group #3. If the call sent to IP Group #1 is rejected, the device employs alternative routing and sends it to IP Group #4.

Once you have configured your IP Group Set, to implement call load-balancing by IP Groups, do one of the following:

- In the IP-to-IP Routing table, configure the routing rule’s 'Destination Type' parameter to **IP Group Set**, and then assign it the IP Group Set in the 'IP Group Set' parameter.

- If you are routing to IP Groups based on Dial Plan tags:
  
  - In the IP Group Set table (see below), specify the tag name.
  
  - In the IP-to-IP Routing table, configure the routing rule's 'Destination Type' parameter to **Destination Tag**, and then specify the tag name in the 'Routing Tag Name' parameter.

For more information on IP-to-IP Routing rules, see Configuring SBC IP-to-IP Routing Rules on page 501. For more information on routing based on destination Dial Plan tags, see Using Dial Plan Tags for Routing Destinations on page 551.

IP Group Sets are configured using two tables with parent-child type relationship:

- **Parent table**: IP Group Set table, which defines the name and load-balancing policy of the IP Group Set.

- **Child table**: IP Group Set Member table, which assigns IP Groups to IP Group Sets. You can assign up to five IP Groups per IP Group Set.

The following procedure describes how to configure IP Group Sets through the Web interface. You can also configure it through other management platforms:

- **IP Group Set Table**: ini file (IPGroupSet) or CLI (configure voip > sbc routing ip-group-set)

- **IP Group Set Member Table**: ini file (IPGroupSetMember) or CLI (configure voip > sbc routing ip-group-set-member)
To configure an IP Group Set:

1. Open the IP Group Set table (Setup menu > Signaling & Media tab > SBC folder > Routing > IP Group Set).

2. Click New; the following dialog box appears:

   ![Figure 27-9: IP Group Set Table - Dialog Box]

3. Configure the IP Group Set according to the parameters described in the table below.

4. Click Apply.

**Table 27-5: IP Group Set Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>Index [IPGroupSet_Index]</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name [IPGroupSet_Name]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables.</td>
</tr>
<tr>
<td></td>
<td>The valid value is a string of up to 40 characters. By default, no name is defined. If you don't configure a name, the device automatically assigns a name in the following format: &quot;IPGroupSet_&lt;index&gt;&quot;. For example, if you add a new row to Index 0, the following name is assigned: &quot;IPGroupSet_0&quot;</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
**Policy**
- **policy**
  - [IPGroupSet_Policy]
  - Defines the load-balancing policy.
    - [0] Round-Robin = (Default) The device selects the next consecutive, available IP Group for each call. The device selects the first IP Group in the table (i.e., lowest index) for the first call and the next consecutive IP Groups for the next calls. For example, first call to IP Group at Index 0, second call to IP Group at Index 2, third call to IP Group at Index 3, and so on. If an IP Group is offline, the device selects the next consecutive IP Group. Once the last IP Group in the IP Group Set list is selected for a call, the device goes to the beginning of the list and sends the next call to the first IP Group, and so on.
    - [1] Random Weight = The device selects IP Groups at random and their weights determine their probability of getting chosen over others. The higher the weight, the more chance of the IP Group being chosen.
    - [2] Homing = The device always attempts to send all calls to the first IP Group in the table (i.e., lowest index). If unavailable, it sends the calls to the next consecutive, available IP Group. However, if the first IP Group comes online again, the device selects it.

**Note:** For the **Random Weight** optional value, use the 'Weight' parameter in the IP Group Set Member table (below) to configure weight value per IP Group.

**Tags**
- **tags**
  - [IPGroupSet_Tags]
  - Assigns a Dial Plan tag that is used to determine whether the incoming SIP dialog is sent to IP Groups belonging to this IP Group Set. The parameter is used when IP-to-IP Routing rules are configured for destination based on tags (i.e., 'Destination Type' parameter configured to **Destination Tag**). For more information on routing based on destination tags, see Using Dial Plan Tags for Routing Destinations on page 551.

**Note:** If the IP Groups belonging to the IP Group Set are also configured with Dial Plan tags, the Dial Plan tag configured for the parameter takes precedence. If the same Dial Plan tag is also configured for other IP Groups in the IP Groups table, the IP Group Set takes precedence and the device sends the SIP dialog to the IP Group(s) belonging to the IP Group Set.

5. Select the IP Group Set row for which you want to assign IP Groups, and then click the **IP Group Set Member** link located below the table; the IP Group Set Member table appears.
6. Click **New**; the following dialog box appears:

   **Figure 27-10: IP Group Set Member Table - Dialog Box**

   ![IP Group Set Member Table - Dialog Box]

7. Configure IP Group Set members according to the parameters described in the table below.

8. Click **Apply**, and then save your settings to flash memory.

   **IP Group Set Member Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>[IPGroupSetMember_IPGroupSetMemberIndex]</td>
<td>Assigns an IP Group to the IP Group Set.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The IP Group can only be a Server-type or Gateway-type.</td>
</tr>
<tr>
<td>[IPGroupSetMember_IPGroupName]</td>
<td>Assigns an IP Group to the IP Group Set.</td>
</tr>
<tr>
<td>[IPGroupSetMember.Weight]</td>
<td>Defines the weight of the IP Group. The higher the weight, the more chance of the IP Group being selected as the destination of the call.</td>
</tr>
<tr>
<td>[IPGroupSetMember.Weight]</td>
<td>The valid value is 1 to 9. The default is 1.</td>
</tr>
<tr>
<td>[IPGroupSetMember.Weight]</td>
<td><strong>Note:</strong> The parameter is applicable only if you configure the 'Policy' parameter to <strong>Random Weight</strong>.</td>
</tr>
</tbody>
</table>
28 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 393.

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).

![Image: SIP URI Manipulation in IP-to-IP Routing](image)

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 Groups 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=z9hGLLLLan
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=01LAN;parameter1 =abe
To: <sip:1000@10.2.2.3;user=phone>
Call-ID: 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;parameter1=abe
To: <sip: 9721000@ ITSP;user=phone>
Call-ID: USEVWWAN@212.179.1.12
CSeq: 38 INVITE
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 (contribution 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;parameter1=abe
to
From:<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;parameter1=abe
```

- **Source IP Group name (i.e., SIP URI host name) from "10.2.2.6" to "IP_PBX":**

```
From:<sip:7000@10.2.2.6;user=phone;x=y;z=a>;tag=OlLAN;parameter1=abe
to
From:<sip:97000@IP_PBX;user=phone;x=y;z=a>;tag=OWan;parameter1=abe
```

- **Inbound destination SIP URI user name from "1000" to 9721000":**
28. SBC Manipulations

28.1 Configuring IP-to-IP Inbound Manipulations

The Inbound Manipulations table lets you configure up to 3,000 IP-to-IP Inbound Manipulation rules. An 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).

Configuration of Inbound Manipulation rules includes two areas:

- **Match**: Defines the matching characteristics of an incoming SIP dialog (e.g., source host name).
- **Action**: Defines the action that is done if the incoming call matches the characteristics of the rule. In other words, the device manipulates the source or destination SIP URI user part of the SIP dialog (e.g., removes a user-defined number of characters from the left of the SIP URI user part).

**Note**: Configure stricter classification rules higher up in the table than less strict rules to ensure the desired rule is used to manipulate the incoming dialog. **Strict** refers to the number of matching characteristics configured for the rule. For example, a rule configured with source host name and source IP Group as matching characteristics is stricter than a rule configured with only source host name. If the rule configured with only source host name appears higher up in the table, the device ("erroneously") uses the rule to manipulate incoming dialogs matching this source host name (even if they also match the rule appearing lower down in the table configured with the source IP Group as well).

To configure and apply an Inbound Manipulation rule, the rule must be associated with a Routing Policy. The Routing Policy associates the rule with an SRD(s). Therefore, the Routing Policy lets you configure manipulation rules for calls belonging to specific SRD(s). However, as multiple Routing Policies are relevant only for multi-tenant deployments (if needed), for most deployments, only a single Routing Policy is required. As the device provides a default Routing Policy ("Default_SBCRoutingPolicy"), when only one Routing Policy is required, the device automatically assigns the default Routing Policy to the routing rule. If you are implementing LDAP-based routing (with or without Call Setup Rules) and/or...
Least Cost Routing (LCR), you need to configure these settings for the Routing Policy (regardless of the number of Routing Policies employed). For more information on Routing Policies, see "Configuring SBC Routing Policy Rules" on page 521.

**Note:** The IP Groups 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 345).

The following procedure describes how to configure Inbound Manipulation rules through the Web interface. You can also configure it through ini file (IPInboundManipulation) or CLI (configure voip > sbc manipulation ip-inbound-manipulation).

- **To configure an Inbound Manipulation rule:**
  1. Open the Inbound Manipulations table (Setup menu > Signaling & Media tab > SBC folder > Manipulation > Inbound Manipulations).
  2. Click **New**; the following dialog box appears:

  ![Inbound Manipulations Table - Add Dialog Box](image)

  3. Configure the Inbound Manipulation rule according to the parameters described in the table below.
  4. Click **Apply**.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Policy routing-policy-name [IPInboundManipulation_RoutingPolicyName]</td>
<td>Assigns an Routing Policy to the rule. The Routing Policy associates the rule with an SRD(s). The Routing Policy also defines default LCR settings as well as the LDAP servers if the routing rule is based on LDAP routing (and Call Setup Rules).</td>
</tr>
</tbody>
</table>
If only one Routing Policy is configured in the Routing Policies table, the Routing Policy is automatically assigned. If multiple Routing Policies are configured, no value is assigned.

To configure Routing Policies, see "Configuring SBC Routing Policy Rules" on page 521.

**Note:** The parameter is mandatory.

---

**Parameter** | **Description**
---|---
Index | Defines an index number for the new table record. **Note:** Each table row must be configured with a unique index.

**Name**

*manipulation-name*

* [IPInboundManipulation_ManipulationName]  

Defines an arbitrary name to easily identify the manipulation rule. The valid value is a string of up to 40 characters. By default, no value is defined.

**Additional Manipulation**

* CLI: is-additional-manipulation  

* [IPInboundManipulation_IsAdditionalManipulation]  

Determines whether additional SIP URI user part manipulation is done for the table entry rule listed directly above it.

- **[0]** No = (Default) Regular manipulation rule (not done in addition to the rule above it).
- **[1]** Yes = If the above row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule.

**Note:** Additional manipulation can only be done on a different SIP URI, source or destination, to the rule configured in the row above as configured by the 'Manipulated URI' parameter (see below).

**Manipulation Purpose**

* CLI: purpose  

* [IPInboundManipulation_ManipulationPurpose]  

Defines the purpose of the manipulation:

- **[0]** Normal = (Default) Inbound manipulations affect the routing input and source and/or destination number.
- **[1]** Routing input only = Inbound manipulations affect the routing input only, retaining the original source and destination number.
- **[2]** Shared Line = Used for the Shared-Line Appearance feature. This manipulation is for registration requests to change the destination number of the secondary extension numbers to the primary extension. For more information, see "Configuring BroadSoft’s Shared Phone Line Call Appearance for Survivability" on page 575.

**Match**

**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.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source IP Group</th>
<th>CLI: src-ip-group-name [IPInboundManipulation_SrcIpGroupName]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines the IP Group from where the incoming INVITE is received. The default is Any (i.e., any IP Group).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Username Prefix</th>
<th>CLI: src-user-name-prefix [IPInboundManipulation_SrcUsernamePrefix]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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). <strong>Note:</strong> The prefix can be a single digit or a range of digits. For available notations, see &quot;Dialing Plan Notation for Routing and Manipulation&quot; on page 827.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Host</th>
<th>CLI: src-host [IPInboundManipulation_SrcHost]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination Username Prefix</th>
<th>CLI: dst-user-name-prefix [IPInboundManipulation_DestUsernamePrefix]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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). <strong>Note:</strong> The prefix can be a single digit or a range of digits. For available notations, see &quot;Dialing Plan Notation for Routing and Manipulation&quot; on page 827.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Destination Host</th>
<th>CLI: dst-host [IPInboundManipulation_DestHost]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Operation Rule - Action</th>
<th>Manipulated Item</th>
<th>CLI: manipulated-uri [IPInboundManipulation_ManipulatedURI]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determines whether the source or destination SIP URI user part is manipulated. <strong>Note:</strong></td>
<td>[0] Source = (Default) Manipulation is done on the source SIP URI user part.</td>
<td></td>
</tr>
<tr>
<td>[1] Destination = Manipulation is done on the destination SIP URI user part.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remove From Left</th>
<th>CLI: remove-from-left [IPInboundManipulation_RemoveFromLeft]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 &quot;john&quot;, the new user name is &quot;n&quot;.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Remove From Right</th>
<th>CLI: remove-from-right [IPInboundManipulation_RemoveFromRight]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 &quot;john&quot;, the new user name is &quot;j&quot;. <strong>Note:</strong> If both 'Remove From Right' and 'Leave From Right' parameters are configured, the 'Remove From Right' parameter will be ignored.</td>
<td></td>
</tr>
</tbody>
</table>
### 28.2 Configuring IP-to-IP Outbound Manipulations

The Outbound Manipulations table lets you configure up to 3,000 IP-to-IP Outbound Manipulation rules. An 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. Outbound Manipulation rules can be applied to any SIP request type (e.g., INVITE). 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 (if exists), P-Preferred (if exists), and Remote-Party-ID (if exists).

Configuration of Outbound Manipulation rules includes two areas:

- **Match:** Defines the matching characteristics of an incoming SIP dialog (e.g., source host name). As the device performs outbound manipulations only after the routing process, destination IP Groups can also be used as matching characteristics.

- **Action:** Defines the action that is done if the incoming call matches the characteristics of the rule. In other words, the device manipulates the source or destination SIP URI user part or calling name of the SIP dialog (e.g., removes a user-defined number of characters from the left of the SIP URI user part).

**Note:**

- Configure stricter classification rules higher up in the table than less strict rules to ensure the desired rule is used to manipulate the outbound dialog. *Strict* refers to the number of matching characteristics configured for the rule. For example, a rule configured with source host name and source IP Group as matching characteristics is stricter than a rule configured with only source host name. If the rule configured with only source host name appears higher up in the table, the device ("erroneously") uses the rule to manipulate outbound dialogs matching this source host name (even if they also match the rule appearing lower down in the table configured with the source IP Group as well).

- SIP URI host name (source and destination) manipulations can also be configured in the IP Groups table (see "Configuring IP Groups" on page 345). 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 Outbound Manipulations rules through the Web interface. You can also configure it through ini file (IPOutboundManipulation) or CLI (configure voip > sbc manipulation ip-outbound-manipulation).

To configure Outbound Manipulation rules:

1. Open the Outbound Manipulations table (Setup menu > Signaling & Media tab > SBC folder > Manipulation > Outbound Manipulations).
2. Click New; the following dialog box appears:

![Outbound Manipulations Table- Add Dialog Box](image)

3. Configure an Outbound Manipulation rule according to the parameters described in the table below.
4. Click Apply.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routing Policy routing-policy-name [IPOutboundManipulation_RoutingPolicyName]</td>
<td>Assigns a Routing Policy to the rule. The Routing Policy associates the rule with an SRD(s). The Routing Policy also defines default LCR settings as well as the LDAP servers if the routing rule is based on LDAP routing (and Call Setup Rules). If only one Routing Policy is configured in the Routing Policies table, the Routing Policy is automatically assigned. If multiple Routing Policies are configured, no value is assigned. To configure Routing Policies, see &quot;Configuring SBC Routing Policy Rules&quot; on page 521. <strong>Note:</strong> The parameter is mandatory.</td>
</tr>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Index [IPOutboundManipulation_Index]</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name manipulation-name [IPOutboundManipulation_ManipulationName]</td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 40 characters. By default,</td>
</tr>
</tbody>
</table>
### Parameter | Description
---|---
**onName** | no value is defined.

**Additional Manipulation**

**is-additional-manipulation**

**[IPOutboundManipulation_IsAdditionalManipulation]**

Determines whether additional manipulation is done for the table entry rule listed directly above it.

- **[0]** No = (Default) Regular manipulation rule - not done in addition to the rule above it.
- **[1]** Yes = If the previous table row entry rule matched the call, consider this row entry as a match as well and perform the manipulation specified by this rule.

**Note:** Additional manipulation can only be done on a different item (source URI, destination URI, or calling name) to the rule configured in the row above (configured by the 'Manipulated URI' parameter).

**Call Trigger**

**trigger**

**[IPOutboundManipulation_Trigger]**

Defines the reason (i.e., trigger) for the re-routing of the SIP request:

- **[0]** Any = (Default) Re-routed for all scenarios (re-routes and non-re-routes).
- **[1]** 3xx = Re-routed if it triggered as a result of a SIP 3xx response.
- **[2]** REFER = Re-routed if it triggered as a result of a REFER request.
- **[3]** 3xx or REFER = Applies to options [1] and [2].
- **[4]** Initial only = Regular requests that the device forwards to a destination. In other words, re-routing of requests triggered by the receipt of REFER or 3xx does not apply.

**Match**

**Request Type**

**request-type**

**[IPOutboundManipulation_RequestType]**

Defines the SIP request type to which the manipulation rule is applied.

- **[0]** All = (Default) all SIP messages.
- **[1]** INVITE = All SIP messages except REGISTER and SUBSCRIBE.
- **[2]** REGISTER = Only SIP REGISTER messages.
- **[3]** SUBSCRIBE = Only SIP SUBSCRIBE messages.
- **[4]** INVITE and REGISTER = All SIP messages except SUBSCRIBE.
- **[5]** INVITE and SUBSCRIBE = All SIP messages except REGISTER.

**Source IP Group**

**src-ip-group-name**

**[IPOutboundManipulation_SrcIPGroupName]**

Defines the IP Group from where the INVITE is received. The default value is **Any** (i.e., any IP Group).

**Destination IP Group**

**dst-ip-group-name**

**[IPOutboundManipulation_DestIPGroupName]**

Defines the IP Group to where the INVITE is to be sent. The default value is **Any** (i.e., any IP Group).

**Source Username Prefix**

**src-user-name-prefix**

**[IPOutboundManipulation_SrcUserName]**

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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>amePrefix</td>
<td>username prefix). The prefix can be a single digit or a range of digits. For available notations, see &quot;Dialing Plan Notation for Routing and Manipulation&quot; on page 827. <strong>Note:</strong> If you need to manipulate calls of many different source URI user names, you can use tags (see 'Source Tags' parameter below) instead of this parameter.</td>
</tr>
<tr>
<td>Source Host &lt;br&gt; src-host</td>
<td>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).</td>
</tr>
</tbody>
</table>
| Source Tags <br> src-tags | Assigns a prefix tag to denote source URI user names corresponding to the tag configured in the associated Dial Plan. The valid value is a string of up to 20 characters. The tag is case insensitive. To configure prefix tags, see "Configuring Dial Plans" on page 541. **Note:**  
  - Make sure that you assign the Dial Plan in which you have configured the prefix tag, to the related IP Group or SRD.  
  - Instead of using tags and configuring the parameter, you can use the 'Source Username Prefix' parameter to specify a specific URI source user or all source users. |
| Destination Username Prefix <br> dst-user-name-prefix | 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 827. **Note:** If you need to manipulate calls of many different destination URI user names, you can use tags (see 'Destination Tags' parameter below) instead of this parameter. |
| Destination Host <br> dst-host | Defines the destination SIP URI host name - full name, typically located in the Request-URI and To headers. The default value is the asterisk (*) symbol (i.e., any destination host name). |
| Destination Tags <br> dest-tags | Assigns a prefix tag to denote destination URI user names corresponding to the tag configured in the associated Dial Plan. The valid value is a string of up to 20 characters. The tag is case insensitive. To configure prefix tags, see "Configuring Dial Plans" on page 541. **Note:**  
  - Make sure that you assign the Dial Plan in which you have configured the prefix tag, to the related IP Group or SRD.  
  - Instead of using tags and configuring the parameter, you can use the 'Destination Username Prefix' parameter to specify a specific URI destination user or all destinations users. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calling Name Prefix</td>
<td>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.</td>
</tr>
<tr>
<td>calling-name-prefix</td>
<td></td>
</tr>
<tr>
<td>[IPOutboundManipulation_CallingNamePrefix]</td>
<td></td>
</tr>
<tr>
<td>message-condition-name</td>
<td></td>
</tr>
<tr>
<td>[IPOutboundManipulation_MessageConditionName]</td>
<td></td>
</tr>
<tr>
<td>ReRoute IP Group</td>
<td>Defines the IP Group that initiated (sent) the SIP redirect response (e.g., 3xx) or REFER message. The parameter is typically used for re-routing requests (e.g., INVITEs) when interworking is required for SIP 3xx redirect responses or REFER messages.</td>
</tr>
<tr>
<td>re-route-ip-group-name</td>
<td>The default is Any (i.e., any IP Group).</td>
</tr>
<tr>
<td>[IPOutboundManipulation_ReRouteIPGroupName]</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td></td>
</tr>
<tr>
<td>Manipulated Item</td>
<td>Defines the element in the SIP message that you want manipulated.</td>
</tr>
<tr>
<td>manipulated-uri</td>
<td></td>
</tr>
<tr>
<td>[IPOutboundManipulation_IsAdditionalManipulation]</td>
<td></td>
</tr>
<tr>
<td>Remove From Left</td>
<td>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 &quot;john&quot;, the new user name is &quot;n&quot;.</td>
</tr>
<tr>
<td>remove-from-left</td>
<td></td>
</tr>
<tr>
<td>[IPOutboundManipulation_RemoveFromLeft]</td>
<td></td>
</tr>
<tr>
<td>Remove From Right</td>
<td>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 &quot;john&quot;, the new user name is &quot;j&quot;.</td>
</tr>
<tr>
<td>remove-from-right</td>
<td></td>
</tr>
<tr>
<td>[IPOutboundManipulation_RemoveFromRight]</td>
<td></td>
</tr>
<tr>
<td>Leave From Right</td>
<td>Defines the number of digits to keep from the right of the manipulated item.</td>
</tr>
<tr>
<td>leave-from-right</td>
<td></td>
</tr>
<tr>
<td>[IPOutboundManipulation_LeaveFromRight]</td>
<td></td>
</tr>
<tr>
<td>Prefix to Add</td>
<td>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 &quot;john&quot;, the new user name is &quot;userjohn&quot;.</td>
</tr>
<tr>
<td>prefix-to-add</td>
<td></td>
</tr>
<tr>
<td>[IPOutboundManipulation_PrefixToAdd]</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>d]</td>
<td>If you set the ‘Manipulated Item’ parameter to <strong>Source URI</strong> or <strong>Destination URI</strong>, you can configure the parameter to a string of up 49 characters. If you set the ‘Manipulated Item’ parameter to <strong>Calling Name</strong>, you can configure the parameter to a string of up 36 characters.</td>
</tr>
<tr>
<td>Suffix to Add suffix-to-add</td>
<td>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 &quot;john&quot;, the new user name is &quot;john01&quot;. If you set the ‘Manipulated Item’ parameter to <strong>Source URI</strong> or <strong>Destination URI</strong>, you can configure the parameter to a string of up 49 characters. If you set the ‘Manipulated Item’ parameter to <strong>Calling Name</strong>, you can configure the parameter to a string of up 36 characters.</td>
</tr>
<tr>
<td>Privacy Restriction Mode</td>
<td>Defines user privacy handling (i.e., restricting source user identity in outgoing SIP dialogs).</td>
</tr>
<tr>
<td>privacy-restriction-mode</td>
<td>[IPOutboundManipulation_PrivacyRestrictionMode]</td>
</tr>
<tr>
<td></td>
<td>• [0] Transparent = (Default) No intervention in SIP privacy.</td>
</tr>
<tr>
<td></td>
<td>• [1] Don't change privacy = The user identity in the outgoing SIP dialog remains the same as in the incoming SIP dialog. If a restricted number exists, the restricted presentation is normalized as follows:</td>
</tr>
<tr>
<td></td>
<td>✓ From URL header: &quot;<a href="mailto:anonymous@anonymous.invalid">anonymous@anonymous.invalid</a>&quot;</td>
</tr>
<tr>
<td></td>
<td>✓ 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 &quot;id&quot;.</td>
</tr>
<tr>
<td></td>
<td>• [2] Restrict = The user identity is restricted. The restriction presentation is as follows:</td>
</tr>
<tr>
<td></td>
<td>✓ From URL header: &quot;<a href="mailto:anonymous@anonymous.invalid">anonymous@anonymous.invalid</a>&quot;</td>
</tr>
<tr>
<td></td>
<td>✓ 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 &quot;id&quot;.</td>
</tr>
<tr>
<td></td>
<td>• [3] Remove Restriction = The device attempts to reveal the user identity by setting user values in the From header and removing the privacy &quot;id&quot; 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).</td>
</tr>
<tr>
<td>Note:</td>
<td>• Restriction is done only after user number manipulation (if any).</td>
</tr>
<tr>
<td></td>
<td>• The device identifies an incoming user as restricted if one of the following exists:</td>
</tr>
<tr>
<td></td>
<td>✓ From header user is &quot;anonymous&quot;.</td>
</tr>
<tr>
<td></td>
<td>✓ P-Asserted-Identity and Privacy headers contain the value &quot;id&quot;.</td>
</tr>
</tbody>
</table>
29 Configuring Dial Plans

Dial Plans let you categorize users (source and/or destination) according to source and/or destination numbers of the incoming SIP dialog-initiating requests. The device categorizes users by searching in the Dial Plan for rules that match these numbers according to prefix, suffix, and/or whole number. The categorization result in the Dial Plan is a tag corresponding to the matched rules. You can then use the tags to represent these users (source and/or destination users) as matching characteristics (source and/or destination tags) for the following:

- IP-to-IP Routing rules (see "Using Dial Plan Tags for IP-to-IP Routing" on page 550)
- Outbound Manipulations rules ("Using Dial Plan Tags for Outbound Manipulation" on page 556)
- Call Setup Rules (Using Dial Plan Tags for Call Setup Rules on page 557)
- Message Manipulation (Using Dial Plan Tags for Message Manipulation on page 558)

The figure below shows a conceptual example of routing based on tags, where users categorized as tag "A" are routed to SIP Trunk "X" and those categorized as tag "B" are routed to SIP Trunk "Y":

![Routing based on Prefix Tags](image)

**Note:**
- User categorization by Dial Plan is done only after the device's Classification and Inbound Manipulation processes, and before the routing process.
- Once the device successfully categorizes an incoming call by Dial Plan, it not only uses the resultant tag in the immediate routing or manipulation process, but also in subsequent routing and manipulation processes that may occur, for example, due to alternative routing or local handling of call transfer and call forwarding (SIP 3xx/REFER).
- For manipulation, tags are applicable only to outbound manipulation.

You can assign a Dial Plan to an IP Group or SRD. After Classification and Inbound Manipulation, the device checks if a Dial Plan is associated with the incoming call. It first checks the source IP Group and if no Dial Plan is assigned, it checks the SRD. If a Dial Plan is assigned to the IP Group or SRD, the device first searches the Dial Plan for a dial plan rule that matches the source number and then it searches the Dial Plan for a rule that matches the destination number. If matching dial plan rules are found, the tags configured for these rules are used in the routing and/or manipulation processes as source and/or destination tags.
Note: When tags are used in the IP-to-IP Routing table to determine destination IP Groups (i.e., 'Destination Type' parameter configured to Destination Tag), the device searches the Dial Plan for a matching destination (called) prefix number only.

The Dial Plan itself is a set of dial plan rules having the following attributes:

- **Prefix**: The prefix is matched against the source and/or destination number of the incoming SIP dialog-initiating request.
- **Tag**: The tag corresponds to the matched prefix of the source and/or destination number and is the categorization result.

You can use various syntax notations to configure the prefix numbers in dial plan rules. You can configure the prefix as a complete number (all digits) or as a partial number using some digits and various syntax notations (patterns) to allow the device to match a dial plan rule for similar source and/or destination numbers. For more information, see the description of the 'Prefix' parameter (DialPlanRule_Prefix) described later in this section.

The device employs a "best-match" method instead of a "first-match" method to match the source/destination numbers to prefixes configured in the dial plan. The matching order is done digit-by-digit and from left to right. The numbers are first matched to the rule configured with the most constrained (specific) character set. Most constrained implies that the dial plan pattern that has the fewest possible matches for a digit is matched first. For example, if one rule contains the "x" wildcard character, which has ten possible matches (i.e., 0-9) and another rule a specific digit (e.g., 4), the rule with the specific digit is selected as the matching rule. The best match priority is listed below in chronological order:

- Specific character (prefix)
- Number range
- "x" wildcard, which denotes any digit (0-9)
- Suffix, where the longest digits is first matched. For example, ([001-999]) takes precedence over ([01-99]) which takes precedence over ([1-9]).
- . (dot), which denotes any character

For example, the table below shows the best match priority of an incoming call with prefix number "5234":

<table>
<thead>
<tr>
<th>Dial Plan</th>
<th>Tag</th>
<th>Best Match Priority (Where 1 is Highest)</th>
</tr>
</thead>
<tbody>
<tr>
<td>523x</td>
<td>A</td>
<td>3</td>
</tr>
<tr>
<td>523([4]) or ([5234])</td>
<td>A</td>
<td>4</td>
</tr>
<tr>
<td>523[2-6]</td>
<td>A</td>
<td>2</td>
</tr>
<tr>
<td>523.</td>
<td>A</td>
<td>5</td>
</tr>
<tr>
<td>5234</td>
<td>B</td>
<td>1</td>
</tr>
</tbody>
</table>
The following examples show how the best-matching method is done. Each example has two dial plan rules which are shown listed in chronological order as they would be configured in the table.

- For incoming calls with prefix number "5234", the rule with tag B is chosen (more specific for digit "4"):

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>523x</td>
<td>A</td>
</tr>
<tr>
<td>5234</td>
<td>B</td>
</tr>
</tbody>
</table>

- For incoming calls with prefix number "5234", the rule with tag B is chosen (more specific for digit "4"):

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>523x</td>
<td>A</td>
</tr>
<tr>
<td>523[1-9]</td>
<td>B</td>
</tr>
</tbody>
</table>

- For incoming calls with prefix number "5321111", the rule with tag B is chosen (more specific for fourth digit):

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>532[1-9]111</td>
<td>A</td>
</tr>
<tr>
<td>5321</td>
<td>B</td>
</tr>
</tbody>
</table>

- For incoming calls with prefix number "53124", the rule with tag B is chosen (more specific for digit "1"):

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>53([2-4])</td>
<td>A</td>
</tr>
<tr>
<td>531(4)</td>
<td>B</td>
</tr>
</tbody>
</table>

- For incoming calls with prefix number "321444", the rule with tag A is chosen and for incoming calls with prefix number "32144", the rule with tag B is chosen:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>321xxx</td>
<td>A</td>
</tr>
<tr>
<td>321</td>
<td>B</td>
</tr>
</tbody>
</table>

- For incoming calls with prefix number "5324", the rule with tag B is chosen (prefix is more specific for digit "4"):

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>532[1-9]</td>
<td>A</td>
</tr>
<tr>
<td>532[2-4]</td>
<td>B</td>
</tr>
</tbody>
</table>
For incoming calls with prefix number "53124", the rule with tag C is chosen (longest suffix - C has three digits, B two digits and A one digit):

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>53([2-4])</td>
<td>A</td>
</tr>
<tr>
<td>53([01-99])</td>
<td>B</td>
</tr>
<tr>
<td>53([001-999])</td>
<td>C</td>
</tr>
</tbody>
</table>

For incoming calls with prefix number "53124", the rule with tag B is chosen (suffix is more specific for digit "4"):

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>53([2-4])</td>
<td>A</td>
</tr>
<tr>
<td>53(4),B</td>
<td>B</td>
</tr>
</tbody>
</table>

Dial Plans are configured using two tables with parent-child type relationship:

- Parent table: Dial Plan table, which defines the name of the Dial Plan. You can configure up to 50 Dial Plans.
- Child table: Dial Plan Rule table, which defines the actual dial plans (rules) per Dial Plan. You can configure up to 2,000 for Mediant VE of less than 16-GB RAM and 20,000 for greater than (inclusive) 16-GB RAM dial plan rules in total (where all can be configured for one Dial Plan or configured between different Dial Plans).

The following procedure describes how to configure Dial Plans through the Web interface. You can also configure it through other management platforms:

- **Dial Plan table:** ini file (DialPlan) or CLI (configure voip > sbc dial-plan)
- **Dial Plan Rule table:** ini file (DialPlanRule) or CLI (configure voip > sbc dial-plan-rule)

**To configure Dial Plans:**

1. Open the Dial Plan table (Setup menu > Signaling & Media tab > SBC folder > Dial Plan).
2. Click New; the following dialog box appears:

   **Figure 29-2: Dial Plan Table - Add Dialog Box**

3. Configure a Dial Plan name according to the parameters described in the table below.
4. Click Apply.

   **Table 29-2: Dial Plan Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 29. Configuring Dial Plans

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 15 characters.</td>
</tr>
<tr>
<td><strong>Index</strong></td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 15 characters.</td>
</tr>
</tbody>
</table>

**Note:** Each row must be configured with a unique index.

**Note:** Each row must be configured with a unique name.

5. In the Dial Plan table, select the row for which you want to configure dial plan rules, and then click the **Dial Plan Rule** link located below the table; the Dial Plan Rule table appears.

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

![Figure 29-3: Dial Plan Rule Table - Add Dialog Box](image)

7. Configure a dial plan rule according to the parameters described in the table below.

8. Click **New**, and then save your settings to flash memory.

### Table 29-3: Dial Plan Rule Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Index</strong></td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td><strong>Name</strong></td>
<td>Defines a descriptive name, which is used when associating the row in other tables. The valid value is a string of up to 15 characters.</td>
</tr>
<tr>
<td><strong>Prefix</strong></td>
<td>Defines the prefix number of the source or destination number. The valid value is up to 50 characters. The following syntax can be used:</td>
</tr>
<tr>
<td><strong>Tag</strong></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Each row must be configured with a unique index.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a-z</td>
<td>Lower-case letter.</td>
</tr>
<tr>
<td>A-Z</td>
<td>Upper-case letter.</td>
</tr>
<tr>
<td>*</td>
<td>(Asterisk symbol) If it is the only character in the rule, it denotes any number. To denote the asterisk &quot;*&quot; symbol itself, precede it with the escape &quot; character (see below).</td>
</tr>
<tr>
<td>\</td>
<td>(Backslash escape character) When it prefixes a wildcard character (*, z, n, and x), the character itself is used and not the meta-meaning. For example, \x&quot; denotes the character &quot;x&quot;, while &quot;x&quot; is the wildcard denoting any digits from 0-9.</td>
</tr>
<tr>
<td>#</td>
<td>(Pound or hash symbol) When used at the end of the prefix it denotes the end of the number. For example, &quot;54324#&quot; represents a 5-digit number that starts with the digits 54324.</td>
</tr>
<tr>
<td>.</td>
<td>(Period) Denotes any letter or digit.</td>
</tr>
</tbody>
</table>
| [n-m], (n-m), or ([n1-m1,n2-m2,a,b,c,n3-m3]) | Represents a mixed notation of single numbers and multiple ranges. To represent the prefix, the notation is enclosed by square brackets [...]; to represent the suffix, the notation is enclosed by square brackets which are enclosed by parenthesis ([...]). For example, to denote numbers 123 through 130, 455, 766, and 780 through 790: 
  ✓ Prefix: [123-130,455,766,780-790]  
  ✓ Suffix: ([123-130,455,766,780-790])  
  Note: The ranges and the single numbers in the syntax must have the same amount of digits. For example, each number range and single number in the example above consists of three digits. |

**Tag**  
**tag**  
[DialPlanRule_Tag]  
Defines a tag.  
The valid value is up to 16 characters (alphanumeric and special characters such as the dollar $ sign). The tag is case insensitive. The tag can be a name (e.g., "India") or a name with a value (e.g., "Country=India").  
You can also configure multiple tags, where each tag is separated by a semicolon (e.g., "Belgium;Country1=England;Country2=India").  
**Note:** 
- Only one tag name without a value can be configured. For example, "Belgium;South Africa" is an invalid configuration.  
- Tag names with values (i.e., name=value) must be unique. For example, "Country=England;Country=India" is invalid. A valid configuration would be "Country1=England;Country2=India".  
- The same tag can be configured in multiple Dial Plan rules..

## 29.1 Importing and Exporting Dial Plans

You can import and export Dial Plans in comma-separated value (CSV) file format. The Web interface lets you import and export Dial Plans from and to a local folder on the PC running the Web client. The CLI lets you import and export Dial Plans from and to a remote server.  
The following procedures describe how to export configured Dial Plans.
To export all configured dial plan rules:

- Web interface (to a local folder):
  1. Open the Dial Plan table.
  2. From the 'Action' drop-down menu, choose Export; the following dialog box appears:

![Figure 29-4: Exporting Dial Plan](image)

3. Select the Save File option, and then click OK; the file is saved to the default folder on your PC for downloading files.

- CLI (to a remote server):
  ```
  (config-voip)# sbc dial-plan-rule export-csv-to all <URL to CSV file>
  ```

To export rules of a specific Dial Plan:

- Web interface (to a local folder):
  1. Open the Dial Plan table.
  2. Select the required Dial Plan, and then click the Dial Plan Rule link; the Dial Plan Rule table opens, displaying the rules of the selected Dial Plan.
  3. From the 'Action' drop-down menu, choose Export; a dialog box appears (as shown above).
  4. Select the Save File option, and then click OK; the file is saved to the default folder on your PC for downloading files.

- CLI (to a remote server):
  ```
  (config-voip)# sbc dial-plan-rule export-csv-to <Dial Plan name or index> <URL of server>
  ```
  For example:
  ```
  (config-voip)# sbc dial-plan-rule export-csv-to 0 http://10.8.8.20/upload/index_0_Dial_Plans.csv
  ```

The following procedures describe how to import a Dial Plan file.

To overwrite all existing Dial Plans with imported Dial Plan file:

- Web interface (from a local folder):
  1. Open the Dial Plan table.
2. From the 'Action' drop-down menu, choose **Import**; the following dialog box appears:

**Figure 29-5: Importing Dial Plan Rules for Specific Dial Plan**

![Import CSV dialog box](image)

3. Use the **Browse** button to select the Dial Plan file on your PC, and then click **OK**.

**CLI (from a remote server):**

```
(config-voip)# sbc dial-plan-rule import-csv-from all <URL of server>
```

**Note:**
- The file import feature only imports rules of Dial Plans that already exist in the Dial Plan table. If a Dial Plan in the file does not exist in the table, the specific Dial Plan is not imported.
- Make sure that the names of the Dial Plans in the imported file are **identical** to the existing Dial Plan names in the Dial Plan table; otherwise, Dial Plans in the file with different names are not imported.
- When importing a file, the rules in the imported file replace all existing rules of the corresponding Dial Plan. For existing Dial Plans in the Dial Plan table that are not listed in the imported file, the device deletes all their rules. For example, if the imported file contains only the Dial Plan "MyDialPlan1" and the device is currently configured with "MyDialPlan1" and "MyDialPlan2", the rules of "MyDialPlan1" in the imported file replace the rules of "MyDialPlan1" on the device, and the rules of "MyDialPlan2" on the device are deleted (the Dial Plan name itself remains).

➢ **To import dial plan rules for a specific Dial Plan:**

- **Web interface (from a local folder):**
  1. Open the Dial Plan table.
  2. Select the required Dial Plan, and then click the **Dial Plan Rule** link; the Dial Plan Rule table opens, displaying all the rules of the selected Dial Plan.
3. From the 'Action' drop-down menu, choose **Import**; the following dialog box appears:

**Figure 29-6: Importing Dial Plan Rules for Specific Dial Plan**

![Import CSV dialog box](image)

4. Use the **Browse** button to select the Dial Plan file on your PC, and then click **OK**.

**Note:** The rules in the imported file replace all existing rules of the specific Dial Plan.

- CLI (from a remote server):
  
  ```
  (config-voip)# sbc dial-plan-rule import-csv-from <Dial Plan name or index> <URL path to CSV file>
  ```
  
  For example:
  
  ```
  (config-voip)# sbc dial-plan-rule import-csv-from 0 http://10.8.8.20/upload/Dial_Plan_1_Rules.csv
  ```
  
  For creating Dial Plans in a CSV file for import, see "Creating Dial Plan Files for Import" on page 549.

### 29.2 Creating Dial Plan Files

You can configure Dial Plans in an external file (*.csv) and then import them into the device, as described in "Importing and Exporting Dial Plans" on page 546. You can create the file using any text-based editor such as Notepad or Microsoft Excel. The file must be saved with the *.csv file name extension.

To configure Dial Plans in a file, use the following syntax:

```
DialPlanName,Name,Prefix,Tag
```

Where:

- **DialPlanName**: Name of the Dial Plan.
- **Name**: Name of the dial plan rule belonging to the Dial Plan.
- **Prefix**: Source or destination number prefix.
- **Tag**: Result of the user categorization and can be used as matching characteristics for routing and outbound manipulation

For example:

```
PLAN1,rule_100,5511361xx,A
PLAN1,rule_101,551136184[4000-9999]#,B
MyDialPlan,My_rule_200,5511361840000#,itsp_1
MyDialPlan,My_rule_201,666666#,itsp_2
```
29.3 Using Dial Plan Tags for IP-to-IP Routing

You can use Dial Plan tags with IP-to-IP Routing rules in the IP-to-IP Routing table, where tags can be used for the following:

- Matching routing rules by source and/or destination prefix numbers (see Using Dial Plan Tags for Matching Routing Rules on page 550)
- Locating destination IP Group (see Using Dial Plan Tags for Routing Destinations on page 551)

29.3.1 Using Dial Plan Tags for Matching Routing Rules

For deployments requiring hundreds of routing rules (which may exceed the maximum number of rules that can be configured in the IP-to-IP Routing table), you can employ tags to represent the many different calling (source URI user name) and called (destination URI user name) prefix numbers in your routing rules. Tags are typically implemented when you have users of many different called and/or calling numbers that need to be routed to the same destination (e.g., IP Group or IP address). In such a scenario, instead of configuring many routing rules to match all the required prefix numbers, you need only to configure a single routing rule using the tag to represent all the possible prefix numbers.

An example scenario where employing tags could be useful is in deployments where the device needs to service calls in a geographical area that consists of hundreds of local area codes, where each area code is serviced by one of two SIP Trunks in the network. In such a deployment, instead of configuring hundreds of routing rules to represent each local area code, you can simply configure two routing rules where each is assigned a unique tag representing a group of local area codes and the destination IP Group associated with the SIP Trunk servicing them.

Note:
- Source and destination tags can be used in the same routing rule.
- The same tag can be used for source and destination tags in the same routing rule.

The following procedure describes how to configure IP-to-IP routing based on tags.

➢ To configure IP-to-IP routing based on tags:

1. In the Dial Plan table, configure a Dial Plan (see "Configuring Dial Plans" on page 541). For example, the Dial Plan file below defines two tags, "LOC" and "INTL" to represent different called number prefixes for local and long distance (International) calls:

```
<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>PREFIX</th>
<th>TAG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Local</td>
<td>4252/3-5</td>
<td>LOC</td>
</tr>
<tr>
<td>1</td>
<td>Local</td>
<td>4232/7</td>
<td>LOC</td>
</tr>
<tr>
<td>2</td>
<td>Local</td>
<td>42529</td>
<td>LOC</td>
</tr>
<tr>
<td>3</td>
<td>Internal</td>
<td>4252/00</td>
<td>INTL</td>
</tr>
<tr>
<td>4</td>
<td>Internal</td>
<td>4251/00</td>
<td>INTL</td>
</tr>
</tbody>
</table>
```

2. For the IP Group or SRD associated with the calls for which you want to use tag-based routing, assign the Dial Plan that you configured in Step 1.

- IP Groups table: 'Dial Plan' parameter (IPGroup_SBCDialPlanName) - see "Configuring IP Groups" on page 345
29. Configuring Dial Plans

- SRDs table: 'Dial Plan' parameter (SRD_SBCDialPlanName) - see "Configuring SRDs" on page 327

3. In the IP-to-IP Routing table (see "Configuring SBC IP-to-IP Routing Rules" on page 501), configure a routing rule with the required destination and whose matching characteristics include the tag(s) that you configured in your Dial Plan in Step 1. The tags are assigned under the **Match** group, using the following parameters:
  - 'Source Tags' parameter (IP2IPRouting_SrcTags): tag denoting the calling user
  - 'Destination Tags' parameter (IP2IPRouting_DestTags): tag denoting the called user

### 29.3.2 Using Dial Plan Tags for Routing Destinations

You can use Dial Plan tags for determining the destination (IP Group) of an IP-to-IP Routing rule. One of the benefits of using Dial Plan tags is that it can reduce the number of IP-to-IP Routing rules that you would normally need to configure. For example, assume that you need to route calls from IP Group "A" to two different IP Groups, "B" and "C", based on called (destination) prefix number (e.g., 102 and 103). When not using Dial Plan tags, you would need to configure two IP-to-IP Routing rules, where one rule sends calls with prefix number 102 to IP Group "B" and another rule sends calls with prefix number 103 to IP Group "C". However, when using Dial Plan tags, you would need to configure only a single IP-to-IP Routing rule whose destination IP Group is based on a Dial Plan tag.

The following briefly describes the process for using Dial Plan tags in IP-to-IP Routing rules:

1. The device searches the Dial Plan Index, associated with the source IP Group of the incoming SIP dialog, for a Dial Plan rule whose 'Prefix' parameter is configured with the same called prefix number as the SIP dialog (e.g., 102). If found, the device inspects the tags in the 'Tag' parameter (e.g., "Country=England;City=London;Essex") configured for that Dial Plan rule.

2. The device searches for a matching rule in the IP-to-IP Routing table and if the 'Destination Type' parameter is configured to **Destination Tag**, it checks the tag name configured in the 'Routing Tag Name' parameter and compares it with the tags found in the Dial Plan rule. If the 'Routing Tag Name' parameter is configured as "default", the device selects the first tag name in the Dial Plan rule that is configured without a value, for example, "Essex" (see Step 1). If the 'Routing Tag Name' parameter is configured with a specific tag name (e.g., "Country"), the device selects the tag name with its value (e.g., "Country=England") in the Dial Plan rule.

3. The device searches the IP Groups table and IP Group Set table for an IP Group whose 'Tags' parameter is configured with the same tag as selected from the Dial Plan rule. If found, the device routes the call to this IP Group.
The following figure displays the device’s SIP dialog processing when Dial Plan tags are used to determine the destination IP Group:

**Figure 29-7: SIP Dialog Handling for Tag-Based Routing**

The following procedure describes how to configure routing to destination IP Groups determined by Dial Plan tags. The procedure is based on the following example scenario: Calls from IP Group "HQ" with destination (called) prefix number 102 are sent to IP Group "ENG" while calls with destination prefix number 103 are sent to IP Group "BEL". The destination IP Groups are determined by the Dial Plan tags, where the tag "Country=England" is used to send calls to IP Group "ENG" and the tag "Country=Belgium" is used to send calls to IP Group "BEL".

- **To configure routing to destination IP Groups based on Dial Plan tags:**
  1. In the Dial Plan table, configure a Dial Plan with Dial Plan rules, where the 'Prefix' parameter is the destination (called) prefix number. In our example, we will configure a Dial Plan called "Dial Plan 1" with two Dial Plan rules:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Index 0</th>
<th>Index 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>UK</td>
<td>Bel-Neth</td>
</tr>
<tr>
<td>Prefix</td>
<td>102</td>
<td>103</td>
</tr>
</tbody>
</table>

The following procedure describes how to configure routing to destination IP Groups determined by Dial Plan tags. The procedure is based on the following example scenario: Calls from IP Group "HQ" with destination (called) prefix number 102 are sent to IP Group "ENG" while calls with destination prefix number 103 are sent to IP Group "BEL". The destination IP Groups are determined by the Dial Plan tags, where the tag "Country=England" is used to send calls to IP Group "ENG" and the tag "Country=Belgium" is used to send calls to IP Group "BEL".
The following displays the configuration in the Web interface of the Dial Plan rule for Index 0:

Figure 29-8: Dial Plan configuration Example

2. In the IP Groups table, configure your IP Groups. Make sure that you assign the source IP Group with the Dial Plan that you configured in Step 1 and that you configure each destination IP Group with one of the required Dial Plan tags. If the tag has a value, include it as well. In our example, we will configure three IP Groups:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Index 0</th>
<th>Index 1</th>
<th>Index 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>HQ</td>
<td>ENG</td>
<td>BEL</td>
</tr>
<tr>
<td>Dial Plan</td>
<td>Dial Plan 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tags</td>
<td></td>
<td>Country=England</td>
<td>Country=Belgium</td>
</tr>
</tbody>
</table>

3. In the IP-to-IP Routing table, configure a routing rule where the 'Destination Type' parameter is configured to Destination Tag and the 'Routing Tag Name' to one of your Dial Plan tags. In our example, the tag "Country" is used:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Index 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Europe</td>
</tr>
<tr>
<td>Source IP Group</td>
<td>HQ</td>
</tr>
<tr>
<td>Destination Type</td>
<td>Destination Tag</td>
</tr>
<tr>
<td>Routing Tag Name</td>
<td>Country</td>
</tr>
</tbody>
</table>

Note:
- Please adhere to the following regarding the 'Tag' parameter in the Dial Plan Rule table:
  - Only one tag name without a value can be configured. In the above example, "Holland" is the tag name without a value. If an additional tag name is configured, for example, "Holland;France", the setting is invalid.
  - Tag names with values (i.e., name=value) must be unique within a Dial Plan rule. In the above example, "Country=England" is a tag name with value. Configuring the parameter with "Country=England;Country=Scotland" is invalid.
A valid configuration would be "Country=England;Country1=Scotland".

- Configure the 'Routing Tag Name' parameter with only the name of the tag (i.e., without the value, if exists). For example, instead of "Country=England", configure it as "Country" only.
- If the same Dial Plan tag is configured for an IP Group in the IP Groups table and an IP Group Set in the IP Group Set table, the IP Group Set takes precedence and the device sends the SIP dialog to the IP Group(s) belonging to the IP Group Set.

### 29.3.3 Dial Plan Backward Compatibility

**Note:** This section is for backward compatibility only. It is recommended to migrate your Dial Plan configuration to the latest Dial Plan feature (see "Using Dial Plan Tags for IP-to-IP Routing" on page 550).

Configure prefix tags in the Dial Plan file using the following syntax:

```
[ PLAN<index> ]
<prefix number>,0,<prefix tag>
```

where:
- **Index** is the Dial Plan index
- **prefix number** is the called or calling number prefix (ranges can be defined in brackets)
- **prefix tag** is the user-defined prefix tag of up to nine characters, representing the prefix number

Each prefix tag type - called or calling - must be configured in a dedicated Dial Plan index number. For example, Dial Plan 1 can be for called prefix tags and Dial Plan 2 for calling prefix tags.

The example Dial Plan file below defines the prefix tags "LOCL" and "INTL" to represent different called number prefixes for local and long distance calls:

```
[ PLAN1 ]
42520[3-5],0,LOCL
425207,0,LOCL
42529,0,LOCL
425200,0,INTL
425100,0,INTL
....
```

**Note:**
- Called and calling prefix tags can be used in the same routing rule.
- When using prefix tags, you need to configure manipulation rules to remove the tags before the device sends the calls to their destinations.

The following procedure describes how to configure IP-to-IP routing using prefix tags.

➢ **To configure IP-to-IP routing using prefix tags:**

1. Configure a Dial Plan file with prefix tags, and then load the file to the device.
2. Add the prefix tags to the numbers of specific incoming calls using Inbound Manipulation rules:
29. Configuring Dial Plans

a. Open the Inbound Manipulations table (see "Configuring IP-to-IP Inbound Manipulations" on page 531), and then click New.

b. Configure matching characteristics for the incoming call (e.g., set 'Source IP Group' to "1").

c. From the 'Manipulated Item' 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. Configure 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 (see "Configuring SBC IP-to-IP Routing Rules" on page 501), and then click New.

b. Configure the prefix tag in the 'Source Username Prefix' or 'Destination Username Prefix' fields (e.g., "LOCL").

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 Outbound Manipulations table (see "Configuring IP-to-IP Outbound Manipulations" on page 535), and then click New.

b. Configure matching characteristics for the incoming call (e.g., set 'Source IP Group' to "1"), including calls with the prefix tag (in the 'Source Username Prefix' or 'Destination Username Prefix' fields, enter the prefix tag to remove).

c. Configure the 'Remove from Left' or 'Remove from Right' fields (depending on whether you added the tag at the beginning or end of the URI user part, respectively), enter the number of characters making up the tag.
29.4 Using Dial Plan Tags for Outbound Manipulation

You can use Dial Plan tags to denote source and/or destination URI user names in Outbound Manipulation rules in the Outbound Manipulations table. The following procedure describes how to configure Outbound Manipulation based on tags.

To configure Outbound Manipulation based on tags:

1. In the Dial Plan table, configure a Dial Plan (see "Configuring Dial Plans" on page 541).

2. In the IP Group or SRD associated with the calls for which you want to use tag-based routing, assign the Dial Plan that you configured in Step 1.
   - IP Groups table: 'Dial Plan' parameter (IPGroup_SBCDialPlanName) - see "Configuring IP Groups" on page 345
   - SRDs table: 'Dial Plan' parameter (SRD_SBCDialPlanName) - see "Configuring SRDs" on page 327

3. In the Outbound Manipulations table (see "Configuring IP-to-IP Outbound Manipulations" on page 535), configure a rule with the required manipulation and whose matching characteristics include the tag(s) that you configured in your Dial Plan in Step 1. The tags are assigned using the following parameters:
   - 'Source Tags' parameter (IPOutboundManipulation_SrcTags): tag denoting the calling users
   - 'Destination Tags' parameter (IPOutboundManipulation_DestTags): tag denoting the called users
29.5 Using Dial Plan Tags for Call Setup Rules

You can use Dial Plan tags in Call Setup rules, configured in the Call Setup Rules table (see Configuring Call Setup Rules on page 384). The Call Setup rule can be assigned to an IP Group and is processed by the device for the classified source IP Group immediately before the routing process (i.e., Classification > Manipulation > Dial Plan table > Call Setup rules > Routing). The result of the Call Setup rule (i.e., source and/or destination tag) can be used as the matching characteristics for locating a suitable IP-to-IP Routing rule in the IP-to-IP Routing table.

You can configure Call Setup rules to query the Dial Plan table for a specified search key (prefix) in a specified Dial Plan to obtain the corresponding tag. The Call Setup rule can then perform many different manipulations (based on Message Manipulation syntax), including modifying the name of the tag. The tags can be used only in the 'Condition', 'Action Subject' and 'Action Value' fields.
29.6 Using Dial Plan Tags for Message Manipulation

You can use Dial Plan tags (srctags and dsttags) in Message Manipulation rules, configured in the Message Manipulations table (see Configuring SIP Message Manipulation on page 393). The tags can be used only in the 'Condition' and 'Action Value' fields. For example, you can configure a rule that adds the SIP header "City" with the value "ny" (i.e., City: ny) to all outgoing SIP INVITE messages associated with the source tag "ny":

![Message Manipulations Table](image)

*Note:* You cannot modify Dial Plan tags using Message Manipulation rules.
30 Configuring Malicious Signatures

The Malicious Signature table lets you configure up to 30 Malicious Signature patterns. Malicious Signatures are signature patterns that identify SIP user agents (UA) who perform malicious attacks on SIP servers by SIP scanning. Malicious Signatures allow you to protect SBC calls handled by the device from such malicious activities, thereby increasing your SIP security. The Malicious Signature patterns identify specific scanning tools used by attackers to search for SIP servers in the network. The feature identifies and protects against SIP (Layer 5) threats by examining new inbound SIP dialog messages. Once the device identifies an attack based on the configured malicious signature pattern, it marks the SIP message as invalid and discards it or alternatively, rejects it with a SIP response (by default, 400), configured in the Message Policies table. Protection applies only to new dialogs (e.g., INVITE and REGISTER messages) and unauthenticated dialogs.

Malicious signatures can also be used with the Intrusion Detection System (IDS) feature (see "Configuring IDS Policies" on page 172). You can configure an IDS Policy that is activated if the device detects a malicious signature (when the 'Reason' parameter is configured to Dialog establishment failure).

Malicious signature patterns are typically based on the value of SIP User-Agent headers, which attackers use as their identification string (e.g., "User-Agent: VaxSIPUserAgent"). However, you can configure signature patterns based on any SIP header. To configure signature patterns, use the same syntax as that used for configuring Conditions in the Message Manipulations table (see "Configuring SIP Message Manipulation" on page 393).

Below are configured signature patterns based on the User-Agent header:

- Malicious signature for the VaxSIPUserAgent malicious UA:
  header.user-agent.content prefix 'VaxSIPUserAgent'

- Malicious signature for the scanning tool "sip-scan":
  Header.User-Agent.content prefix 'sip-scan'

By default, the table provides preconfigured malicious signatures of known, common attackers.

Note:
- Malicious Signatures do not apply to the following:
  - Calls from IP Groups where Classification is by Proxy Set.
  - In-dialog SIP sessions (e.g., refresh REGISTER requests and re-INVITEs).
  - Calls from users that are registered with the device.
- If you delete all the entries in the table, when you next reset the device, the table is populated again with all the default signatures.

You can export / import Malicious Signatures in CSV file format to / from a remote server through HTTP, HTTPS, or TFTP. To do this, use the following CLI commands:

```
(config-voip)# sbc malicious-signature-database <export-csv-to | import-csv-from> <URL>
```

To apply malicious signatures to calls, you need to enable the use of malicious signatures for a Message Policy and then assign the Message Policy to the SIP Interface associated with the calls (i.e., IP Group). To configure Message Policies, see "Configuring SIP Message Policy Rules".

The following procedure describes how to configure Malicious Signatures through the Web interface. You can also configure it through ini file (MaliciousSignatureDB) or CLI (configure voip > sbc malicious-signature-database).
To configure a Malicious Signature:

1. Open the Malicious Signature table (Setup menu > Signaling & Media tab > SBC folder > Malicious Signature).

2. Click New; the following dialog box appears:

   ![Figure 30-1: Malicious Signature Table - Add Dialog Box]

3. Configure a Malicious Signature according to the parameters described in the table below.

4. Click Apply.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index [ConditionTable_Index]</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name [MaliciousSignatureDB_Name]</td>
<td>Defines an name to easily identify the row. The valid value is a string of up to 30 characters. <strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>Pattern [MaliciousSignatureDB_Pattern]</td>
<td>Defines the signature pattern. The valid value is a string of up to 60 characters. <strong>Note:</strong> The parameter is mandatory.</td>
</tr>
</tbody>
</table>
31 Advanced SBC Features

31.1 Configuring Call Preemption for SBC Emergency Calls

The device supports emergency call preemption for SBC calls by prioritizing emergency calls over regular calls. If the device receives an incoming emergency call when there are unavailable resources to process the call, the device preempts one of the regular calls to free up resources for sending the emergency call to its destination (i.e., emergency service provider), instead of rejecting it. The device may preempt more than one active call in order to provide sufficient resources for processing the emergency call. Available resources depend on the number of INVITE messages currently processed by the device.

If the device preempts a call, it disconnects the call as follows:

- If the call is being setup (not yet established), it sends a SIP 488 response to the incoming leg and a SIP CANCEL message to the outgoing leg.
- If the call is already established, it sends a SIP BYE message to each leg. The device includes in the SIP BYE message, the Reason header describing the cause as "preemption".

Once the device terminates the regular call, it immediately sends the INVITE message of the emergency call to its destination without waiting for any response from the remote sides (e.g., 200 OK after BYE). If the device is unable to preempt a call for the emergency call, it rejects the emergency call with a SIP 503 "Emergency Call Failed" (instead of "Service Unavailable") response.

For the device to identify incoming calls as emergency calls, you need to configure a Message Condition rule in the Message Conditions table. Below are examples of Message Condition rules for identifying emergency calls:

Figure 31-1: Examples of Message Conditions for Identifying Emergency Calls

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>CONDITION</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Emergency1 - RP header</td>
<td>header.resource-priority contains &quot;emergency&quot;</td>
</tr>
<tr>
<td>1</td>
<td>Emergency2 - RP header</td>
<td>header.resource-priority contains &quot;esnet&quot;</td>
</tr>
<tr>
<td>2</td>
<td>Emergency1 - user with providers address</td>
<td>param.call.dist.user=&quot;911&quot;</td>
</tr>
<tr>
<td>3</td>
<td>Emergency2 - user with providers address</td>
<td>param.call.dist.user=&quot;100&quot;</td>
</tr>
<tr>
<td>4</td>
<td>Emergency3 - user with providers address</td>
<td>header.request.uri contains &quot;urn:service:sos&quot;</td>
</tr>
</tbody>
</table>

- Indices 0 and 1: SIP Resource-Priority header contains a string indicating an emergency call.
- Indices 2 to 4: Destination user-part contains the emergency provider's address.

The device applies the Message Condition rule only after call classification (but, before inbound manipulation).

Note:

- The device does not preempt established emergency calls.
- The device does not monitor emergency calls with regards to Quality of Experience (QoE).

➢ To configure SBC emergency call preemption:

1. In the Message Conditions table (see "Configuring Message Condition Rules" on page 501), configure a Message Condition rule to identify incoming emergency calls. See
above for examples.

2. Open the SBC General Settings page (Setup menu > Signaling & Media tab > SBC folder > SBC General Settings), and then scroll down to the Call Priority and Preemption group:

   **Figure 31-2: Configuring Emergency SBC Call Preemption**

   ![Figure 31-2](image)

<table>
<thead>
<tr>
<th>Preemption Mode</th>
<th>Enable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emergency Message Condition</td>
<td>1</td>
</tr>
<tr>
<td>Emergency RTP DiffServ</td>
<td>46</td>
</tr>
<tr>
<td>Emergency Signaling DiffServ</td>
<td>40</td>
</tr>
</tbody>
</table>

3. From the 'Preemption Mode' drop-down list (SBCPreemptionMode), select **Enable** to enable call preemption.

4. In the 'Emergency Message Condition' field, enter the row index of the Message Condition rule that you configured in Step 1.

5. (Optional) Assign DiffServ levels (markings) to packets belonging to emergency calls:
   a. In the 'Emergency RTP DiffServ' field (SBCEmergencyRTPDiffServ), enter the QoS level for RTP packets.
   b. In the 'Emergency Signaling DiffServ' field (SBCEmergencySignalingDiffServ), enter the QoS level for SIP signaling packets.

6. Click **Apply**.

### 31.2 Emergency Call Routing using LDAP to Obtain ELIN

The device can route emergency calls (e.g., 911) for INVITE messages that are received without an ELIN number. This is in contrast to when the device is deployed in a Microsoft Skype for Business environment, whereby INVITE messages received from Skype for Business contain ELIN numbers. For a detailed explanation on ELIN numbers and handling of emergency calls by emergency server providers, see "E9-1-1 Support for Microsoft Skype for Business" on page 288.

To obtain an ELIN number for emergency calls received without ELINs, you can configure the device to query an LDAP server for the 911 caller's ELIN number. The device adds the resultant ELIN number and a Content-Type header for the PIDF XML message body to the outgoing INVITE message, for example:

```
Content-Type: application/pidf+xml
<NAM>1234567890</NAM>
```
To enable emergency call routing using LDAP to obtain ELIN:

1. Configure a Call Setup rule in the Call Setup Rules table (see "Configuring Call Setup Rules" on page 405). The following example shows a Call Setup rule that queries an Active Directory (AD) server for the attribute "telephoneNumber" whose value is the E9-1-1 caller's number (source), and then retrieves the user's ELIN number from the attribute "numberELIN".

   ![Figure 31-3: Example of Call Setup Rule for LDAP Query for ELIN](image)

2. Enable the E9-1-1 feature, by configuring the 'PSAP Mode' parameter to PSAP Server in the IP Groups table for the IP Group of the PSAP server (see "Enabling the E9-1-1 Feature" on page 298).

3. Configure routing rules in the IP-to-IP Routing table for routing between the emergency callers' IP Group and the PSAP server's IP Group. The only special configuration required for the routing rule from emergency callers to the PSAP server:
   - Configure the emergency number (e.g., 911) in the 'Destination Username Prefix' field.
   - Assign the Call Setup rule that you configured for obtaining the ELIN number from the AD (see Step 1) in the 'Call Setup Rules Set ID' field (see "Configuring SBC IP-to-IP Routing Rule for E9-1-1" on page 298).
31.3 Enabling Interworking of SIP and SIP-I Endpoints

The device can interwork between SIP and SIP-I endpoints for SBC calls. SIP-I endpoints are entities that are connected to the SS7 PSTN network, referred to as the ISDN User Part (ISUP) domain. The device supports the SIP-I Application-layer signaling protocol, which is the standard for encapsulating a complete copy of the SS7 ISUP message in SIP messages, according to ITU-T Q.1912.5, Interworking between Session Initiation Protocol (SIP) and Bearer Independent Call Control protocol or ISDN User Part. In other words, SIP-I is SIP encapsulated with ISUP and the interworking is between SIP signaling and ISUP signaling. This allows you to deploy the device in a SIP environment where part of the call path involves the PSTN.

The SIP-I sends calls, originating from the SS7 network, to the SIP network by adding ISUP messaging in the SIP INVITE message body. The device can receive such a message from the SIP-I and remove the ISUP information before forwarding the call to the SIP endpoint. In the other direction, the device can receive a SIP INVITE message that has no ISUP information and before forwarding it to the SIP-I endpoint, create a SIP-I message by adding ISUP information in the SIP body. For SIP-I to SIP-I calls, the device can pass ISUP data transparently between the endpoints.

For the interworking process, the device maps between ISUP data (including cause codes) and SIP headers. For example, the E.164 number in the Request-URI of the outgoing SIP INVITE is mapped to the Called Party Number parameter of the IAM message, and the From header of the outgoing INVITE is mapped to the Calling Party Number parameter of the IAM message.

The ISUP data is included in SIP messages using the Multipurpose Internet Mail Extensions (MIME) body part, for example (some headers have been removed for simplicity):

```
INVITE sip:1774567@172.20.1.177;user=phone SIP/2.0
Via: SIP/2.0/UDP 172.20.73.230:5060;branch=z9hG4bK.iI
Accept: application/sdp, application/isup, applicatio
Content-Type: multipart/mixed; boundary=unique-bounda
MIME-Version: 1.0
Content-Length: 350

Content-Type: application/isup; version=FTSSURI; base
Content-Disposition: signal; handling=required
01 00 40 01 0a 02 02 08 06 83 10 71 47 07 08
01 00 00
--unique-boundary-1--
D6 SIP-T ISUP/IAM (Initial address message)
--> len:>> Nature of connection indicators
Oct 1 : 00-- Echo ctrl = Half echo not included
----00-- Cont. check = Not required
-------00 Satellite = No circuit
--> len:>> Forward call indicators
Oct 1 : 01----- ISUP pref. = Not req. all the way
```
ISUP data, received in the MIME body of the incoming SIP message is parsed according to the ISUP variant (SPIROU itu or ansi), indicated in the SIP Content-Type header. The device supports the following ISUP variants (configured by the 'ISUP Variant' parameter in the IP Profile table):

- **French (France) specification, SPIROU (Système Pour l'Interconnexion des Réseaux OUverts),** which regulates Telecommunication equipment that interconnect with networks in France. For SPIROU, the device sets the value of the SIP Content-Type header to "version=spirou; base=itu-t92+".

- **ITU-92, where the device sets the value of the SIP Content-Type header to "version=itu-t92+; base=itu-t92+".**

To configure interworking of SIP and SIP-I endpoints, using the 'ISUP Body Handling' parameter (IpProfile_SBCISUPBodyHandling) in the IP Profile table (see "Configuring IP Profiles" on page 417).

You can manipulate ISUP data, by configuring manipulation rules for the SIP Content-Type and Content-Disposition header values in the Message Manipulations table (see "Configuring SIP Message Manipulation" on page 393). For a complete description of the ISUP manipulation syntax, refer to the *SIP Message Manipulation Reference Guide*. In addition, you can use AudioCodes proprietary SIP header X-AC-Action in Message Manipulation rules to support the following call actions (e.g., SIP-I SUS and RES messages) for the ISUP SPIROU variant:

- Disconnect call (optionally, after a user-defined time): `disconnect[;delay=<time in ms>]`
- Resume previously suspended call: `abort-disconnect`
- Reply to the message with a SIP response without forwarding the response to the other side: `reply[;response=<response code, e.g., 200>]`
- Switch IP Profile for the call (re-INVITE only), as defined in the IP Group: `switch-profile[;reason=<reason - PoorInVoiceQuality or PoorInVoiceQualityFailure >]`
31.4 WebRTC

The device supports interworking of Web Real-Time Communication (WebRTC) and SIP-based VoIP communication. The device interworks WebRTC calls made from a Web browser (WebRTC client) and the SIP destination. The device provides the media interface to WebRTC.

WebRTC is a browser-based real-time communication protocol. WebRTC is an open source, client-side API definition (based on JavaScript) drafted by the World Wide Web Consortium (W3C) that supports browser-to-browser applications for voice calling (video chat, and P2P file sharing) without plugins. Currently, WebRTC is supported only by Mozilla Firefox and Google Chrome Web browsers. Though the WebRTC standard has obvious implications for changing the nature of peer-to-peer communication, it is also an ideal solution for customer-care solutions to allow direct access to the contact center. An example of a WebRTC application is a click-to-call button on a consumer Web site (see following figure). After clicking the button, the customer can start a voice and/or video call with a customer service personnel directly from the browser without having to download any additional software plugins. The figure below displays an example of a click-to-call application from a customer Web page, where the client needs to enter credentials (username and password) before placing the call.

Figure 31-5: Example of WebRTC for Click-to-Call Application

Note: The WebRTC feature is a license-dependent feature and is available only if it is included in the License Key that is installed on the device. For ordering the feature, please contact your AudioCodes sales representative.

The WebRTC standard requires the following mandatory components, which are supported by the device:
Voice coders: Narrowband G.711 and wideband Opus (Version 1.0.3, per RFC 6176).

Video coders: VP8 video coder. The device transparently forwards the video stream, encoded with the VP8 coder, between the endpoints.

ICE (per RFCs 5389/5245): Resolves NAT traversal problems, using STUN and TURN protocols to connect peers. For more information, see "ICE Lite".

DTLS-SRTP (RFCs 5763/5764): Media channels must be encrypted (secured) through Datagram Transport Layer Security (DTLS) for SRTP key exchange. For more information, see "SRTP using DTLS Protocol" on page 207.

SRTP (RFC 3711): Secures media channels by SRTP.

RTP Multiplexing (RFC 5761): Multiplexing RTP data packets and RTCP control packets onto a single port for each RTP session. For more information, see "Interworking RTP-RTCP Multiplexing".

Secure RTCP with Feedback (i.e., RTP/SAVPF format in the SDP - RFC 5124): Combines secured voice (SRTP) with immediate feedback (RTCP) to improve session quality. The SRTP profile is called SAVPF and must be in the SDP offer/answer (e.g., "m=audio 11050 RTP/SAVPF 103"). For more information, see the IP Profile parameter, IPProfile_SBCRTCPFeedback (see "Configuring IP Profiles" on page 417).

WebSocket: WebSocket is a signaling (SIP messaging) transport protocol, providing full-duplex communication channels over a single TCP connection for Web browsers and clients. SIP messages are sent to the device over the WebSocket session. For more information, see "SIP over WebSocket" on page 568.

For more information on WebRTC, go to http://www.webrtc.org/. Below shows a summary of the WebRTC components and the device’s interworking of these components between the WebRTC client and the SIP user agent:
1. The WebRTC client uses a Web browser to visit the Web site page.
3. When the client clicks the Call button or call link, the browser runs the JavaScript code which sends the HTTP upgrade request for WebSocket in order to establish a WebSocket session with the device. The address of the device is typically included in the JavaScript code.
4. A WebSocket session is established between the WebRTC client and the device in order for the WebRTC client to register with the device. This is done using a SIP REGISTER message sent over the WebSocket session (SIP over WebSocket). Registration can be initiated when the client enters credentials (username and password) on the Web page or it can be done automatically when the client initially browses to the page. This depends on the design of the Web application (JavaScript).
5. Once registered with the device, the client can receive or make calls, depending on the Web application.
6. To make a call, the client clicks the call button or link on the Web page.
7. Negotiation of a workable IP address between the WebRTC client and the device is done through ICE.
8. Negotiation of SRTP keys using DTLS is done between WebRTC and the client on the media.
9. Media flows between the WebRTC client and the SIP client located behind the device.

31.4.1 SIP over WebSocket

The device supports the transmission of SIP signaling over WebSocket. WebSocket is a protocol providing real-time, full-duplex (two-way) communication over a single TCP connection (socket) between a Web browser or page (client) and a remote host (server). This is used for browser-based applications such as click-to-call from a Web page. As
WebSocket has been defined by the WebRTC standard as mandatory, its support by the device is important for deployments implementing WebRTC.

A WebSocket connection starts as an HTTP connection between the Web client and the server, guaranteeing full backward compatibility with the pre-WebSocket world. The protocol switch from HTTP to WebSocket is referred to as the WebSocket handshake, which is done over the same underlying TCP/IP connection. A WebSocket connection is established using a handshake between the Web browser (WebSocket client) and the server (i.e., the device). The browser sends a request to the server, indicating that it wants to switch protocols from HTTP to WebSocket. The client expresses its' desire through the Upgrade header (i.e., upgrade from HTTP to WebSocket protocol) in an HTTP GET request, for example:

```plaintext
GET /chat HTTP/1.1
Upgrade: websocket
Connection: Upgrade
Host: <IP address:port of SBC device>
Sec-WebSocket-Protocol: SIP
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: <server that provided JavaScript code to browser, e.g., http://domain.com>
Sec-WebSocket-Version: 13
```

If the server understands the WebSocket protocol, it agrees to the protocol switch through the Upgrade header in an HTTP 101 response, for example:

```plaintext
HTTP/1.1 101 Switching Protocols
Upgrade: WebSocket
Connection: Upgrade
Sec-WebSocket-Accept: rLHCkw/8Ks09Gah/7SFhBATDKrU=
Sec-WebSocket-Protocol: SIP
Server: SBC
```

At this stage, the HTTP connection breaks down and is replaced by a WebSocket connection over the same underlying TCP/IP connection. By default, the WebSocket connection uses the same ports as HTTP (80) and HTTPS (443).

Once a WebSocket connection is established, the SIP messages are sent over the WebSocket session. The device, as a "WebSocket gateway" or server can interwork WebSocket browser originated traffic to SIP over UDP, TCP or TLS, as illustrated below:

```
```

The SIP messages over WebSocket are indicated by the "ws" value, as shown in the example below of a SIP REGISTER request received from a client:

```plaintext
REGISTER sip:10.132.10.144 SIP/2.0
Via: SIP/2.0/WS v6iqlt81ne5c.invalid;branch=z9hG4bK778566
Max-Forwards: 69
To: <sip:101@10.132.10.144>
From: "joe" <sip:101@10.132.10.144>;tag=ub50pqjgpr
Call-ID: fhddgc3kc3hhu32h01fgh1
```
To keep a WebSocket session alive, it is sometimes necessary to send regular messages to indicate that the channel is still being used. Some servers, browsers or proxies may close an idle connection. The Ping-Pong WebSocket messages are designed to send non-application level traffic that prevents the channel from being prematurely closed. You can configure how often the device pings the WebSocket client, using the WebSocketProtocolKeepAlivePeriod parameter (see "Configuring WebRTC" on page 570). The device always replies to ping control messages with a pong message.

**Note:** When the device operates in High-Availability (HA) mode, if a WebSocket connection has been established and a switchover subsequently occurs, the WebSocket session is not copied to the redundant device. As Chrome does not renew the WebSocket connection with the device, WebRTC calls remain open indefinitely; the Chrome side will stop the call, but the device will keep all of the call's resources open and the other side will have an active call with no voice. To prevent this, for the IP Profile associated with the WebRTC clients, configure the 'Broken Connection Mode' parameter to Disconnect.

### 31.4.2 Configuring WebRTC

To support WebRTC, you need to perform special configuration settings for the device's SBC leg interfacing with the WebRTC client (i.e., Web browser), as described in the following procedure.

For the WebRTC deployment environment, you need to install a signed certificate by a Certificate Authority (CA) on your Web server machine (hosting the WebRTC JavaScript) and on your AudioCodes SBC device (i.e., WebSocket server).

**Note:**
- Google announced a security policy change that impacts new versions of the Chrome Web browser. Any Web site that has integrated WebRTC, geolocation technology, screen-sharing and more, now requires to be served from a secure (HTTPS) site, including WebRTC-based WebSocket servers (WSS instead of WS). The configuration described below accommodates for this basic requirement.
- WebRTC JavaScript configuration is beyond the scope of this document.

#### To configure WebRTC:

1. Configure a TLS Context (certification):
   a. Open the TLS Contexts table (see "Configuring TLS Certificate Contexts" on page 105).
   b. Add a new TLS Context (e.g., "WebRTC") or edit an existing one and configure the DTLS version (TLSContexts_DTLSVersion).
   c. Create a certificate signing request (CSR) to request a digitally signed certificate from a Certification Authority (CA).
d. Send the CSR to the CA for signing.

e. When you have received the signed certificate, install it on the device as the "Device Certificate" and install the CA's root certificate into the device's trusted root store ("Trusted Certificates").

For more information on CSR, see "Assigning CSR-based Certificates to TLS Contexts" on page 109.

2. Configure the keep-alive interval with the WebSocket client:

   a. On the Transport Settings page (Setup menu > Signaling & Media tab > SIP Definitions folder > Transport Settings), and then in the 'WebSocket Keep-Alive Period' field (WebSocketProtocolKeepAlivePeriod), enter the keep-alive interval:

      Figure 31-6: Configuring Keep-alive with WebSocket Client

      WebSocket Keep-Alive Period [sec] 0

   b. Click Apply.

3. Configure a SIP Interface for the WebRTC clients that identifies WebSocket traffic:

   a. Open the SIP Interfaces table (see "Configuring SIP Interfaces" on page 337).

   b. Do the following:

      ♦ From the 'Encapsulating Protocol' drop-down list (SIPInterface_EncapsulatingProtocol), select WebSocket.

      ♦ In the 'TLS Port' field, configure the TLS port.

      ♦ From the 'TLS Context Name' drop-down list, assign the TLS Context that you configured in Step 1 (e.g., "WebRTC").

      Figure 31-7: Configuring SIP Interface for WebRTC Clients

      c. Click Apply.
4. Configure an IP Profile for the WebRTC clients:
   a. Open the IP Profiles table (see "Configuring IP Profiles" on page 417).
   b. Do the following:
      ♦ From the ‘ICE Mode’ drop-down list (IPProfile_SBCIceMode), select **Lite** to enable ICE.
      ♦ From the ‘RTCP Mux’ drop-down list (IPProfile_SBCRTCPMux), select **Supported** to enable RTCP multiplexing.
      ♦ From the ‘RTCP Feedback’ drop-down list (IPProfile_SBCRTCPFeedback), select **Feedback On** to enable RTCP feedback.

   ![Figure 31-8: Configuring WebRTC-related Parameters for IP Profile](image)

   c. Click **Apply**.

5. Configure an IP Group for the WebRTC clients:
   a. Open the IP Groups table (see "Configuring IP Groups" on page 345).
   b. Do the following:
      ♦ From the ‘Type’ drop-down list, select **User**.
      ♦ From the ‘IP Profile’ drop-down list, select the IP Profile that you configured for the WebRTC clients in Step 3 (e.g., "WebRTC").
      ♦ From the ‘DTLS Context’ drop-down list, select the TLS Context that you configured in Step 1. For more information on DTLS, see "SRTP using DTLS Protocol" on page 207.

   ![Figure 31-9: Configuring IP Group for WebRTC Clients](image)

6. Configure IP-to-IP routing rules to route calls between the WebRTC clients and the enterprise:

b. Configure routing rules for the following call scenarios:
   ♦ Call routing from WebRTC clients (IP Group configured in Step 4) to the enterprise.
   ♦ Call routing from the enterprise to the WebRTC clients (IP Group configured in Step 4).

31.5 Call Forking

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

31.5.1 Initiating SIP Call Forking

The SBC device supports call forking of an incoming call to multiple SBC users (destinations). Call forking is supported by the device's capability of registering multiple SIP client user phone contacts (mobile and fixed-line extensions) under the same Address of Record (AOR) in its registration database. This feature can be implemented in the following example scenarios:

- An enterprise Help Desk, where incoming customer calls are simultaneously sent to multiple customer service agent extensions.
- An employee's phone devices, where the incoming call is simultaneously sent to multiple devices (e.g., to the employee's office phone and mobile SIP phone).
- An enterprise reception desk, where an incoming call is simultaneously sent to multiple receptionists.

The device supports various modes of call forking. For example, in Parallel call forking mode, the device sends the INVITE message simultaneously to all the users registered under the same AOR, resulting in the ringing of all extensions; the first extension to pick up the call receives the call, and all other extensions stop ringing. The Call Forking feature is configured by creating a User-type IP Group and configuring the IP Groups table's parameter, "SBC Client Forking Mode" (see "Configuring IP Groups" on page 345).

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.

31.5.2 Configuring SIP Forking Initiated by SIP Proxy

The device can handle the receipt of multiple SIP 18x responses as a result of SIP forking initiated by a proxy server. This occurs when the device forwards an INVITE, received from a user agent (UA), to a proxy server and the proxy server then forks the INVITE request to multiple UAs. Several UAs may answer and the device may therefore, receive several replies (responses) for the single INVITE request. Each response has a different 'tag' value in the SIP To header.

During call setup, forked SIP responses may result in a single SDP offer with two or more SDP answers. The device "hides" all the forked responses from the INVITE-initiating UA, except the first received response ("active" UA) and it forwards only subsequent requests and responses from this active UA to the INVITE-initiating UA. All requests/responses from the other UAs are handled by the device; SDP offers from these UAs are answered with an "inactive" media.

The device supports two forking modes:

- **Latch On First**: The device forwards only the first received 18x response to the
INVITE-initiating UA and disregards subsequently received 18x forking responses (with or without SDP).

- Sequential: The device forwards all 18x responses to the INVITE-initiating UA, sequentially (one after another). If 18x arrives with an offer only, only the first offer is forwarded to the INVITE-initiating UA.

To configure the call forking mode:

1. Open the SBC General Settings page (Setup menu > Signaling & Media tab > SBC folder > SBC General Settings).
2. From the 'Forking Handling Mode' drop-down list (SBCForkingHandlingMode), select the required mode:

   *Latch On First*

3. Click **Apply**.

The device also supports media synchronization for call forking. If the active UA is the first one to send the final response (e.g., 200 OK), the call is established and all other final responses are acknowledged and a BYE is sent if needed. If another UA sends the first final response, it is possible that the SDP answer that was forwarded to the INVITE-initiating UA is irrelevant and thus, media synchronization is needed between the two UAs. Media synchronization is done by sending a re-INVITE request immediately after the call is established. The re-INVITE is sent without an SDP offer to the INVITE-initiating UA. This causes the INVITE-initiating UA to send an offer which the device forwards to the UA that confirmed the call. Media synchronization is enabled by the EnableSBCMediaSync parameter.

### 31.5.3 Configuring 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 501.

### 31.6 Call Survivability

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

#### 31.6.1 Enabling Auto-Provisioning of Subscriber-Specific Information of 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. The feature enables local users to dial a local extension (or any other configured alias) that identifies another local user, in survivability mode.

In normal operation, when subscribers (such as IP phones) register with 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 URLs (aliases), as shown in the example below:
<?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>

The device forwards the 200 OK to the subscriber (without the XML body). The call flow is shown below:

**Figure 31-10: 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 communication with the BroadWorks server is lost. 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.

➢ **To enable the BroadWorks survivability feature:**

1. Open the SBC General Settings page (Setup menu > Signaling & Media tab > SBC folder > SBC General Settings).
2. From the 'BroadWorks Survivability Feature’ drop-down list (SBCExtensionsProvisioningMode), select Enable:

   BroadWorks Survivability Feature  
   - Enable

3. Click Apply.

### 31.6.2 Configuring BroadSoft's Shared Phone Line Call Appearance for 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.

The 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 "Configuring SIP Forking Initiated by SIP Proxy" on page 573. Note that incoming calls specific to extensions 601 or 602 ring only at these specific extensions.
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.

**Note:**
- 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 BroadSoft's Shared Line feature:**

1. In the IP Groups table (see "Configuring IP Groups" on page 345), add a Server-type IP Group for the BroadWorks server.
2. In the IP Groups 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 501), add a rule for routing calls between the above configured IP Groups.
4. In the Inbound Manipulations table (see "Configuring IP-to-IP Inbound Manipulations" on page 531), 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):
   - 'Manipulation Purpose': **Shared Line**
   - Match:
     - 'Request Type': REGISTER
     - 'Source IP Group': IP Group created for the users (e.g., 2)
     - 'Source Username Prefix': Represents the secondary extensions, e.g., 601 and 602
   - Action:
     - 'Manipulated URI': **Source** (manipulates the source URI)
     - 'Remove From Right': "1" (removes the last digit of the extensions, e.g., 601 is changed to 60)
     - 'Suffix to Add': "0" (adds 0 to the end of the manipulated number, e.g., 60 is changed to 600).
31.6.3 Configuring Call Survivability for Call Centers

The device supports call survivability for call centers. When a communication failure (e.g., in the network) occurs with the remote voice application server responsible for handling the call center application (such as IVR), the device routes the incoming calls received from the customer (i.e., from the TDM gateway) to the call center agents.

In normal operation, the device registers the agents in its users registration database. Calls received from the TDM gateway are forwarded by the device to the application server, which processes the calls and sends them to specific call center agents, through the device. Upon a failure with the application server, the device routes the calls from the TDM Gateway to the agents. The device routes the call to the first available user it finds. If the call is not answered by the user, the device routes it to the next available user. The SBC can handle a sequence of up to five users, after which the session is timed out and the call is dropped.

Figure 31-12: Normal Operation in Call Center Application
To configure call survivability for a call center application:

1. In the IP Groups table (see "Configuring IP Groups" on page 345), 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 493), 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 501), add the following IP-to-IP routing rules:
   - For normal operation:
     - Routing from TDM Gateway to Application server.
     - Routing from Application server to call center agents.
   - For call survivability mode: Routing from TDM Gateway to call center agents. This configuration is unique due to the following settings:
     - The 'Source IP Group' field is set to the IP Group of the TDM Gateway.
     - The 'Destination Type' field is set to Hunt Group, which is specifically used for call center survivability.
     - The 'Destination IP Group' field is set to the IP Group of the call center agents.
The figure below displays a routing rule example, assuming IP Group "1" represents the TDM Gateway and IP Group "3" represents the call center agents:

**Figure 31-14: Routing Rule Example for Call Center Survivability**

### 31.6.4 Enabling Survivability 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, "Stand Alone Mode" on their LCD screens.

If you enable the feature and the 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:

```xml
<?xml version="1.0" encoding="utf-8"?>
<LMIDocument version="1.0">
  <LocalModeStatus>
    <LocalModeActive>true</LocalModeActive>
    <LocalModeDisplay>StandAlone Mode</LocalModeDisplay>
  </LocalModeStatus>
</LMIDocument>
```

**To enable survivability display on Aastra phones:**

1. Load an ini file to the device that includes the following parameter setting:

```
SBCEnableSurvivabilityNotice = 1
```
31.7 Alternative Routing on Detection of Failed SIP Response

The device can detect failure of a sent SIP response (e.g., TCP timeout, and UDP ICMP). In such a scenario, the device re-sends the response to an alternative destination. This support is in addition to alternative routing if the device detects failed SIP requests. For example, assume the device sends a SIP 200 OK in response to a received INVITE request. If the device does not receive a SIP ACK in response to this, it sends a new 200 OK to the next alternative destination. This new destination can be the next given IP address resolved from a DNS from the Contact or Record-Route header in the request related to the response.

31.8 VolPerfect

AudioCodes VolPerfect™ feature combines the device's Access and Enterprise SBC technology to ensure high speech (call) quality (MOS) between the Enterprise SBC and the Access SBC (located at the Internet service provider / ISP) during periods of adverse WAN network conditions (such as packet loss and bandwidth reduction). VolPerfect adapts itself to current network conditions. Before adverse WAN network conditions can affect the quality of the call, VolPerfect employs sophisticated technology using the Opus coder (as later explained in this section) to ensure that high call quality is maintained.

VolPerfect guarantees that 95% of your calls will achieve a Perceptual Evaluation of Speech Quality (PESQ) score greater than or equal to 3.6 if the summation of bandwidth overuse and packet loss is less than or equal to 25%. ISPs can therefore offer service level agreements (SLAs) to their customers based on the VolPerfect feature. For more information, contact your AudioCodes sales representative. In addition, by ensuring high call quality even in adverse network conditions, VolPerfect may reduce costs for ISPs such as SIP trunk providers and Unified Communications as a Service (UCaaS) by eliminating the need for dedicated WAN links (such as MPLS and leased links) and instead, allow the use of standard broadband Internet connections. However, it can also be used in tandem with existing infrastructure.

VolPerfect uses Temporary Maximal Media Stream Bit Rate (TMMBR) negotiation capabilities for Opus coders. Through TMMBR, VolPerfect can receive indications of network quality and dynamically change the coder's payload bit rate accordingly during the call to improve voice quality. TMMBR is an RTCP feedback message (per RFC 4585) which enables SIP users to exchange information regarding the current bit rate of the media stream. The information can be used by the receiving side to change the media stream parameters (e.g., coder rate or coder) to enhance voice quality. TMMBR is negotiated in the SDP Offer/Answer model using the 'tmbr' attribute and following syntax:

\[a=rtcp-fb:<payload type> ccm tmmbr smaxpr=<sent TMMBR packets]\]

VolPerfect also supports the SDP attribute 'a=rtcp-rsize', which reduces the RTCP message size (RFC 5506). As feedback messages are frequent and take a lot of bandwidth, the attribute attempts to reduce the RTCP size. The attribute can only be used in media sessions defined with the AVPF profile and must also be included in sessions supporting TMMBR; otherwise, the call is rejected.

VolPerfect supports two modes of operation, where the Access SBC can be configured to support both modes and each Enterprise SBC serviced by the Access SBC can be configured to support one of the modes:

- Managed Opus: If the SBC detects WAN network impairments during a call using the Opus coder between the Enterprise SBC and Access SBC, it can adjust the Opus coder's attributes (e.g., bit rate) for that specific call to ensure high voice quality is maintained. The advantage of the Opus coder is that its' bit rate can change dynamically according to bandwidth availability. This mode is useful for unstable networks, allowing Opus to dynamically adapt to adverse network conditions.
Configuration of the Enterprise SBC:

- Coders Group with Opus
- Allowed Audio Coders Group with Opus
- IP Profile:
  - Extension Coders Group: Coders Group with Opus
  - Allowed Audio Coders: Allowed Audio Coders Group with Opus
  - Allowed Coders Mode: Restriction
  - Voice Quality Enhancement: Enable
  - RTCP Feedback: Feedback On
  - Max Opus Bandwidth: 0

**Smart Transcoding:** If the SBC (Enterprise or Access) detects WAN network impairments during a call between the Enterprise SBC and Access SBC, the SBC employs voice transcoding by switching the coder from G.711 to Opus for that specific call only. Transcoding is done only on the path between the Enterprise SBC and Access SBC. As Smart Transcoding is applied only on a per call basis, it preserves valuable DSP resources that may be required for other functionalities. An advantage of using the Opus coder is that it consumes less bandwidth than G.711 and overcomes packet loss (by dynamic packet redundancy), allowing the SBC to support more concurrent calls than with G.711 for the same bandwidth. This mode is useful for WAN networks that are relatively stable, allowing the use of G.711 whenever possible and switching to Opus only during adverse network conditions.

Configuration of the Enterprise SBC:

- Device’s License Key includes the SBC transcoding feature
- Coder Groups (see Configuring Coder Groups on page 407):
  - Coders Group with G.711
  - Coders Group with Opus
- Allowed Audio Coders Groups (see Configuring Allowed Audio Coder Groups on page 412):
  - Allowed Audio Coders Group with G.711
  - Allowed Audio Coders Group with Opus
- Main IP Profile (see Configuring IP Profiles on page 417):
  - Extension Coders Group: Coders Group with G.711
  - Allowed Audio Coders: Allowed Audio Coders Group with G.711
Allowed Coders Mode: Restriction
RTCP Feedback: Feedback On
Voice Quality Enhancement: Enable

Alternative IP Profile:
- Extension Coders Group: Coders Group with Opus
- Allowed Audio Coders: Allowed Audio Coders Group with Opus
- Allowed Coders Mode: Restriction
- RTP Redundancy Mode: Enable
- RTCP Feedback: Feedback On
- Voice Quality Enhancement: Enable
- Max Opus Bandwidth: 80000

Quality of Service Rules (see Configuring Quality of Service Rules on page 316):
- Rule Metric: Poor InVoice Quality
- Alternative IP Profile Name: name of Alternative IP Profile (above)

Configuration of the Access SBC for both methods:

Coder Groups:
- Coders Group with G.711 and Opus
- Coders Group with Opus

Allowed Audio Coders Group with Opus

IP Profile:
- Extension Coders Group: Coders Group with G.711 and Opus
- Voice Quality Enhancement: Enable
- RTP Redundancy Mode: Enable
- RTCP Feedback: Feedback On
- Max Opus Bandwidth: 0

Alternative IP Profile:
- Extension Coders Group: Coders Group with Opus
- Allowed Audio Coders: Allowed Audio Coders Group with Opus
- Allowed Coders Mode: Restriction
- Voice Quality Enhancement: Enable
- RTP Redundancy Mode: Enable
- RTCP Feedback: Feedback On
- Max Opus Bandwidth: 0

Quality of Service Rules (see Configuring Quality of Service Rules on page 316):
- Rule Metric: Poor InVoice Quality
- Alternative IP Profile Name: name of Alternative IP Profile (above)
Note:

- VolPerfect is applicable only to G.711 calls.
- If you are deploying a third-party device between the Enterprise SBC and Access SBC, make sure that the third-party device adheres to the following:
  - ✓ Enable RFC 2198 in SDP negotiation
  - ✓ Enable TMMBR in SDP negotiation
  - ✓ Forward the SDP with feedback (SAVPF) as is
  - ✓ Forward TMMBR messages as is
  - ✓ Forward RTCP messages as is (not terminate them)
  - ✓ (Smart Transcoding only) Forward re-INVITE messages for using Opus as is
  - ✓ (Smart Transcoding only) Forward the SIP header, X-Ac-Action as is
This page is intentionally left blank.
Part VI

Cloud Resilience Package
32 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. 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.

Note:
- The CRP application is applicable only to Mediant VE SBC.
- The CRP application is available only if the device is installed with a License Key that includes this feature. For installing a License Key, see "License Key" on page 669.
- 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 965.

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>
<thead>
<tr>
<th>Survivability</th>
<th>Quality of Experience/Service</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSTN fallback</td>
<td>QoE monitoring</td>
<td>Layer 3 to 7 protection</td>
</tr>
<tr>
<td>WAN redundancy</td>
<td>Call Admission Control</td>
<td>Media encryption</td>
</tr>
<tr>
<td>Local mode</td>
<td>SLA fulfillment</td>
<td>Call control encryption</td>
</tr>
<tr>
<td>High availability</td>
<td>SIP mediation</td>
<td>NAT traversal</td>
</tr>
<tr>
<td>Emergency calling (E911)</td>
<td>Media transcoding</td>
<td>Topology hiding</td>
</tr>
<tr>
<td>Basic call routing between</td>
<td>Test call agent</td>
<td></td>
</tr>
<tr>
<td>registering users and device,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>or any other route to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>responding server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Short number dialog (short</td>
<td></td>
<td></td>
</tr>
<tr>
<td>numbers are learned</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dynamically in the</td>
<td></td>
<td></td>
</tr>
<tr>
<td>registration process)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survivability indication to IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>phone</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Survivability</td>
<td>Quality of Experience/Service</td>
<td>Security</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>• Call hold and retrieve</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Call transfer (if IP phone initiates REFER)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Basic Shared Line Appearance (excluding correct busy line indications)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Call waiting (if supported by IP phone)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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.
33 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 Admission Control" on page 489
- "Configuring Allowed Audio Coder Groups" on page 412
- "Configuring Classification Rules" on page 493
- "Configuring Message Condition Rules" on page 501
- "Configuring SBC IP-to-IP Routing Rules" on page 501
- "Configuring SIP Response Codes for Alternative Routing Reasons" on page 519
- "Configuring IP-to-IP Inbound Manipulations" on page 531
- "Configuring IP-to-IP Outbound Manipulations" on page 535

**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".

33.1 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 License Key that includes this feature. For installing a License Key, see "License Key" on page 669.

➢ To enable the CRP application:

1. Open the Applications Enabling page (Setup menu > Signaling & Media tab > Core Entities folder > Applications Enabling).
2. From the 'CRP Application' drop-down list, select Enable.
3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

33.2 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 33-1: 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.
33. CRP Configuration

- **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 33-2: CRP in Always Emergency Mode](image)

  To configure the Call Survivability mode:
  1. Open the General Settings page (Setup menu > Signaling & Media tab > CRP folder > CRP General Settings).
  2. From the ‘CRP Survivability Mode’ drop-down list, select the required mode.
  3. Click Apply.

### 33.3 Pre-Configured IP Groups

For CRP, the device is pre-configured with the following IP Groups in the IP Groups table:

<table>
<thead>
<tr>
<th>Index</th>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;CRP Users&quot;</td>
<td>User</td>
<td>LAN users (e.g., IP phones) at the branch office</td>
</tr>
<tr>
<td>2</td>
<td>&quot;CRP Proxy&quot;</td>
<td>Server</td>
<td>Server (e.g., hosted IP PBX at the Enterprise's headquarters)</td>
</tr>
<tr>
<td>3</td>
<td>&quot;CRP Gateway&quot;</td>
<td>Server</td>
<td>Device's interface with the PSTN</td>
</tr>
</tbody>
</table>

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 592).

**Note:**
- 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 Groups table and for a description of its parameters, see "Configuring IP Groups" on page 345.
33.4 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 589.

Note:
- 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 501.

33.4.1 Normal Mode

The pre-configured IP-to-IP routing rules for the Normal CRP call survivability mode are shown in the table below:

<table>
<thead>
<tr>
<th>Index</th>
<th>Source IP Group / Emergency</th>
<th>Request Type</th>
<th>Destination Type</th>
<th>Destination IP Group</th>
<th>Destination Address</th>
<th>Alternative Route Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Any</td>
<td>OPTIONS</td>
<td>Dest Address</td>
<td>-</td>
<td>Internal</td>
<td>Route Row</td>
</tr>
<tr>
<td>3</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#2 [CRP Proxy]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td>4</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#1 [CRP Users]</td>
<td>-</td>
<td>Alternative Route Ignore Inputs</td>
</tr>
<tr>
<td>5</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#3 [CRP Gateway]</td>
<td>-</td>
<td>Alternative Route Ignore Inputs</td>
</tr>
<tr>
<td>6</td>
<td>#2 [CRP Proxy]</td>
<td>All</td>
<td>IP Group</td>
<td>#1 [CRP Users]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td>7</td>
<td>#2 [CRP Proxy]</td>
<td>All</td>
<td>IP Group</td>
<td>#3 [CRP Gateway]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td>8</td>
<td>#3 [CRP Gateway]</td>
<td>All</td>
<td>IP Group</td>
<td>#2 [CRP Proxy]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td>9</td>
<td>#3 [CRP Gateway]</td>
<td>All</td>
<td>IP Group</td>
<td>#1 [CRP Users]</td>
<td>-</td>
<td>Alternative Route Ignore Inputs</td>
</tr>
</tbody>
</table>

Note: Index 7 appears only if the CRPGatewayFallback parameter is enabled (see 'Configuring PSTN Fallback' on page 594). This routing rule is used if the device can't find a matching destination user for IP Group 1 (User-type IP Group) in its registration database. If the CRPGatewayFallback parameter is disabled and no matching user is found, the device rejects the call.
33.4.2 **Emergency Mode**

The pre-configured IP-to-IP routing rules for the Emergency CRP call survivability mode are shown in the table below:

**Table 33-3: Pre-Configured IP-to-IP Routing Rules for Emergency Mode**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Index</th>
<th>Source IP Group / Emergency</th>
<th>Request Type</th>
<th>Destination Type</th>
<th>Destination IP Group ID</th>
<th>Destination Address</th>
<th>Alternative Route Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always Emergency</td>
<td>1</td>
<td>Any</td>
<td>OPTIONS</td>
<td>Dest Address</td>
<td>-</td>
<td>Internal</td>
<td>Route Row</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#1 [CRP Users]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#3 [CRP Gateway]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>#3 [CRP Gateway]</td>
<td>All</td>
<td>IP Group</td>
<td>#1 [CRP Users]</td>
<td>-</td>
<td>Route Row</td>
</tr>
</tbody>
</table>

**Note:** The routing rule at Index 5 appears only if the CRPGatewayFallback parameter is enabled (1).

33.4.3 **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 33-4: Pre-Configured IP-to-IP Routing Rule for Auto Answer to Registrations Mode**

<table>
<thead>
<tr>
<th>Mode</th>
<th>Index</th>
<th>Source IP Group</th>
<th>Request Type</th>
<th>Destination Type</th>
<th>Destination IP Group</th>
<th>Destination Address</th>
<th>Alternative Route Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto Answer to Registrations</td>
<td>1</td>
<td>Any</td>
<td>OPTIONS</td>
<td>Dest Address</td>
<td>-</td>
<td>Internal</td>
<td>Route Row</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>Any</td>
<td>REGISTER</td>
<td>IP Group</td>
<td>#2 [CRP Proxy]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#1 [CRP Users]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#3 [CRP Gateway]</td>
<td>-</td>
<td>Alternative Route Ignore Inputs</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>#1 [CRP Users]</td>
<td>All</td>
<td>IP Group</td>
<td>#3 [CRP Gateway]</td>
<td>-</td>
<td>Alternative Route Ignore Inputs</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>#2 [CRP Proxy]</td>
<td>All</td>
<td>IP Group</td>
<td>#1 [CRP Users]</td>
<td>-</td>
<td>Route Row</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>#2 [CRP Proxy]</td>
<td>All</td>
<td>IP Group</td>
<td>#3 [CRP Gateway]</td>
<td>-</td>
<td>Route Row</td>
</tr>
</tbody>
</table>
### 33.5 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.

**Note:**
- 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).
34 HA Overview

The device's High Availability (HA) feature provides 1+1 system redundancy using two Mediant Software E-SBC devices. If failure occurs in the active device, a switchover occurs to the redundant device which takes over the call handling process. Thus the continuity of call services is ensured. All active calls (signaling and media) are maintained upon switchover.

The figure below illustrates the Active-Redundant HA devices under normal operation. Communication between the two devices is through a Maintenance interface, having a unique IP address for each device. The devices have identical software and configuration including network interfaces (i.e., OAMP, Control, and Media), and have identical local-port cabling of these interfaces.
The figure below shows two Virtual Machines -- Mediant VE SBCs -- running on different servers to work in an HA configuration:

**Figure 34-1: Mediant VE SBC HA - Virtual Network Setup**

The physical NICs used by the Mediant SBC VE virtual machine must not share traffic with other applications such as other virtual machines or the hypervisor itself. This also applies to the physical NICs used for the HA link because overloading these NICs may cause false switchovers.

### 34.1 Connectivity and Synchronization between Devices

In HA mode, the Ethernet connectivity between the two devices is through a special LAN interface on each device, referred to as the Maintenance interface. Each device has its own Maintenance interface with a unique address, and each device knows the Maintenance address of the other. The Maintenance interface can use a dedicated Ethernet port group or share the same Ethernet port group with the other network interface types (i.e., OAMP, Media, and Control).

When only one of the devices is operational it is in HA stand-alone state. This means that the device has no connectivity to the second device. When the second device is powered up, it recognizes the active device through the Maintenance network and acquires the HA redundant state. It then begins synchronizing for HA with the active device through the Maintenance network. During synchronization, the active device sends the redundant device its current configuration settings, including Auxiliary files. The active device also sends its software file (.cmp) if the redundant device is running a different software version. Once loaded to the redundant device, the redundant device reboots to apply the new configuration and/or software. This ensures that the two units are synchronized regarding configuration and software.
34. HA Overview

Note: If the active unit runs an earlier version (e.g., 7.0) than the redundant unit (e.g., 7.2), the redundant unit is downgraded to the same version as the active unit (e.g., 7.0).

Thus, under normal operation, one of the devices is in active state while the other is in redundant state, where both devices share the same configuration and software. Any subsequent configuration update or software upgrade on the active device is also done on the redundant device.

In the active device, all logical interfaces (i.e., Media, Control, OAMP, and Maintenance) are active. In the redundant device, only the Maintenance interface is active, which is used for connectivity to the active device. Therefore, management is done only through the active device. Upon a failure in the active device, the redundant device becomes active and activates all its logical interfaces exactly as was used on the active device.

34.2 Device Switchover upon Failure

When a failure occurs in the active device, a switchover occurs to the redundant device making it the new active device. Whether a switchover is later done back to the repaired failed device, depends on whether you have enabled the Preempt mode:

- **Enabled:** The Preempt mode specifies one of the device's as the "preferred" device. This is done by assigning different priority levels (1 to 10, where 1 is the lowest) to the two devices. Typically, you would configure the active device with a higher priority level (number) than the redundant device. The only factor that influences the configuration is which device has the greater number; the actual number is not important. For example, configuring the active with 5 and redundant with 4, or active with 9 and redundant with 2 both assign highest priority to the active device. Whenever the device with higher priority recovers from a failure, it first becomes the redundant device but then initiates a switchover to become the active device once again; otherwise, after recovery, it becomes the redundant device and remains as redundant. If you change the priority level of the redundant device to one that is higher than the active device and then reset the redundant device, a switchover occurs to the redundant device making it the active device and the "preferred" device. If both devices are configured with the same priority level, Preempt mode is disabled. Please see note below when using priority level 10.

- **Disabled:** A switchover is done only upon failure of the currently active device.

Failure detection by the devices is done by the constant keep-alive messages they send between themselves to verify connectivity. Upon detection of a failure in one of the devices, the following occurs:

- **Failure in active device:** The redundant device initiates a switchover. The failed device resets and the previously redundant device becomes the active device in stand-alone mode. If at a later stage this newly active device detects that the failed device has been repaired, the system returns to HA mode. If Preempt mode is enabled and the originally active device was configured with a higher priority, a switchover occurs to this device; otherwise, if it was configured with a lower priority (or Preempt mode was disabled), the repaired device is initialized as the redundant device.

- **Failure in redundant device:** The active device moves itself into stand-alone mode until the redundant device is returned to operation. If the failure in the redundant device is repaired after reset, it's initialized as the redundant device once again and the system returns to HA mode.

Connectivity failure triggering a switchover can include, for example, one of the following:

- **Loss of physical (link) connectivity:** If one or more physical network groups (i.e.,
Ethernet port pair) used for one or more network interfaces of the active device disconnects (i.e., no link) and these physical network groups are connected OK on the redundant device, a switchover occurs to the redundant device.

- **Loss of network (logical) connectivity**: No network connectivity, verified by keep-alive packets between the devices. This applies only to the Maintenance interface.

**Note:**
- Switchover triggered by loss of physical connectivity in one or more Ethernet Group is not done if the active device has been configured to a Preempt mode level of 10. In such a scenario, the device remains active.
- After HA switchover, the active device updates other hosts in the network about the new mapping of its Layer-2 hardware address to the global IP address, by sending a broadcast gratuitous Address Resolution Protocol (ARP) message.

### 34.3 Viewing HA Status on Monitor Web Page

You can view the status of the HA system on the Monitor page of the device's Web interface. The page provides a graphical display of both active and redundant devices, as shown below:

You can distinguish between active and redundant devices as follows:

- **Active device**:
  - Color of border surrounding device is green.
  - Title above device is "Active Device". The default name is "Device 1".

- **Redundant device**:
  - Color of border surrounding device is blue.
  - Title above device is "Redundant Device". The default name is "Device 2".

The Monitor page also displays the HA operational status of the device to which you are currently logged in. This is displayed in the 'HA Status' field under the Device Information:

- "Synchronizing": Redundant device is synchronizing with Active device
- "Operational": The device is in HA mode
- "Stand Alone": HA is configured, but the Redundant device is missing and HA is currently unavailable

You can change the name of each device, as described in the following procedure:
To define a name for the device:

1. Open the HA Settings page (Setup menu > IP Network tab > Core Entities folder > HA Settings).
2. In the ‘HA Device Name’ field, enter a name for the active device.
3. In the 'Redundant HA Device Name' field, enter a name for the redundant device.

![Figure 34-2: Configuring Device Names for HA](image)

4. Click Apply.

**Note:** Once the devices are running in HA mode, you can change the name of the redundant device, through the active device only, in the 'Redundant HA Device Name' field.
35 HA Configuration

This section describes HA configuration.

35.1 Initial HA Configuration

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

35.1.1 Network Topology Types and Rx/Tx Ethernet Port Group Settings

Initial HA configuration depends on how you want to deploy your HA system in the network. The Maintenance interface, which is used for the HA link between active and redundant device can use a dedicated Ethernet Device and Ethernet Group (port), or share the same Ethernet Device and Ethernet Group with other IP network interface types (such as OAMP, Media and Control). However, it is recommended that you configure the Maintenance interface with a dedicated Ethernet Device and Ethernet Group to separate it from other IP network interfaces.

If you want to separate the Maintenance interface from other interfaces, separation must also be done externally to the devices, either physically (i.e., different physical networks) or logically (using VLANs). When using VLANs, make sure that you use a different Ethernet Device for each IP network interface (see "Configuring Underlying Ethernet Devices" on page 134 and "Configuring IP Network Interfaces" on page 137).

Note:

- The Maintenance interface is used for heartbeats and data transfer from active to standby device and therefore, any short interval interruption in communication may cause undesired switchovers.
- If you assign the same Underlying Ethernet Device to all the IP network interfaces, logical separation of traffic may not occur.
The Maintenance interface can employ Ethernet port redundancy (recommended), by using two ports. This is enabled by configuring the Ethernet Group associated with the Maintenance interface with two ports. However, for Mediant Virtual Edition (VE), Ethernet port redundancy is not relevant as the virtual NIC ("port") is logical and always available. Therefore, it is sufficient to configure the Ethernet Group with only one port member. To employ Ethernet port redundancy, you need to configure the virtual switch of the hypervisor for Ethernet port redundancy (or bonding) with the physical NICs. Refer to your hypervisor's support material on how to do this.

**Figure 35-1: Connectivity of Maintenance Interfaces for Mediant VE HA**

The required receive (Rx) and transmit (TX) mode for the port pair in the Ethernet Group used by the Maintenance interface is as follows (not applicable to Mediant VE):

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

  **Figure 35-2: Rx/Tx Mode for Direct Connection**

- If the two devices are connected through two (or more) isolated LAN switches (i.e., packets from one switch cannot traverse the second switch), configure the mode to **2RX/2TX**:

  **Figure 35-3: Redundancy Mode for Two Isolated Switches**

- For Geographical HA (both units are located far from each other), **2Rx/1T*Tx** port mode.
35. HA Configuration

Connected to a port aggregation switch is the recommended option:

**Figure 35-4: Rx/Tx Mode for Geographical HA**

![Diagram of HA configuration](image)

**Note:**
- When two LAN switches are used, the LAN switches must be in the same subnet (i.e., broadcast domain).
- To configure Rx/Tx modes of the Ethernet ports, see "Configuring Ethernet Port Groups" on page 132.

### 35.1.2 Configuring the HA Devices

To initially configure the two devices comprising the HA system, do the following procedures listed in chronological order:

1. Configuring the first device for HA (see "Step 1: Configure the First Device" on page 606)
2. Configuring the second device for HA (see "Step 2: Configure the Second Device" on page 607)
3. Activating HA on the devices (see "Step 3: Initialize HA on the Devices" on page 608)

**Note:**
- The HA feature is available only if both devices are installed with a License Key that includes this feature. For installing a License Key, see "License Key" on page 669.
- The physical connections of the first and second devices to the network (i.e., Maintenance interface and OAMP, Control and Media interfaces) **must be identical**. This also means that the two devices must also use the same Ethernet Groups and the port numbers belonging to these Ethernet Groups. For example, if the first device uses Ethernet Group 1 (with ports 1 and 2), the second device must also use Ethernet Group 1 (with ports 1 and 2).
- Before configuring HA, determine the required network topology, as described in "Network Topology Types and Rx/Tx Ethernet Port Group Settings" on page 603.
- The Maintenance network should be able to perform a fast switchover in case of link failure and thus, Spanning Tree Protocol (STP) should not be used in this network; the Ethernet connectivity of the Maintenance interface between the two devices should be constantly reliable without any disturbances.
35.1.2.1 Step 1: Configure the First Device

The first stage is to configure the first device for HA, as described in the following procedure:

**Note:** During this stage, make sure that the second device is powered off or disconnected from the network.

➢ To configure the first device for HA:

1. Configure the network interfaces, including the default OAMP interface:
   a. If you are already connected to the SBC via keyboard and monitor, change the OAMP parameters to suit your networking scheme through CLI (refer to the Installation Manual).
   b. Connect to the SBC's Web interface with the newly assigned OAMP IP address.
   c. Open the IP Interfaces table (see "Configuring IP Network Interfaces" on page 137).
   d. Configure the Control and Media network interfaces, as required.
   e. Add the HA Maintenance interface (i.e., the MAINTENANCE Application Type).

   **Note:** Make sure that the Maintenance interface uses an Ethernet Device and Ethernet Group that is **not** used by any other IP network interface. The Ethernet Group is associated with the Ethernet Device, which is assigned to the interface.

The IP Interfaces table below shows an example where the Maintenance interface is configured with Ethernet Device "vlan 2" (which is associated with Ethernet Group "GROUP_2"), while the other interface is assigned "vlan 1" (associated with Ethernet Group "GROUP_1"):

![Figure 35-5: Configuring MAINTENANCE Interface](image)

2. If the connection is through a switch, the packets of both interfaces should generally be untagged. To do this, open the Ethernet Devices table (see "Configuring Underlying Ethernet Devices" on page 134 ), and then configure the 'Tagging' parameter to **Untagged** for the Ethernet Device assigned to the Maintenance interface. The figure below shows an example (highlighted) where VLAN 2 is configured as the Native (untagged) VLAN ID of the Ethernet Group "GROUP_2":

![Figure 35-6: Configuring Untagged VLAN for Maintenance and Other Interfaces](image)

3. Set the Ethernet port Tx / Rx mode of the Ethernet Group used by the Maintenance interface (see "Configuring Ethernet Port Groups" on page 132). The port mode depends on the type of Maintenance connection between the devices, as described in "Network Topology Types and Rx/Tx Ethernet Port Group Settings" on page 603. For Mediant VE, the Tx / Rx mode is 1 RX / 1 TX as the Ethernet Group is configured with only a single port (for more information, see Network Topology Types and Rx/Tx..."
4. Configure HA parameters:
   a. Open the HA Settings page (Setup menu > IP Network tab > Core Entities folder > HA Settings):

   ![Figure 35-7: HA Settings Page](image)

   b. In the 'HA Remote Address' field, enter the Maintenance IP address of the second device.

   c. (Optional) Enable the HA Preempt feature by configuring the 'Preempt Mode' parameter to Enable, and then setting the priority level of the device in the 'Preempt Priority' field. Make sure that you configure different priority levels for the two devices. Typically, you would configure the active device with a higher priority level (number) than the redundant device. The only factor that influences the configuration is which device has the greater number; the actual number is not important. For example, configuring the active with 5 and redundant with 4, or active with 9 and redundant with 2 both assign highest priority to the active device. Configuring the level to 10 does not cause a switchover upon Ethernet connectivity loss. For more information on the feature, see "Device Switchover upon Failure" on page 599.

5. Burn the configuration to flash **without** a reset.

6. Power down the device.

7. Configure the second device (see "Step 2: Configure the Second Device" on page 607).

### 35.1.2.2 Step 2: Configure the Second Device

Once you have configured the first device for HA, you can configure the second device for HA. As the configuration of the second device is similar to the first device, the following procedure briefly describes each step. For detailed configuration such as the path to the Web configuration pages, refer to the section on configuring the first device ("Step 1: Configure the First Device" on page 606).

**Note:** During this stage, ensure that the first device is powered off or disconnected from the network.

- To configure the second device for HA:
  1. Connect to the device in the same way as you did with the first device.
2. Configure the **same** OAMP, Media, and Control interfaces as configured for the first device.

3. Configure a Maintenance interface for this device. The IP address must be different to that configured for the Maintenance interface of the first device. The Maintenance interfaces of the devices must be in the same subnet.

4. Configure the **same** Ethernet Groups and VLAN IDs of the network interfaces as configured for the first device.

5. Configure the **same** Ethernet port Tx / Rx mode of the Ethernet Group used by the Maintenance interface as configured for the first device.

6. Configure HA parameters in the HA Settings page:
   a. In the 'HA Remote Address' field, enter the Maintenance IP address of the **first** device.
   b. (Optional) Enable the HA Preempt feature by configuring the 'Preempt Mode' parameter to **Enable**, and then setting the priority level of the device in the 'Preempt Priority' field. Make sure that you configure different priority levels for the two devices. Typically, you would configure the active device with a higher priority level (number) than the redundant device. The only factor that influences the configuration is which device has the greater number; the actual number is not important. For example, configuring the active with 5 and redundant with 4, or active with 9 and redundant with 2 both assign highest priority to the active device. Configuring the level to 10 does not cause a switchover upon Ethernet connectivity loss. For more information on the HA Preempt feature, see "Device Switchover upon Failure" on page 599.

7. Burn the configuration to flash **without** a reset.

8. Power down the device.


**35.1.2.3 Step 3: Initialize HA on the Devices**

Once you have configured both devices for HA as described in the previous sections, follow the procedure below to complete and initialize HA so that the devices become operational in HA. This last stage applies to both devices.

➢ **To initialize the devices for HA:**

1. Cable the devices to the network.

   **Note:** You must connect both ports (two) in the Ethernet Group of the Maintenance interface to the network (i.e., two network cables are used). This provides 1+1 Maintenance port redundancy.

2. Power up the devices; the redundant device synchronizes with the active device and updates its configuration according to the active device. The synchronization status is indicated as follows:
   - **Active device:** The Web interface's Monitor page displays "Synchronizing" in the 'HA Status' field.

   When synchronization completes, the redundant device resets to apply the received configuration and software. When both devices become operational in HA, the HA status is indicated as follows:
   - **Both devices:** The Web interface's Monitor page displays "Operational" in the 'HA Status' field.

3. Access the active device with its' OAMP IP address and configure the device as
required. For information on configuration done after HA is operational, see "Configuration while HA is Operational" on page 609.

35.2 Configuration while HA is Operational

When the devices are operating in HA state, subsequent configuration is as follows:

- All configuration, including HA, is done on the active device only.
- Non-HA configuration on the active device is automatically updated on the redundant device (through the Maintenance interface).
- HA-related configuration on the active device is automatically updated on the redundant device:
  - Maintenance interface:
    - Modified Maintenance interface address of the active device: The address is set as the new 'HA Remote Address' value on the redundant device.
    - Modified 'HA Remote Address' value on the active device: The address is set as the new Maintenance interface address on the redundant device (requires a device reset).
    - Modifications on all other Maintenance interface parameters (e.g., Default Gateway and VLAN ID): updated to the Maintenance interface on the redundant device.
  - 'Preempt Mode' parameter (requires a device reset).
  - 'Preempt Priority' parameter is set for the active device.
  - Modified 'Redundant Preempt Priority' value is set for the redundant device (requires a device reset).

Note: If the HA system is already in HA Preempt mode and you want to change the priority of the device, to ensure that system service is maintained and traffic is not disrupted, it is recommended to set the higher priority to the redundant device and then reset it. After it synchronizes with the active device, it initiates a switchover and becomes the new active device (the former active device resets and becomes the new redundant device).
35.3 Configuring Firewall Allowed Rules

If you want to configure firewall rules (see 'Configuring Firewall Rules' on page 165) that block specific network traffic, you must first configure firewall rules that allow traffic needed in your deployment. Therefore, in addition to allowing basic traffic (such as OAMP, SIP signalling, and media), you must also allow HA maintenance traffic between the Active and Redundant devices:

- UDP ports 669, 670 and 680 (HA synchronization and keep alive)
- TCP ports 2442 and 80 (HA control and data)

Please configure firewall rules 10 through 21, as shown below, where 10.31.4.61 is the IP address of the Maintenance interface ("HA_IF") of the Redundant device and 10.31.4.62 the IP address of the Maintenance interface ("HA_IF") of the Active device.

**Allowed Firewall Rules for HA**

<table>
<thead>
<tr>
<th>Index</th>
<th>Source IP</th>
<th>Source Port</th>
<th>Prefix Length</th>
<th>Start Port</th>
<th>End Port</th>
<th>Protocol</th>
<th>Use Specific Interface</th>
<th>Interface Name</th>
<th>Action Upon Match</th>
<th>Packet Size</th>
<th>Byte Rate</th>
<th>Byte Burst</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10.31.4.61</td>
<td>669</td>
<td>32</td>
<td>669</td>
<td>669</td>
<td>udp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>10.31.4.62</td>
<td>669</td>
<td>32</td>
<td>669</td>
<td>669</td>
<td>udp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>10.31.4.61</td>
<td>0</td>
<td>32</td>
<td>2442</td>
<td>2442</td>
<td>tcp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>10.31.4.62</td>
<td>2442</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>tcp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>10.31.4.61</td>
<td>2442</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>tcp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>14</td>
<td>10.31.4.62</td>
<td>0</td>
<td>32</td>
<td>2442</td>
<td>2442</td>
<td>tcp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>10.31.4.61</td>
<td>80</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>tcp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>10.31.4.62</td>
<td>80</td>
<td>32</td>
<td>0</td>
<td>65535</td>
<td>tcp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>17</td>
<td>10.31.4.61</td>
<td>670</td>
<td>32</td>
<td>670</td>
<td>670</td>
<td>udp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>10.31.4.62</td>
<td>670</td>
<td>32</td>
<td>670</td>
<td>670</td>
<td>udp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>19</td>
<td>10.31.4.61</td>
<td>680</td>
<td>32</td>
<td>680</td>
<td>680</td>
<td>udp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>10.31.4.62</td>
<td>670</td>
<td>32</td>
<td>670</td>
<td>670</td>
<td>udp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>10.31.4.62</td>
<td>680</td>
<td>32</td>
<td>680</td>
<td>680</td>
<td>udp</td>
<td>Enable</td>
<td>HA_IF</td>
<td>Allow</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>0.0.0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>65535</td>
<td>Any</td>
<td>Disable</td>
<td>--</td>
<td>Block</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Note:**
- The index numbers in the table above may change according to your specific allow and block rules.
- The last rule (Index 22) is an example of a blocking traffic rule (blocks all other traffic).
- Configure the firewall on the Active device. This configuration is automatically applied to the Redundant device.
- If you have an external firewall located between the Active and the Redundant HA Maintenance interfaces, you must open (allow) the same port ranges as configured in the table above, on that external firewall.
35.4 Monitoring IP Entity and HA Switchover upon Ping Failure

The device can monitor a specified network entity, using pings. If the device does not receive a ping response from the entity, a switchover to the redundant device occurs. The switchover happens only if a ping was initially successful and then a subsequent ping failed. The feature is referred to as HA Network Reachability. The feature can be used, for example, to check connectivity with a nearby router (first hop) that the device uses to reach other destinations.

The network entity is defined by IP address. The IP interface from where the ping is sent can be selected from one of the device's configured network interfaces in the IP Interfaces table.

**Note:**
- The HA Network Reachability feature is not functional under the following conditions:
  - HA is disabled (i.e., active device is in standalone mode).
  - HA Preempt Priority is used (to prevent endless loops of switchovers).
  - Number of Ethernet Groups in the redundant device that are in "up" state are less than on the active device (to prevent endless loops of switchovers).
- If you have configured the HA Network Reachability feature, but the feature is not operational (see note above), the device sends the SNMP trap event, acHANetworkWatchdogStatusAlarm to notify of the situation.
- If a switchover occurs due to no ping reply, the device sends the SNMP trap alarm, acHASystemFaultAlarm to notify of the switchover due to the HA Network Reachability feature.
- For a detailed description of the HA ping parameters, see "HA Parameters" on page 864.

➢ To configure monitoring of IP entity using pings:

1. Open the HA Settings page (Setup menu > IP Network tab > Core Entities folder > HA Settings).
2. Configure the following:
   - From the 'HA Network Reachability' (HAPingEnabled) drop-down list, select Enable.
   - In the 'Destination Address' field, enter the address of the IP entity that you want to monitor.
   - In the 'Source Interface Name' field, enter the device's IP network interface from where you want to ping the destination entity.
   - In the 'Ping Timeout' field, enter the timeout for which the ping request waits for a response.
In the 'Ping Retries' field, enter the number of ping requests that the device sends after no ping response is received from the destination, before it considers the destination as unavailable.

**Figure 35-8: Configuring HA Network Reachability**

3. Click **Apply**.

If the feature is operational, the status of the connectivity to the pinged destination is displayed in the 'Monitor Destination Status' read-only field:

- **“Enabled”:** Ping is sent as configured.
- **"Disabled by configuration and HA state":** HA and ping are not configured.
- **"Disabled by HA state":** same as above.
- **"Disabled by configuration":** same as above.
- **"Disabled by invalid configuration":** invalid configuration, for example, invalid interface name or destination address (destination address must be different than a local address and from the redundant device's Maintenance address).
- **"Disabled by HA priority in use":** when HA priority is used, ping mechanism is disabled.
- **"Disabled by Eth groups error":** when the number of Ethernet Groups in the redundant device becomes less than in the active device, the ping mechanism is disabled.
- **“Failed to be activated”:** Internal error (failed activating the ping mechanism).
36 HA Maintenance

This section describes HA maintenance procedures.

36.1 Maintenance of Redundant Device

The only interface that is operational on the redundant device is the Maintenance interface. The following protocols can be used for maintenance purposes for this interface:

- **Syslog:** To receive Syslog messages from the redundant device, make sure that you have configured a valid VLAN and routing rule from the Maintenance network to the Syslog server.
- **Telnet:** A Telnet server is always available on the redundant device (even if disabled by configuration).

The active device runs the above maintenance protocols on its' OAMP interface.

36.2 Replacing a Failed Device

If you need to replace a faulty device, the new device must be configured exactly as the second device, as described in "Configuring the HA Devices" on page 605.

36.3 Initiating an HA Switchover

You can initiate a switchover from the Active to Redundant device.

**Note:** When performing an HA switchover, the HA mode becomes temporarily unavailable.

➢ **To perform a switch-over:**

1. Open the High Availability Maintenance page:
   - Toolbar: Click the Actions button, and then from the drop-down menu, choose Switchover.

   ![Figure 36-1: Performing a Device HA Switchover](image-url)

2. Click Switch Over; a confirmation box appears requesting you to confirm.
3. Click OK.
36.4 Resetting the Redundant Unit

You can reset the Redundant device, if necessary.

Note: When resetting the Redundant device, the HA mode becomes temporarily unavailable.

To reset the Redundant device:
1. Open the High Availability Maintenance page:
   - Toolbar: Click the Actions button, and then from the drop-down menu, choose Switchover.

   Figure 36-2: Resetting Redundant Device

   ![Redundant Options]

   Reset The Redundant Board [Reset]

2. Click Reset; a confirmation box appears requesting you to confirm.
3. Click OK.

36.5 Software Upgrade

You can perform the following types of software upgrades on the HA system:

- **Software Upgrade with Device Reset**: Both active and redundant devices burn and reboot with the new software version. This method is quick and simple, but it disrupts traffic (i.e., traffic affecting).

- **Hitless Software Upgrade**: This method maintains service (i.e., not traffic affecting).

For more information, see "Software Upgrade Wizard" on page 683.
36.6 Rescue Options

The device features a System Snapshots mechanism that provides the capability of returning the system to a previous state. The mechanism may be used as a rescue option if a system malfunction occurs.

Note: For Mediant VE SBC, in addition to the functionality described in this chapter, you can use the snapshots functionality provided by the virtual machine hypervisor.

36.6.1 Taking a Snapshot

Taking a System Snapshot captures a complete state of the device, including:
- Installed software
- Current configuration
- Auxiliary files
- License Key

The first ‘factory’ snapshot is automatically taken when initial installation is performed. Additional snapshots (up to 10) may be taken. The device can be returned to a snapshot, as described below.

➢ To take a snapshot in the CLI:
1. Connect to the CLI interface.
2. At the prompt, type the following and then press Enter:
   > enable
3. At the prompt, type the password and then press Enter:
   Password: Admin
4. At the prompt, type the following to save the current configuration (burn) before creating a snapshot:
   # write
5. Type the following commands to take a snapshot:
   # configure troubleshoot
   (config-troubleshoot)# startup-n-recovery
   (startup-n-recovery)# create-system-snapshot <name>

36.6.2 Viewing Available Snapshots

Currently available system snapshots can be viewed by using the `show-system-snapshots` command. The ‘default’ snapshot is indicated by an asterisk.

(startup-n-recovery)# show-system-snapshots
first-install-2010-01-01_03-18-29
pre-production-6.70.037.010-2010-01-08_00-39-58
*production-6.70.037.010-2010-01-08_00-41-30
36.6.3 Changing the Default Snapshot

The ‘default’ snapshot indicates a restore point that is used by Automatic Recovery in the case of software malfunction (see “Automatic Recovery” on page 618) and/or Manual Recovery (see “Manual Recovery” on page 616). The last user-created snapshot is automatically set as ‘default’ though it can be changed using the following command:

```bash
(setup-n-recovery)# set-default-snapshot pre-production-6.70.037.010-2010-01-08_00-40-27
```

36.6.4 Deleting a Snapshot

To delete a snapshot, use the following command:

```bash
(setup-n-recovery)# delete-system-snapshot pre-production-6.70.037.010-2010-01-08-00-39-58
```

36.6.5 Manual Recovery

You can perform a Manual recovery. When the device reboots, a GRUB menu is displayed that lets you select one of the following rescue options:

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

36.6.5.1 Returning to the Default Snapshot

➢ To return to the default snapshot:

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

The system returns to the default snapshot, restoring the software version and the full configuration. The process can take up to 10 minutes to complete.

Figure 36-3: System Returning to Snapshot State
36.6.5.2 Fixing the Current Installation

➢ To fix the current installation:
  ▪ In the GRUB menu, select Fix current installation, and then press Enter; the system is repaired while the currently installed software version and its configuration are preserved. The process can take up to 10 minutes to complete.

36.6.5.3 Returning to an Arbitrary Snapshot

➢ To return to an arbitrary (non-default) system snapshot:
  ▪ GRUB Menu:
    1. Select Browse available system snapshots, and then press Enter; you’re prompted to select a snapshot.
    2. Select a snapshot, and then press Enter; the system returns to the selected snapshot, restores the software version and the full configuration. The process may take up to 10 minutes to complete.
  ▪ CLI:

```
# configure troubleshoot
(config-troubleshoot)# startup-n-recovery
(startup-n-recovery)# load-from-snapshot <Name of Snapshot>
```

Note: Loading a snapshot through CLI is not supported in HA mode.

36.6.5.4 Returning to a Factory Snapshot

➢ To return to a factory snapshot (after install from CD):
  ▪ In the GRUB menu, select Return to factory snapshot (after install from CD), and then press Enter; the system returns to the first snapshot automatically taken when initial installation from CD was performed. The process can take up to 10 minutes to complete.
36.6 Automatic Recovery

The device activates Automatic Recovery when it encounters a severe software malfunction that prevents it from successfully booting for three subsequent attempts. Automatic Recovery returns the system to the ‘default’ snapshot and may take up to 10 minutes to complete.

36.7 Disconnecting and Reconnecting HA

You can disconnect the two devices in the HA system and return them to standalone devices.

➢ To disconnect HA:

On the active device, enter the following CLI command:

```
# debug ha disconnect-system < new OAMP address of redundant device >
```

**Note:**

- The new OAMP address of the redundant device must be different to the active device.
- The HA Maintenance network interface (in the IP Interfaces table) on the redundant device is unaffected by the command.
- **All Media + Control** network interfaces (in the IP Interfaces table) are deleted on the redundant device.
- The 'HA Remote Address' (HARemoteAddr) field value, which specifies the HA Maintenance address of the active device, is deleted on the redundant device.
- The command causes the redundant device to reset.

You can later restore the HA system, by following the below procedure.

➢ To restore the HA system:

1. On the previously **redundant** device, configure the HA Maintenance interface address of the active device in the 'HA Remote Address' field, or through CLI:

```
(config-network)# high-availability
(ha)# remote-address < HA Maintenance address >
```

2. Reset the redundant device.

**Note:** The procedure assumes that no network changes were made to both devices' HA Maintenance interface or Ethernet Devices (VLAN); otherwise, the devices may not be able to communicate with each other.
Part VIII

Media Transcoding Cluster
37 Media Transcoding Cluster Overview

The device can use an external source of DSP resources for media-related features requiring DSPs such as vocodec transcoding, fax transcoding, and DTMF detection. The external source of DSP resources is provided by a cluster of AudioCodes' Media Transcoders, which serve as a dedicated farm of DSP resources for the SBC device. In such an architecture, the SBC device is also referred to as a Cluster Manager, which manages the cluster. Media Transcoding Cluster is typically implemented when the SBC device demands higher DSP resources than it can normally provide on its own. In such a deployment, the device offloads its DSP-required media processing to the external Media Transcoders. The main benefit of the feature is scalability, where you can initially deploy a few Media Transcoders to meet your transcoding demands and then later add Media Transcoders as capacity of transcoding requirements grow. The following table describes the Media Transcoders:

<table>
<thead>
<tr>
<th>Media Transcoder Type</th>
<th>Term Used</th>
<th>Maximum Deployed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware-based appliance</td>
<td>&quot;MT&quot;</td>
<td>8</td>
</tr>
<tr>
<td>Virtualized machine</td>
<td>&quot;vMT&quot;</td>
<td>5</td>
</tr>
</tbody>
</table>

Note: All Media Transcoders in the Media Transcoding Cluster deployment must be of one type (all MTs or all vMTs; not both).

Media Transcoders provide only DSP functionality (i.e., no SIP routing functionalities) and some system functionalities such as debugging through Syslog. Media Transcoders are "hidden" from the endpoints that are serviced by the SBC device. Once you have initially configured (e.g., OAMP and Cluster interfaces) the Media Transcoders, all management of Media Transcoders is through the SBC device.

The SBC device can operate with Media Transcoding Cluster in a High-Availability (HA) or Standalone mode. If a switchover occurs in the HA system, communication between the redundant device and the Media Transcoders is established and the redundant device (now active) starts allocating new transcoding sessions to the Media Transcoders. During the switchover, all current transcoding sessions handled by the Media Transcoders are maintained. HA configuration and setup for the SBC device is the same as when operating without the Media Transcoding Cluster feature.

The Media Transcoding Cluster can also operate in clustered high-availability mode with load-sharing capabilities. If a failure occurs in a Media Transcoder, its transcoding sessions are automatically distributed to all other available Media Transcoders.
Note:

- The maximum transcoding capacity is according to the License Key installed on the SBC device. However, to achieve the licensed capacity you must have the required number of Media Transcoders deployed in the cluster.
- When the feature is enabled, the SBC device does not use any of its internal DSP resources.
- Currently, the Media Transcoding Cluster feature is supported only by Mediant VE SBC based on OpenStack KVM.

37.1 Network Topology and Cluster Interfaces

In addition to the SBC device's existing local IP network connections such as SIP Control and Media interfaces for routing between SIP entities, and the OAMP interface for management, you need to configure additional network interfaces for connectivity with the Media Transcoders. The IP network interfaces used for communication with the Media Transcoders are configured with the special Application Type, "Cluster Media + Control" (hereafter, referred to as the Cluster interface). The number of ports configured for the Cluster interface depends on the employed network connection type between the SBC device and the Media Transcoders, which can be one of the following:

- **Indirect Connection (through switch):** The Media Transcoder is connected to the SBC device through an Ethernet switch. The Ethernet port on the SBC device for this connection is associated with a Cluster network interface that can be used for multiple Media Transcoders. The Ethernet port can also be associated with OAMP and/or SIP Control network interface application types (not Media).
- **MT:** The Cluster interface on the SBC device is composed of a single Ethernet port while the Cluster interface on the Media Transcoder can be composed of a single port or two ports (for port redundancy):
• **vMT**: The Cluster interfaces on the SBC device and Media Transcoder are composed of a single Ethernet port:

**Figure 37-1: Indirect Connection between vMT Media Transcoders and SBC Device**

The configured operating mode of the Ethernet ports depends on the number of ports used for the Cluster interface:

**Number of Ports in Cluster Interface and Port Mode**

<table>
<thead>
<tr>
<th>Number of Ports in Cluster Interface</th>
<th>Port Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Single</td>
</tr>
<tr>
<td>2</td>
<td>Redun 2RX 1TX</td>
</tr>
</tbody>
</table>
37.2 Prerequisites

The prerequisites for the Media Transcoding Cluster feature include the following:

- **Device**: The device must be installed with a License Key that:
  - Enables functionality as a Cluster Manager ("CM")
  - Defines the maximum number of transcoding sessions

- **Media Transcoders**: Number of Media Transcoders to meet your transcoding requirements.
38 Configuring Media Transcoding Cluster

This chapter describes how to configure the Media Transcoding Cluster feature.

38.1 Initial Cluster Configuration

This section describes Media Transcoding Cluster configuration for initial installation.

38.1.1 Enabling the Cluster Manager

For the SBC device to serve as a Cluster Manager, you must enable Cluster Manager functionality. If already enabled (default), skip this section.

➢ To enable the SBC device for Cluster Manager functionality:

1. In the SBC device's Web interface, open the Cluster Manager Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > Cluster Manager Settings).
2. From the 'Cluster Manager Functionality' drop-down list, select Enable:

   Figure 38-1: Enabling the Media Transcoding Feature

3. Click Apply, and then reset the SBC device with a save-to-flash for your settings to take effect.

38.1.2 Adding Media Transcoders

Add one or more Media Transcoders, as described in Adding a Media Transcoder on page 626.

38.1.3 Configuring Cluster Redundancy Mode

The Media Transcoding Cluster feature provides load-sharing and cluster redundancy between multiple Media Transcoders. Load sharing attempts to distribute transcoding sessions equally between Media Transcoders. Cluster redundancy offers the following schemes:

- **HA Mode:** (Default) The SBC device guarantees that in case of a failure in a Media Transcoder, sufficient DSP resources are available on the other Media Transcoders to take over the active transcoding sessions of the failed Media Transcoder. Therefore, a situation never occurs where all the Media Transcoders are utilizing all their DSP resources at any given time. Once a failure occurs in a Media Transcoder, the SBC device temporarily operates in Best Effort mode (see below) until the faulty Media Transcoder returns to service, after which the SBC device operates again in HA mode. If a failure still exists in the Media Transcoder and you want the SBC device to continue operating in HA mode, you need to remove the failed Media Transcoder from the table of configured Media Transcoders.

- **Best Effort:** The SBC device allocates transcoding sessions to the Media Transcoder without guaranteeing availability of DSP resources on other Media Transcoders should the Media Transcoder fail. Therefore, a situation may occur where all Media Transcoders are utilizing all their DSP resources at any given time. However, if a failure occurs in a Media Transcoder and there are some DSP resources available on other Media Transcoders, the SBC device transfers as many active sessions as possible to the other Media Transcoders. Sessions for which DSP resources are
unavailable are dropped. Therefore, the Best Effort mode enables greater session capacity, but at the expense of cluster HA.

➢ To configure cluster redundancy:
1. In the SBC device's Web interface, open the Cluster Manager Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > Cluster Manager Settings).
2. From the 'Cluster Redundancy Mode' drop-down list, select the redundancy method Best Effort or HA Mode:

   Figure 38-2: Configuring Cluster Redundancy Mode

   

3. Click Apply.

38.1.4 Configuring Maximum Bandwidth for Cluster Interfaces

You can configure a maximum bandwidth allowed on each Cluster interface. If the bandwidth of a Cluster interface approaches this limit, the SBC device stops allocating new transcoding sessions to the Media Transcoders associated with the specific Cluster interface. Allocation is resumed only once bandwidth drops below the limit. Maximum bandwidth configuration is typically needed for setups in which multiple Media Transcoders are associated with the same Cluster interface. It ensures that the total traffic on the Cluster interface does not exceed available bandwidth in your network.

➢ To configure maximum bandwidth for Cluster interfaces:
1. In the SBC device's Web interface, open the Cluster Manager Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > Cluster Manager Settings).
2. In the 'Bandwidth Limit per Cluster Interface' field, enter the maximum bandwidth (in Mbps):

   Figure 38-3: Configuring Maximum Bandwidth for Cluster Interfaces

3. Click Apply.

38.2 Adding a Media Transcoder

This section describes how to add a Media Transcoder to the Media Transcoding Cluster. It is based on an example setup consisting of three Media Transcoders, each associated with a different Cluster interface.

38.2.1 Step 1: Configure Device with Cluster Network Interface

The first step is to configure on the SBC device a Cluster interface for communicating with the Media Transcoder.

➢ To configure a Cluster interface:
1. Access the SBC device's Web interface.
2. Open the Physical Ports table (see Configuring Physical Ethernet Ports on page 130), and then configure the Ethernet port for the Cluster interface (e.g., port "GE_2"):

![Figure 38-4: Configuring Ethernet Port for Cluster Interface (Example)](image)

3. Open the Ethernet Groups table (see Configuring Ethernet Port Groups on page 132), and then assign the Ethernet port to an Ethernet Group (e.g., Group 2) and configure its operating mode (e.g., Single):

![Figure 38-5: Configuring Ethernet Group for Cluster Interface (Example)](image)

4. Open the Ethernet Devices table (see Configuring Underlying Ethernet Devices on page 134), and then configure an Ethernet Device by assigning it the Ethernet Group that you configured in the previous step and a VLAN ID (e.g., "vlan 2"):

![Figure 38-6: Configuring Ethernet Device for Cluster Interface (Example)](image)

5. Open the IP Interfaces table (see Configuring IP Network Interfaces on page 137), and then configure a Cluster interface with 'Application Type' Cluster Media + Control, IP address (e.g., 10.99.91.130), and assigned the Ethernet Device (e.g., "vlan 1") that you configured in the previous step:

![Figure 38-7: Configuring IP Network Interface for Cluster Interface (Example)](image)

Note: Configure the Cluster interface with an IPv4 address only.

6. Reset the SBC device with a save-to-flash for your settings to take effect.
38.2.2 Step 2: Initial Configuration of Media Transcoder

Before you can add the Media Transcoder to the Cluster Manager, you need to configure the Media Transcoder's OAMP interface address and port(s) to suit your network IP addressing scheme. Configuration is done through its Web-based management interface, connected using the default OAMP interface address (see table below).

### Default OAMP Interface of Media Transcoder

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>&quot;O+M+C&quot;</td>
</tr>
<tr>
<td>Application Type</td>
<td>OAMP + Media + Control</td>
</tr>
<tr>
<td>IP Address</td>
<td>▪ MT: 192.168.0.2</td>
</tr>
<tr>
<td></td>
<td>Note: The IP address is assigned to the device's Ethernet ports GE 1 and GE 2 (two top-left ports). By default, these ports belong to Ethernet Group “GROUP 1”. ▪ vMT: 192.168.01</td>
</tr>
<tr>
<td>Prefix Length</td>
<td>24 (i.e., 255.255.255.0)</td>
</tr>
<tr>
<td>Default Gateway</td>
<td>192.168.0.1</td>
</tr>
<tr>
<td>Ethernet Device</td>
<td>vlan 1 (untagged)</td>
</tr>
</tbody>
</table>

Note:
- Media Transcoders send Syslog and debug recording messages from their OAMP interfaces to the same servers as configured for the SBC device.
- All OAMP IP addresses in the deployment (SBC device and Media Transcoders) must be unique.
- Once you have configured the Media Transcoder, you can later change its OAMP address as described in Changing OAMP Address of Media Transcoders on page 635.

38.2.2.1 Changing OAMP Address for MT

The following procedure describes how to initially access an "MT" Media Transcoder's Web interface and change its OAMP address.

➢ To initially configure the MT's OAMP address:

1. Connect the Media Transcoder's Ethernet port GE 1 or GE 2, located on the front panel (top-left ports), directly to your computer's network interface port, using a straight-through Ethernet cable:

2. Change the IP address and subnet mask of your computer to correspond with the Media Transcoder's default OAMP IP address and subnet mask.

3. Access the Web interface using the default IP address and then login using the default username ("Admin") and password ("Admin").

4. Configure the Ethernet ports that you want to use for the OAMP interface:
   a. In the Physical Ports table, configure the port speed and duplex mode (see Configuring Physical Ethernet Ports on page 130), as shown in the example below:
38. Configuring Media Transcoding Cluster

b. In the Ethernet Groups table, assign up to two Ethernet ports to an Ethernet Group (see Configuring Ethernet Port Groups on page 132), as shown in the example below:

c. In the Ethernet Devices table, configure an Ethernet Device by assigning it the Ethernet Group and configuring a VLAN ID (see Configuring Underlying Ethernet Devices on page 134), as shown in the example below:

5. Open the IP Interfaces table (see Configuring IP Network Interfaces on page 137), and then modify the OAMP interface address ("O+M+C") to suit your networking environment:

6. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

7. Disconnect the cable between the Media Transcoder and your computer, and then cable the Media Transcoder to your network using the Ethernet port configured for OAMP.

8. Reconnect to the Media Transcoder's Web interface using the new OAMP interface address.

38.2.2.2 Changing OAMP Address of vMT

The following procedure describes how to initially access an "vMT" Media Transcoder's Web interface and change its OAMP address.

➢ To initially configure the vMT's OAMP address:

1. Connect to the virtual machine’s console (e.g., in vSphere, switch to Console tab).

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 enable and press ENTER:

   > enable

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

   Password: Admin

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

   # configure network

   (config-network)# interface network-if 0

   (network-if-0)#

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

   (network-if-0)# ip-address 10.3.91.130

   (network-if-0)# prefix-length 16

   (network-if-0)# gateway 10.3.0.1

8. At the prompt, type exit to complete the network-if configuration:

   (network-if-0)# activate

   (network-if-0)# exit

9. If the vMT is connected to the IP network that uses VLAN ID, type the following commands to configure it (otherwise skip to step 10):
10. Connect to the Media Transcoder’s Web interface using the new OAMP interface address.

38.2.3 Step 3: Configure Media Transcoder with Cluster Interface

Once you have configured the Media Transcoder's OAMP interface (if required), you need to configure it with a Cluster interface.

Note:
- The Cluster interface must be in the same subnet as its corresponding Cluster interface on the SBC device.

➢ To configure the Media Transcoder with a Cluster interface:

1. Access the Media Transcoder's Web interface.
2. In the Physical Ports table (see Configuring Physical Ethernet Ports on page 130), configure the port (e.g., "GE 3") with a speed and duplex mode:
3. In the Ethernet Groups table (see Configuring Ethernet Port Groups on page 132), assign the Ethernet port to an Ethernet Group (e.g., "Group 2"):
4. In the Ethernet Devices table (see Configuring Underlying Ethernet Devices on page 134), configure an Ethernet Device by assigning it the Ethernet Group and configuring a VLAN ID (e.g., "vlan 2"):
5. Open the IP Interfaces table (see Configuring IP Network Interfaces on page 137), and then configure a Cluster interface with ‘Application Type’ Cluster Media + Control and assigned the Ethernet Device (e.g., “vlan 2”) that you configured in the previous step:

Note: Configure the Cluster interface with an IPv4 address only.

6. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

38.2.4 Step 4: Cable Device to Media Transcoder

Cabling between the device and Media Transcoder depends on the chosen connectivity method:

■ Indirect Connectivity (through Ethernet switch):

1. Connect the SBC device's Ethernet port that you assigned to the Cluster interface for the Media Transcoder to the network.
2. Connect the Media Transcoder's Ethernet port that you assigned to the Cluster interface, to the network.
3. Verify that the Ethernet link status for the port is "Up" (Monitor menu > Monitor tab > Network Status folder > Ethernet Port Information).
38. Configuring Media Transcoding Cluster

38.2.5  Step 5: Configure Media Transcoder with Device's Cluster Address

The Media Transcoder communicates with the SBC device using the IP address of the Cluster interface configured for the Media Transcoder on the SBC device (see Step 1: Configure Device with Cluster Network Interface on page 626). You need to configure this destination address on the Media Transcoder.

➢ To configure Media Transcoder with Cluster Manager address:

1. Access the Media Transcoder's Web interface.
2. Open the Cluster Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > MTC Settings).
3. In the 'Cluster Manager IP Address' field, enter the IP address (e.g., 10.99.91.130) of the Cluster interface that you configured on the SBC device for the Media Transcoder:

Figure 38-8: Configuring Media Transcoder with Cluster Manager Address

4. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

38.2.6  Step 6: Adding Media Transcoder to Cluster Manager

The device's Media Transcoders table lets you configure the Media Transcoders. A Media Transcoder is defined on the SBC device by the IP address of its OAMP interface. The SBC device identifies the Media Transcoder by this OAMP address. For the maximum number of supported Media Transcoders, see Media Transcoding Cluster Overview on page 621.

Once you have configured a Media Transcoder, the table also displays status and other information relating to the selected Media Transcoder. For more information, see Viewing Status per Media Transcoder on Cluster Manager on page 641.

Note: If you have configured firewall rules to allow and/or block various network traffic, make sure that your firewall also allows traffic received from the Media Transcoders. For configuring firewall rules, see Configuring Firewall Rules 165.

The following procedure describes how to configure Media Transcoders through the Web interface. You can also configure it through ini file (MtcEntities) or CLI (configure network > mtc entity).

➢ To configure a Media Transcoder on the device:

1. Access the SBC device's Web interface.
2. Open the Media Transcoders table (Setup menu > IP Network tab > Transcoding Cluster folder > Media Transcoders).
3. Click **New**; the following dialog box appears:

**Figure 38-9: Media Transcoders Table - Add Dialog Box**

![Media Transcoders Table - Add Dialog Box](image)

4. Configure a Media Transcoder according to the parameters described in the table below.

5. Click **Apply**.

The following figure displays an example of the Media Transcoder defined on the device in the table by its OAMP address (configured in Step 2: Initial Configuration of Media Transcoder on page 628):

**Figure 38-10: Configuring Device with OAMP of Media Transcoder**

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>OAMP ADDRESS</th>
<th>ADMIN STATE</th>
<th>VERSION</th>
<th>STATUS</th>
<th>UTILIZATION</th>
<th>ALARM SEVERITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>130</td>
<td>10.391:130</td>
<td>Unlocked</td>
<td>7.20A.000.022</td>
<td>Connected</td>
<td>0%</td>
<td>None</td>
</tr>
</tbody>
</table>

**Media Transcoders Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[MtcEntities_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Name</td>
<td>Defines a name for the Media Transcoder.</td>
</tr>
<tr>
<td>[MtcEntities_MtceName]</td>
<td>The valid value is a string of up to 21 characters. By default, the device automatically assigns it the following name: &quot;MtcEntities_&lt;row index&gt;&quot; (e.g., &quot;MtcEntities_1&quot; if added to Index 1).</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each row must be configured with a unique name.</td>
</tr>
<tr>
<td>OAMP IP Address</td>
<td>Defines the IP address of the Media Transcoder's OAMP interface.</td>
</tr>
<tr>
<td>[MtcEntities_MtceOamIpAddr]</td>
<td>The valid value is an IP address in dotted-decimal notation. The default is 0.0.0.0.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Each Media Transcoder must have a unique OAMP IP address.</td>
</tr>
</tbody>
</table>
39 Media Transcoding Cluster Maintenance

This section describes maintenance operations that you can do on the Media Transcoding Cluster.

39.1 Configuring Graceful Timeout

Many maintenance actions on Media Transcoders (such as locking, resetting or upgrading) let you employ a "grace" period (graceful timeout) before the action is started by the Cluster Manager. The graceful timeout lets the Media Transcoder continue processing currently active transcoding sessions until the timeout expires. During the timeout, the Cluster Manager does not allocate new transcoding sessions to the Media Transcoder. If all transcoding sessions end before the timeout expires, the Cluster Manager does not wait for the timeout to expire, but performs the maintenance action immediately. If active transcoding sessions still exist when the timeout expires, the Cluster Manager transfers all the transcoding sessions on the Media Transcoder to other Media Transcoders and then performs the maintenance action on the Media Transcoder. Transcoding sessions that cannot be transferred due to unavailable DSP resources on other Media Transcoders are immediately terminated. These terminated sessions are indicated in Syslog messages.

➢ To configure the graceful timeout:

1. Open the Cluster Manager Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > Cluster Manager Settings).
2. In the 'Cluster Graceful Timeout' field, enter the graceful timeout:

   Figure 39-1: Configuring Graceful Timeout

3. Click Apply.

39.2 Actions on Media Transcoders

In addition to adding a Media Transcoder (as described in Adding a Media Transcoder on page 626), you can perform the following on each Media Transcoder in the Media Transcoders table:

- Lock and Unlock (see Locking and Unlocking Media Transcoders on page 633)
- Reset (Resetting Media Transcoders on page 634)
- Delete (Deleting Media Transcoders on page 634)
- Modify name (Changing the Name of Media Transcoders on page 635)

**Note:** You cannot do any administrative actions on locked Media Transcoders (reset, add, edit, delete, or software upgrade).

39.2.1 Locking and Unlocking Media Transcoders

When you lock a Media Transcoder, the Cluster Manager stops allocating new transcoding sessions to it. You can lock the Media Transcoder "gracefully", whereby the Cluster Manager allows the Media Transcoder to continue processing currently active transcoding
sessions for a user-defined time (graceful timeout). For more information on graceful timeout, see Configuring Graceful Timeout on page 633. Unlocking a Media Transcoder takes the Media Transcoder out of locked state and allows the Cluster Manager to start allocating new transcoding sessions to the Media Transcoder.

➢ To lock or unlock a Media Transcoder:

1. Open the Media Transcoders table (see Step 6: Adding Media Transcoder to Cluster Manager on page 631).
2. Select the required Media Transcoder, and then from the Action button on the table’s toolbar, choose one of the following:
   - Lock: The Media Transcoder locks and the 'Admin State' read-only field displays "Locked".
   - UnLock: The Media Transcoder unlocks and the 'Admin State' read-only field displays "Unlocked".

39.2.2 Resetting Media Transcoders

You can reset a Media Transcoder from the Cluster Manager. You can reset the Media Transcoder "gracefully", whereby the Cluster Manager allows the Media Transcoder to continue processing currently active transcoding sessions for a user-defined time (graceful timeout). For more information on graceful timeout, see Configuring Graceful Timeout on page 633.

➢ To reset a Media Transcoder:

1. Open the Media Transcoders table (see Step 6: Adding Media Transcoder to Cluster Manager on page 631).
2. Select the required Media Transcoder row, and then from the Action button on the table's toolbar, choose Reset.

39.2.3 Deleting Media Transcoders

The following procedure describes how to delete a Media Transcoder from the Media Transcoding Cluster. Once deleted, the Cluster Manager stops allocating transcoding sessions to it.

Note:

- You can delete only locked Media Transcoders.
- You cannot delete Media Transcoders that are undergoing a software upgrade or reset.

➢ To delete a Media Transcoder:

1. Open the Media Transcoders table (see Step 6: Adding Media Transcoder to Cluster Manager on page 631).
2. Select the required Media Transcoder row, and then from the Action button on the table's toolbar, choose Lock to lock the Media Transcoder.
3. When the status of the Media Transcoder is "Locked", select the Media Transcoder again, and then from the table's toolbar, click ; a confirmation message box appears.
4. Click Yes, Delete to confirm deletion.
39.2.4 Changing the Name of Media Transcoders

The following procedure describes how to change the name of a Media Transcoder.

➢ To change the name of a Media Transcoder:

1. Access the device's Web interface.
2. Open the Media Transcoders table (see Step 6: Adding Media Transcoder to Cluster Manager on page 631).
3. Select the required Media Transcoder row, and then click Edit.
4. In the 'Name' field, change the name of the Media Transcoder, and then click Apply.

39.3 Changing OAMP Address of Media Transcoders

The following procedure describes how to change the OAMP interface's IP address of a Media Transcoder.

➢ To modify the OAMP IP address of a Media Transcoder:

1. In the device's Web interface, lock the Media Transcoder (see Locking and Unlocking Media Transcoders on page 633); the Cluster Manager stops allocating new transcoding sessions to the Media Transcoder and transfers the existing ones to other Media Transcoders.
2. In the device's Web interface, delete the Media Transcoder from the Media Transcoders table (see Deleting Media Transcoders on page 634).
3. In the Media Transcoder's Web interface, modify the OAMP address in the IP Interfaces table, and then reset the Media Transcoder with a save-to-flash for your settings to take effect.
4. In the device's Web interface, add the Media Transcoder with the new OAMP address (the name can remain the same) to the Media Transcoders table (see Step 6: Adding Media Transcoder to Cluster Manager on page 631).

39.4 Upgrading the Cluster Software Version

You can upgrade the software version (*.cmp file) of Media Transcoders through the Cluster Manager. You can use any of the following upgrade modes:

■ Manual: Load the file to the Cluster Manager and then manually reset the Media Transcoder. When the Media Transcoder re-connects to the device after resetting, it automatically loads the new file from the Cluster Manager. The benefit of this mode is that you can choose the Media Transcoder to upgrade. The method also maintains the Media Transcoder's current transcoding sessions, by transferring them to other available Media Transcoders (assuming they have sufficient DSP resources).

■ Hitless: (Default) The Cluster Manager sequentially upgrades all Media Transcoders to maintain active transcoding sessions. Only Media Transcoders whose current software version is different from the one loaded from the Cluster Manager are upgraded. Once a Media Transcoder loads the software version file, the Cluster Manager transfers all the Media Transcoder's current transcoding sessions to other available Media Transcoders. If any sessions cannot be transferred due to unavailable DSP resources on other Media Transcoders, the Cluster Manager waits a user-defined "graceful" period. Upon expiry of the period, the Cluster Manager tries again to transfer these transcoding sessions. Transcoding sessions that still cannot be transferred after this period are terminated. The Media Transcoder then resets to
apply the new software version. Once it has reset and connectivity between the Cluster Manager and Media Transcoder is restored, the Cluster Manager allocates transcoding sessions to the Media Transcoder as usual and then does the same hitless upgrade to the next Media Transcoder, and so on. If the upgrade process of a Media Transcoder fails (the Cluster Manager attempts three times), you can reset the Media Transcoder and restore the Media Transcoder to its previous software version.

**Immediate:** This mode is similar to the Hitless mode, except that it is service-affecting, blocking new transcoding sessions and terminating all current transcoding sessions. This makes the upgrade process much faster than the Hitless method. Only once connectivity between the Cluster Manager and Media Transcoders are restored after reset, does the Cluster Manager allocate new transcoding sessions to the Media Transcoders.

### Note:

- During Immediate and Hitless upgrade, you cannot perform any administrative actions on Media Transcoders (e.g., lock) until all Media Transcoders have been upgraded. For Manual upgrade, this applies only to the Media Transcoder that you are upgrading.
- Hitless upgrade can be performed only when the cluster is in "Full HA" status.
- For Hitless upgrade, you can stop the upgrade process at any time. If you stop the process, you may end up with a situation where some Media Transcoders may have already successfully completed their upgrade.
- Whenever the Cluster Manager establishes a new connection with a Media Transcoder (e.g., after a reset), the Media Transcoder automatically upgrades its software version if the Cluster Manager has a later version for the Media Transcoder.
- The .cmp file name for the device (Cluster Manager) and Media Transcoders are different. Upgrading the Media Transcoding Cluster does not automatically upgrade the device.
- If you need to upgrade the software version of both the device and the Media Transcoders, do the following:
  1) Upgrade the device using the Software Upgrade Wizard, as described in Software Upgrade.
  2) When the device’s upgrade process completes, upgrade the Media Transcoders using the Hitless mode as described below.

➢ **To upgrade Media Transcoders:**

1. Open the Cluster Manager Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > Cluster Manager Settings).
2. Under the Image Repository group, do the following:
   a. From the ‘Media Transcoder Type’ drop-down list, select the type of Media Transcoder (i.e., MT or vMT).
b. Click the **Load File** button, and then select the software version file (.cmp) on your computer.

**Figure 39-2: Selecting Software Version File for Media Transcoder**

![IMAGE REPOSITORY]

Media Transcoder Type: **MT**

Current MT Image Filename: **M400G_SIP_F7.20A.050.450.cmp**

[Load File]

The currently installed software version is displayed in the 'Current Image MT Filename' read-only field.

**Note:** When you add a new Media Transcoder, it automatically installs this loaded software version file to the Media Transcoder (if the file is a later version than the one currently installed on the Media Transcoder).

3. If you want to upgrade a specific Media Transcoder, reset the Media Transcoder, as described in Resetting Media Transcoders on page 634; otherwise, skip this step.

4. For Immediate or Hitless upgrade modes, select the relevant mode:
   - **Immediate Mode:** Media Transcoders are upgraded at the same time.
   - **Hitless Mode:** Media Transcoders are upgraded sequentially.

   For more information on the upgrade modes, see the beginning of this section.

   **Figure 39-3: Selecting Software Upgrade Mode**

   ![SOFTWARE UPGRADE]

   [Start Upgrade]

5. If you selected **Hitless Mode** in the previous step, configure the graceful timeout (see Configuring Graceful Timeout on page 633); otherwise, skip this step.

6. Click the **Start Upgrade** button; the Cluster Manager begins upgrading the Media Transcoders. You can view the status of the software upgrade process, as described in Viewing Cluster Activity Logged Messages on page 639.

To stop the Hitless upgrade, click the **Stop Hitless Upgrade** button.
39.5 Loading Auxiliary Files

When you load an Auxiliary file (e.g., PRT and CPT) to the Cluster Manager, it automatically loads the file to the Media Transcoders. Use the Auxiliary Files page to load Auxiliary files, as described in Loading Auxiliary Files.

Note: Loading a CPT file requires a device reset for the file's settings to take effect.
40 Media Transcoding Cluster Status and Monitoring

This section describes how to view the status of the Media Transcoding Cluster.

40.1 Viewing Cluster Activity Logged Messages

The Cluster Manager logs all actions performed on the Media Transcoders. You can view the logged messages in the Web interface of the Cluster Manager, as described in the following procedure.

➢ To view logged messages:
  ▶ Open the Transcoding Cluster Log page (Setup menu > IP Network tab > Transcoding Cluster folder > Transcoding Cluster Log).

The logged messages are displayed in the following syntax: <time action was logged in hh:mm:ss format> <Media Transcoder ID and name ("MT#") or Cluster Manager ("CM")> <message description>

The table below lists the messages generated per action type. The messages of each action type are listed in chronological order.

Logged Messages

<table>
<thead>
<tr>
<th>Action</th>
<th>Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adding Media Transcoder</td>
<td>1  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Added by Administrator (OAMP IP: 10.4.219.10)&quot;</td>
</tr>
<tr>
<td></td>
<td>2  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Waiting for connection establishment&quot;</td>
</tr>
<tr>
<td></td>
<td>3  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Connection was established successfully, version - &lt;version&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td>4  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: SW upgrade is required, loading new version - &lt;version&gt;&quot;</td>
</tr>
<tr>
<td></td>
<td>5  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Loading DSP tones file&quot;</td>
</tr>
<tr>
<td></td>
<td>6  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Reset performed by the Cluster Manager&quot;</td>
</tr>
<tr>
<td></td>
<td>7  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Waiting for connection establishment&quot;</td>
</tr>
<tr>
<td></td>
<td>8  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Connection was established successfully, version - &lt;version&gt;&quot;</td>
</tr>
<tr>
<td>Locking Media Transcoder</td>
<td>1  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Locked by the Administrator&quot;</td>
</tr>
<tr>
<td></td>
<td>2  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Blocked from new DSP resource allocations&quot;</td>
</tr>
<tr>
<td></td>
<td>3  &quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: All DSP resources were released&quot;</td>
</tr>
<tr>
<td>Action</td>
<td>Messages</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
</tr>
<tr>
<td>Unlocking Media Transcoder</td>
<td>&quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Unlocked by the Administrator&quot;</td>
</tr>
<tr>
<td>Removing Media Transcoder (after being Locked)</td>
<td>&quot;&lt;Time&gt; MT#&lt;ID&gt; &lt;Name&gt;: Removed from the Media Transcoders table by the Administrator&quot;</td>
</tr>
</tbody>
</table>
| Resetting Media Transcoder | 1 "<Time> MT#<ID> <Name>: Reset request by the Administrator"  
2 "<Time> MT#<ID> <Name>: Blocked from new DSP resource allocations"  
3 "<Time> MT#<ID> <Name>: All DSP resources were released"  
4 "<Time> MT#<ID> <Name>: Performing reset"  
5 "<Time> MT#<ID> <Name>: Waiting for connection establishment"  
6 "<Time> MT#<ID> <Name>: Connection was established successfully, version: <version>" |
| Media Transcoder Hardware Failure | "<time> MT#<ID> HW failure – reaching critical temperature threshold"  
Note: Applicable only to MT-type Media Transcoders. |
| Media Transcoder Network Failure | 1 "<Time> MT#<ID> <Name>: Network failure – device is disconnected"  
2 "<Time> MT#<ID> <Name>: Waiting for connection establishment" |
| Media Transcoder Software Upgrade | 1 "<time> CM: New Cluster image file loaded, version: <version>"  
2 "<time> CM: SW upgrade process begin"  
3 "<time> CM: Initiating SW upgrade on MT#<ID> (current version: <version>)"  
 a. "<Time> MT#<ID> <Name>: Loading new version - <version>"  
b. "<Time> MT#<ID> <Name>: Loading DSP tones file"  
c. "<Time> MT#<ID> <Name>: Blocked from new DSP resource allocations"  
d. "<Time> MT#<ID> <Name>: All DSP resources were released"  
e. "<Time> MT#<ID> <Name>: Reset performed by the Cluster Manager"  
f. "<Time> MT#<ID> <Name>: Waiting for connection establishment"  
g. "<Time> MT#<ID> <Name>: Connection was established successfully, version - <version>"  
4 "<time> CM: SW upgrade on MT#<ID> failed (current version: <version>)"  
5 "<time> CM: Retry (1) SW upgrade on MT#<ID> (current version: <version>)" and then same messages as in Item 3 above  
6 "<time> CM: SW upgrade Stopped by the Administrator"  
- or -  
"<time> CM: SW upgrade Completed" |

40.2 Viewing Overall Status of Media Transcoding Cluster

You can view the status of the Media Transcoding Cluster on the Cluster Manager, as described in the following procedure.

➢ To view the status of the Media Transcoding Cluster:

1. Open the Cluster Manager Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > Cluster Manager Settings).
2. Scroll down to the Cluster Status group:

**Figure 40-1: Viewing Status of Media Transcoding Cluster**

<table>
<thead>
<tr>
<th>CLUSTER STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Utilization: 0%</td>
</tr>
<tr>
<td>Media Transcoders Elements Availability: 3/6 Available</td>
</tr>
<tr>
<td>HA Usage: 0%</td>
</tr>
<tr>
<td>HA Status: Full HA</td>
</tr>
</tbody>
</table>

The following information is displayed:

**Description of Cluster Status Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current Utilization</td>
<td>Displays the percentage of currently used DSP resources in the Media Transcoding Cluster (i.e., of all Media Transcoders).</td>
</tr>
</tbody>
</table>
| Media Transcoders Elements Availability | Displays the number of Media Transcoders, out of all configured Media Transcoders in the Media Transcoders table, that are available for processing transcoding sessions. This is displayed in the following syntax: 
"<available>/<total> Available".  
**Note:** A Media Transcoder is considered available if it is in "connected" status. |
| HA Usage | Displays the percentage threshold of utilized DSP resources. If the Media Transcoding Cluster is configured to operate in Best Effort mode, the current utilization might increase to over 100%, indicating that there is no guarantee for HA redundancy for all the transcoding sessions. |
| HA Status | Displays the HA status:  
- "Full HA": Sufficient DSP resources exist to ensure HA for active transcoding sessions if a Media Transcoder fails.  
- "Partial HA": Sufficient DSP resources exist only for some of the active transcoding sessions if a Media Transcoder fails. This can occur, for example, if a Media Transcoder is undergoing a software upgrade when another Media Transcoder fails.  
- "No HA": No resources for HA and therefore, if a Media Transcoder fails, all its active transcoding sessions are terminated. |

### 40.3 Viewing Status per Media Transcoder on Cluster Manager

You can view status and various configuration information of each Media Transcoder from the Cluster Manager.

- **To view Media Transcoder status and configuration information:**
  1. Open the Media Transcoders table (see Step 6: Adding Media Transcoder to Cluster Manager on page 631).
2. Select the required Media Transcoder; the status and configuration information is displayed below the table, as shown in the example:

**Figure 40-2: Viewing Media Transcoder Status and Information**

<table>
<thead>
<tr>
<th>Media Transcoder Status and Information in Media Transcoders Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameter</strong></td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td><strong>Device Settings</strong></td>
</tr>
<tr>
<td>Cluster Manager IP Address</td>
</tr>
</tbody>
</table>
| Admin State | Displays the administrative state of the Media Transcoder, performed by the Cluster Manager:  
  - "Unlocked": Media Transcoder is currently unlocked.  
  - "Locked": Cluster Manager has locked the Media Transcoder.  
  - "Locked before Reset": Media Transcoder is currently locked as the Cluster Manager is initiating a reset on the Media Transcoder. |
### Parameter | Description
---|---
**Utilization** | Displays the percentage of channels currently utilized on the Media Transcoder. For example, if the Media Transcoder supports a maximum of 100 channels and 90 are in use, utilization is 90%.

### Device Information

| Parameter                  | Description |
---|---|
Media Transcoder Type | Displays the Media Transcoder type ("MT" or "vMT"). |
Version | Displays the software version currently running on the Media Transcoder. |
Serial Number | Displays the serial number of the Media Transcoder. |

### Device Status

| Parameter | Description |
---|---|
Status | Displays the status of the Media Transcoder: |
- "Connecting": Cluster Manager is trying to connect to Media Transcoder. |
- "Connected": Cluster Manager is connected to Media Transcoder. |
- "Reset in Progress": DSP channels are blocked and waiting for all active channels to be transferred or closed after graceful timeout. |
- "Waiting to Reconnect": Cluster Manager is waiting to connect to Media Transcoder after a reset. |
- "Lock in Progress": DSP channels are blocked and waiting for all active channels to be transferred or closed after timeout. |
- "Locked": Cluster Manager is connected to Media Transcoder, but Media Transcoder is locked. |
- "Disconnected": Cluster Manager has lost connectivity with Media Transcoder (Media or Control) and retrying to connect. |
- "Hardware Failure": Media Transcoder has a critical hardware failure due to high temperature (overheating). |
- "Software Upgrade": Cluster Manager is loading a software version file to Media Transcoder. |
- "Software Upgrade Failure": Cluster Manager is unable to load software version or Auxiliary file (CPT or PRT) to Media Transcoder. |

Device Up Time | Displays the duration that the Media Transcoder is operational after a reset. The duration is displayed in the syntax **DD:HH:MM:SS**. The duration resets when the Media Transcoder resets. |

Alarm Severity | Displays the highest severity of all currently raised SNMP alarms. |
- "None": No alarms raised |
- "Minor": Highest alarm severity raised is a Minor alarm. |
- "Major": Highest alarm severity raised is a Major alarm. |
- "Critical": Highest alarm severity raised is a Critical alarm. |

### Fan Tray

| Parameter                  | Description |
---|---|
Fan Tray 1 / 2 Status | Displays the status of the Fan Tray modules of the Media Transcoder: |
- "OK": |
- "Not Exist" |
- "N/A" |

### Power Supply
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply 1 / 2 Status</td>
<td>Displays the status of the Power Supply modules of the Media Transcoder:</td>
</tr>
<tr>
<td></td>
<td>- Fan Tray module status:</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;OK&quot;: Power Supply module operating normally</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;Missing&quot;: Power Supply module not installed in chassis slot</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;Failure&quot;: Faulty Power Supply module</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;N/A&quot;: Unable to provide status</td>
</tr>
<tr>
<td></td>
<td>- Fan speed status:</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;LOW SPEED&quot;</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;MEDIUM SPEED&quot;</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;HIGH SPEED&quot;</td>
</tr>
<tr>
<td></td>
<td>✓ &quot;N/A&quot;: Unable to provide status</td>
</tr>
</tbody>
</table>

**DSP Information**

- **DSP SW Name**: Displays the DSP software name.
- **DSP Count**: Displays the number of DSPs on the Media Transcoder.
- **DSP SW Version**: Displays the DSP software version running on the Media Transcoder.
- **DSP Utilization**: Displays the percentage of DSPs currently in use on the Media Transcoder. For example, if the DSPs on the Media Transcoder can support a maximum of 1,000 channels and only 100 channels are currently in use, DSP utilization is 10%.

**Network Interface**: Displays the IP network interfaces configured on the Media Transcoder (in the IP Interfaces table).

**Network Group**: Displays the Ethernet Groups configured on the Media Transcoder (in the Ethernet Groups table).
### 40.4 Viewing Media Transcoder Status

You can view the status of a Media Transcoder through the Media Transcoder's Web interface.

➢ **To view Media Transcoder status:**

1. Access the Media Transcoder's Web interface.
2. Open the Cluster Settings page (Setup menu > IP Network tab > Transcoding Cluster folder > MTC Settings).

![Figure 40-3: Viewing Media Transcoder Status](image)

The following information is displayed:

**Description of Media Transcoder Status Fields**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT Name</td>
<td>Displays the name of the Media Transcoder as configured on the Cluster Manager in the Media Transcoders table (see Step 6: Adding Media Transcoder to Cluster Manager on page 631).</td>
</tr>
<tr>
<td>MT ID</td>
<td>Displays the table row index number of the Media Transcoder as configured on the Cluster Manager in the Media Transcoders table.</td>
</tr>
<tr>
<td>MT Admin State</td>
<td>Displays the administrative state of the Media Transcoder, performed by the Cluster Manager in the Media Transcoders table:</td>
</tr>
<tr>
<td></td>
<td>&quot;Unlocked&quot;: Media Transcoder is currently unlocked.</td>
</tr>
<tr>
<td></td>
<td>&quot;Locked&quot;: Cluster Manager has locked the Media Transcoder.</td>
</tr>
<tr>
<td></td>
<td>&quot;Locked before Reset&quot;: Media Transcoder is currently locked as the Cluster Manager is initiating a reset on the Media Transcoder.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| MT Functionality Status       | Displays the operating status of the Media Transcoder:  
- "Connecting": Cluster Manager is trying to connect to Media Transcoder.  
- "Connected": Cluster Manager is connected to Media Transcoder.  
- "Reset in Progress": DSP channels are blocked and waiting for all active channels to be transferred or closed after graceful timeout.  
- "Waiting to Reconnect": Cluster Manager is waiting to connect to Media Transcoder after a reset.  
- "Lock in Progress": DSP channels are blocked and waiting for all active channels to be transferred or closed after timeout.  
- "Locked": Cluster Manager is connected to Media Transcoder, but Media Transcoder is locked.  
- "Disconnected": Cluster Manager has lost connectivity with Media Transcoder (Media or Control) and retrying to connect.  
- "Hardware Failure": Media Transcoder (MT-type only) has a critical hardware failure due to high temperature (overheating).  
- "SW Upgrade in Progress": Loading a software version file from Cluster Manager.  
- "SW Upgrade Failure": Cluster Manager is unable to load software version or Auxiliary file (CPT or PRT) to Media Transcoder. |
Part IX

Maintenance
41 Basic Maintenance

This section describes basic maintenance procedures.

41.1 Resetting the Device

You can reset the device through the device's management tools. Device reset may be required for maintenance purposes. Certain parameters require a device reset for their settings to take effect. These parameters are displayed in the Web interface with the lightning symbol. In addition, whenever you do any configuration change that requires a reset, the Reset button on the Web interface's toolbar is displayed with a red border, as shown below:

![Reset button](image)

The Web interface also provides you with the following options when resetting the device:

- Save current configuration to the device's flash memory (non-volatile) prior to reset
- Reset the device only after a user-defined time (Graceful Shutdown) to allow current calls to end (calls are terminated after this interval)

To reset the device (and save configuration to flash) through CLI, use the following command:

```
# reset now
```

To reset the device through Web interface:

1. Open the Maintenance Actions page:
   - Toolbar: Click the Reset button.

   ![Maintenance Actions](image)

2. From the 'Save To Flash' drop-down list, select one of the following:
   - **Yes**: Current configuration is saved (burned) to flash memory prior to reset (default).
   - **No**: The device resets without saving the current configuration to flash. All configuration done after the last configuration save will be discarded (lost) after reset.

3. From the 'Graceful Option' drop-down list, select one of the following:
   - **Yes**: Reset starts only after a user-defined time, configured in the 'Shutdown Timeout' field (see next step). During this interval, no new traffic is accepted. If no traffic exists and the time has not yet expired, the device resets immediately.
   - **No**: Reset begins immediately, regardless of traffic. Any existing traffic is immediately terminated.
4. In the 'Shutdown Timeout' field (available only if the 'Graceful Option' field is configured 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.

6. Click OK to confirm device reset; if the 'Graceful Option' field is configured 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 to notify you.

### 41.2 Remotely Resetting Device using SIP NOTIFY

The device can be remotely reset upon the receipt of a SIP NOTIFY that contains an Event header that is set to 'check-sync;reboot=true' (proprietary to AudioCodes), 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 Gateway Advanced Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway Advanced Settings).

2. From the 'Remote Management by Notify' (EnableSIPRemoteReset) drop-down list, select Enable:

   ![Figure 41-2: Resetting Device by SIP NOTIFY](image)

3. Click Apply.
41.3 Locking and Unlocking the Device

You can lock the device so that it stops processing calls. This may be useful, for example, when you want to upload new software files to the device and you don't want any traffic to interfere with the process. Locking the device may be done gracefully, whereby the device stops accepting new calls, but allows existing calls to continue for up to a user-defined duration before terminating them.

➢ To lock the device:

1. Open the Maintenance Actions page:
   - Toolbar: Click the Reset button.

   Figure 41-3: Locking and Unlocking the Device

   ![Locking and Unlocking the Device]

2. From the 'Graceful Option' drop-down list, select one of the following options:
   - Yes: The device locks only after a user-defined time, configured in the 'Lock Timeout' field (see next step). During this interval, no new traffic is accepted. If no traffic exists and the time has not yet expired, the device locks immediately.
   - No: The device locks immediately, terminating all existing traffic.

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

3. If you configured 'Graceful Option' to Yes (see previous step), then in the 'Lock Timeout' field, enter the time (in seconds) after which the device locks.

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

5. Click OK to confirm; if you configured 'Graceful Option' to Yes, a lock icon is displayed and a window appears displaying the number of remaining calls and time. If you configured 'Graceful Option' to No, the lock process begins immediately. The 'Gateway Operational State' read-only field displays "LOCKED" and the device does not process any calls.

➢ To unlock the device:

- Click the UNLOCK button; the device unlocks immediately and accepts new incoming calls. The 'Gateway Operational State' read-only field displays "UNLOCKED".
41.4  Saving Configuration

When you configure parameters and tables in the Web interface and then click the Apply button on the pages in which the configurations are done, changes are saved to the device's volatile memory (RAM). These changes revert to their previous settings if the device subsequently resets (hardware or software) or powers down. Therefore, to ensure that your configuration changes are retained, you must save them to the device's non-volatile memory (i.e., flash memory).

To save your settings to flash, click the Save button located on the toolbar. To remind you to save your settings to flash, the Save button is displayed with a red border, as shown below:

Figure 41-4: Saving Configuration to Flash

To save configuration to flash through CLI, use the following command:

```
# write
```

Note: Saving configuration to flash 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 651).
42 Channel Maintenance

This chapter describes various channel-related maintenance procedures.

42.1 Disconnecting Active Calls

You can forcibly disconnect all active calls, or disconnect specific calls based on Session ID.

- To disconnect calls through CLI:
  - Disconnect all active calls:
    ```bash
    # clear voip calls
    ```
  - Disconnect active calls belonging to a specified Session ID:
    ```bash
    # clear voip calls <Session ID>
    ```
This page is intentionally left blank.
43 Auxiliary Files

You can load the following Auxiliary files to the device.

Table 43-1: Auxiliary Files

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INI</td>
<td>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 through ini file. For more information, see &quot;INI File-Based Management&quot; on page 97.</td>
</tr>
<tr>
<td>Call Progress Tones</td>
<td>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 &quot;Call Progress Tones File&quot; on page 657.</td>
</tr>
<tr>
<td>Prerecorded Tones</td>
<td>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 &quot;Prerecorded Tones File&quot; on page 660.</td>
</tr>
<tr>
<td>Dial Plan</td>
<td>Provides dialing plans, for example, for obtaining the destination IP address for outbound IP routing. For more information, see &quot;Dial Plan File&quot; on page 661.</td>
</tr>
<tr>
<td>User Info</td>
<td>The User Information file maps PBX extensions to IP numbers. The file can be used to represent PBX extensions as IP phones in the global 'IP world'. For more information, see &quot;User Information File&quot; on page 663.</td>
</tr>
<tr>
<td>AMD Sensitivity</td>
<td>Answer Machine Detector (AMD) Sensitivity file containing the AMD Sensitivity suites. For more information, see AMD Sensitivity File on page 667.</td>
</tr>
<tr>
<td>SBC Wizard Template Package</td>
<td>Contains the vendor-interoperability configuration templates for the SBC Configuration Wizard. For more information, see SBC Configuration Wizard on page 711.</td>
</tr>
</tbody>
</table>

43.1 Loading Auxiliary Files

You can load Auxiliary files to the device using one of the following methods:

- Web interface - see "Loading Auxiliary Files through Web Interface" on page 656
- CLI - see Loading Auxiliary Files through CLI on page 657
- TFTP - see Loading Auxiliary Files through ini File using TFTP
- EMS (Software Manager) – refer to the EMS User’s Manual

Note:

- You can automatically load Auxiliary files from a remote server using the device's Automatic Update mechanism (see Automatic Update Mechanism).
- Saving Auxiliary files to flash memory may disrupt traffic on the device. To avoid this, disable all traffic on the device by performing a graceful lock as described in "Locking and Unlocking the Device" on page 651.
43.1.1 Loading Auxiliary Files through Web Interface

The following procedure describes how to load Auxiliary files through the Web interface.

**Note:**
- When loading an ini file through the Auxiliary Files page (as described in this section), only parameter settings specified in the ini file are applied to the device; all other parameters remain at their current settings.
- If you load an ini file containing Auxiliary file(s), the Auxiliary files specified in the file overwrite the Auxiliary files currently installed on the device.

➢ To load Auxiliary files through Web interface:

1. Open the Auxiliary Files page:
   - Toolbar: From the **Actions** drop-down menu, choose **Auxiliary Files**.
   - Navigation tree: **Setup** menu > **Administration** tab > **Maintenance** folder > **Auxiliary Files**.

   ![INI file (Incremental)](Browse... No file selected. Load File)

   ![Voice Prompts file](Browse... No file selected. Load File)

   ![Call Progress Tones file](Browse... No file selected. Load File)

   ![Prerecorded Tones file](Browse... No file selected. Load File)

   ![Dial Plan file](Browse... No file selected. Load File)

2. Click the **Browse** button corresponding to the Auxiliary file type that you want to load, navigate to the folder in which the file is located, and then click **Open**; the name of the file appears next to the **Browse** button.
3. Click the corresponding **Load File** button.
4. Repeat steps 2 through 3 for each file you want to load.
5. Reset the device with a save-to-flash for your settings to take effect (if you have loaded a Call Progress Tones file).
43.1.2 Loading Auxiliary Files through CLI

You can load Auxiliary files from remote servers through CLI:

- **Single Auxiliary file:**
  
  `# copy <file> from <URL of remote server>`
  
  For example:
  
  `# copy call_progress_tones from http://192.169.11.11:80/cpt_us.dat`

- **Multiple (batch) Auxiliary files:** The Auxiliary files must be contained in a TAR (Tape ARchive) file (.tar). The TAR file can contain any number and type of Auxiliary files (e.g., Dial Plan file and CPT file).

  `# copy aux-package from | to <URL of remote server with TAR file name>`
  
  For example:
  
  `# copy aux-package from http://192.169.11.11:80/aux_files.tar`

For more information on CLI, refer to the *CLI Reference Guide*.

43.2 Deleting Auxiliary Files

You can delete loaded Auxiliary files through the Web interface, as described below.

- **To delete a loaded Auxiliary file:**

  1. Open the Device Information page (see "Viewing Device Information" on page 729); the loaded files are listed under the Loaded Files group:

     Figure 43-1: List of Loaded Auxiliary Files

     ![Figure 43-1: List of Loaded Auxiliary Files](image)

     Table: Loaded Files

     | Call Progress Tones File Name: | usa_tones_13.dat | Delete |
     |--------------------------------|-----------------|--------|
     | Loaded Codec Table :          | Default CODERTABLE |        |

  2. Click the **Delete** button corresponding to the file that you want deleted; a confirmation message box appears.

  3. Click **OK** to confirm.

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

43.3 Call Progress Tones File

The Call Progress Tones (CPT) Auxiliary file contains definitions of the CPT (levels and frequencies) that are detected and generated by the device.

You can use one of the supplied Auxiliary files (.dat file format) or create your own file. To create your own file, it's recommended to modify the supplied `usa_tone.ini` file (in any standard text editor) to suit your specific requirements and then convert the modified `ini` file into binary `dat` file format, using AudioCodes DConvert utility. For more information, refer to the *DConvert Utility User's Guide*. 
Note: The CPT file can only be loaded in .dat file format.

You can create up to 32 different Call Progress Tones, each with frequency and format attributes. The frequency attribute can be single or dual-frequency (in the range of 300 to 1980 Hz) or an Amplitude Modulated (AM). Up to 64 different frequencies are supported. Only eight AM tones, in the range of 1 to 128 kHz, can be configured (the detection range is limited to 1 to 50 kHz). Note that when a tone is composed of a single frequency, the second frequency field must be set to zero.

The format attribute can be one of the following:

- **Continuous**: A steady non-interrupted sound (e.g., a dial tone). Only the 'First Signal On time' should be specified. All other on and off periods must be set to zero. In this case, the parameter specifies the detection period. For example, if it equals 300, the tone is detected after 3 seconds (300 x 10 msec). The minimum detection time is 100 msec.

- **Cadence**: A repeating sequence of on and off sounds. Up to four different sets of on/off periods can be specified.

- **Burst**: A single sound followed by silence. Only the 'First Signal On time' and 'First Signal Off time' should be specified. All other on and off periods must be set to zero. The burst tone is detected after the off time is completed.

You can specify several tones of the same type. These additional tones are used only for tone detection. Generation of a specific tone conforms to the first definition of the specific tone. For example, you can define an additional dial tone by appending the second dial tone's definition lines to the first tone definition in the ini file. The device reports dial tone detection if either of the two tones is detected.

The Call Progress Tones section of the ini file comprises the following segments:

- **[NUMBER OF CALL PROGRESS TONES]**: Contains the following key: 'Number of Call Progress Tones' defining the number of Call Progress Tones that are defined in the file.

- **[CALL PROGRESS TONE #X]**: containing the Xth tone definition, starting from 0 and not exceeding the number of Call Progress Tones less 1 defined in the first section (e.g., if 10 tones, then it is 0 to 9), using the following keys:

  - **Tone Type**: Call Progress Tone types:
    - [1] Dial Tone
    - [2] Ringback Tone
    - [3] Busy Tone
    - [4] Congestion Tone
    - [6] Warning Tone
    - [7] Reorder Tone
    - [17] Call Waiting Ringback Tone (heard by the calling party)
    - [18] Comfort Tone
    - [23] Hold Tone
    - [46] Beep Tone

  - **Tone Modulation Type**: Amplitude Modulated (1) or regular (0)

  - **Tone Form**: The tone's format can be one of the following:
    - Continuous (1)
    - Cadence (2)
    - Burst (3)
• Low Freq [Hz]: Frequency (in Hz) of the lower tone component in case of dual frequency tone, or the frequency of the tone in case of single tone. This is not relevant to AM tones.

• High Freq [Hz]: Frequency (in Hz) of the higher tone component in case of dual frequency tone, or zero (0) in case of single tone (not relevant to AM tones).

• Low Freq Level [-dBm]: Generation level 0 dBm to -31 dBm in dBm (not relevant to AM tones).

• High Freq Level: Generation level of 0 to -31 dBm. The value should be set to 32 in the case of a single tone (not relevant to AM tones).

• First Signal On Time [10 msec]: 'Signal On' period (in 10 msec units) for the first cadence on-off cycle. For continuous tones, the parameter defines the detection period. For burst tones, it defines the tone's duration.

• First Signal Off Time [10 msec]: 'Signal Off' period (in 10 msec units) for the first cadence on-off cycle (for cadence tones). For burst tones, the parameter defines the off time required after the burst tone ends and the tone detection is reported. For continuous tones, the parameter is ignored.

• Second Signal On Time [10 msec]: 'Signal On' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.

• Second Signal Off Time [10 msec]: 'Signal Off' period (in 10 msec units) for the second cadence on-off cycle. Can be omitted if there isn't a second cadence.

• Third Signal On Time [10 msec]: 'Signal On' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.

• Third Signal Off Time [10 msec]: 'Signal Off' period (in 10 msec units) for the third cadence on-off cycle. Can be omitted if there isn't a third cadence.

• Fourth Signal On Time [10 msec]: 'Signal On' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.

• Fourth Signal Off Time [10 msec]: 'Signal Off' period (in 10 msec units) for the fourth cadence on-off cycle. Can be omitted if there isn't a fourth cadence.

• Carrier Freq [Hz]: Frequency of the carrier signal for AM tones.

• Modulation Freq [Hz]: Frequency of the modulated signal for AM tones (valid range from 1 to 128 Hz).

• Signal Level [-dBm]: Level of the tone for AM tones.

• AM Factor [steps of 0.02]: Amplitude modulation factor (valid range from 1 to 50). Recommended values from 10 to 25.

Note:

• 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.

Below shows an example of a configured dial tone to 440 Hz only:

```
[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

43.4 Prerecorded Tones File

The Prerecorded Tone (PRT) file contains user-defined prerecorded tones that can be played by the device. For example, it can be used to play a held tone (music on hold) to a call party that has been put on hold or a ringback tone to a calling party. The PRT file overcomes the limitations of the CPT file (such as limited number of predefined tones and limited number of frequency integrations in a single tone).

The PRT file also lets you play different held and ringback tones for different groups of users. To do this, configure an IP Profile with the required ringback tone (IPProfile_LocalRingbackTone) and/or held tone (IPProfile_LocalHeldTone) from the PRT file, and then associate the IP Profile with the required IP Group.

You can define a PRT file with multiple tones of the same tone type but with different coders. If one of the tones is defined with the same coder as used in the current call, the device always selects it in order to eliminate the need for using DSP resources. If the coder of the tone is the same as that of the call, DSPs are not required. If they are different, DSPs are required.

Note:

- The PRT file only generates (plays) tones; detection of tones is according to the CPT file.
- If the PRT file contains a tone that also exists in the CPT file, the tone in the PRT file is played instead (i.e., overrides the tone in the CPT file).
- The device does not require DSPs for playing tones from a PRT file if the coder defined for the tone is the same as that used by the current call. If the coders are different, the device uses DSPs.
- Local generation of tones is not supported.
- For SBC calls, the PRT file supports only the ringback tone and hold tone.

You can include up to 80 user-defined tones (a maximum of 10 minutes) in a PRT file. The tones can be recorded using a standard third-party, recording utility (such as Adobe Audition), and then combined into a single and loadable PRT file (.dat) using the latest version of AudioCodes DConvert utility (refer to the DConvert Utility User's Guide). Once created, you need to install the PRT file on the device (flash memory), using the Web interface (see 'Loading Auxiliary Files' on page 655) or CLI.

You must record the tones (raw data files) with the following properties:

- Coders: G.711 A-law or G.711 µ-law (and other coders)
- Rate: 8 kHz
- Resolution: 8-bit
- Channels: mono
The prerecorded tones are played repeatedly. This allows you to record only part of the
tone and then play the tone for the full duration. For example, if a tone has a cadence of 2
seconds on and 4 seconds off, the recorded file should contain only these 6 seconds. The
device repeatedly plays this cadence for the configured duration. Similarly, a continuous
tone can be played by repeating only part of it.

43.5 Dial Plan File

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

Note: The Dial Plan described in this section is for backward compatibility purposes
only. For the new Dial Plan method, see Configuring Dial Plans on page 541.

43.5.1 Creating a Dial Plan File

Note: The Dial Plan described in this section is for backward compatibility purposes
only. For the new method, see Configuring Dial Plans on page 541.

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 655.
5. Select the Dial Plan index that you want to use. This depends on the feature and is
   described in the respective section.
**43.5.2 Obtaining IP Destination from Dial Plan File**

You can use a Dial Plan index listed in a loaded Dial Plan file for determining the IP destination of SBC (see note below) calls. This enables the mapping of called numbers to IP addresses (in dotted-decimal notation) or FQDNs (up to 15 characters).

**Note:** For the SBC application, the method described in this section for obtaining an IP address using the Dial Plan file is for backward compatibility purposes only. For the new method, see Configuring Dial Plans on page 541.

- 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 661).
  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.

**43.5.3 Viewing Information of Installed Dial Plan File**

You can view information about the Dial Plan file currently installed on the device, through the device's CLI:

- **Viewing Dial Plan file information:** You can view the file name of the installed Dial Plan file and the names of the Dial Plans defined in the Dial Plan file, by entering the following CLI command (in Enable mode):
  ```
  # debug auxiliary-files dial-plan info
  ```
  
  For example, the following shows the file name of the installed Dial Plan file and lists its Dial Plans:
  ```
  # debug auxiliary-files dial-plan info
  File Name: MyDialPlan.txt
  Plans:
  Plan #0 = PLAN1
  Plan #1 = PLAN2
  ```
Note that the index number of the first Dial Plan is 0.

- **Searching a prefix number:** You can check whether a specific prefix number is defined in a specific Dial Plan (and view the corresponding tag if the Dial Plan implements tags), by entering the following CLI command (in Enable mode):

  ```
  # debug auxiliary-files dial-plan match-number <Dial Plan number> <prefix number>
  ```

  For example, the following checks whether the called prefix number 2000 is defined in Dial Plan 1, which is used for obtaining the destination IP address (tag):

  ```
  # debug auxiliary-files dial-plan match-number PLAN1 2000
  Match found for 4 digits
  Matched prefix: 2000
  Tag: 10.33.45.92
  ```

### 43.6 User Information File

This section describes the User Info table.

#### 43.6.1 Enabling the User Info Table

Before you can use the User Info table, you need to enable the User Info functionality, as described in the following procedure.

**To enable the User Info table:**

1. Make sure that your device's License Key provides far-end users support ("FEU"). To view the License Key, see "Viewing the License Key" on page 670.
2. Open the Proxy & Registration page (Setup menu > Signaling & Media tab > SIP Definitions folder > Proxy & Registration).
3. From the 'Enable User-Information Usage' drop-down list (EnableUserInfoUsage), select **Enable**.

![Figure 43-2: Enabling User Info Table](image)

**Note:** The 'Enable User-Information Usage' parameter appears in the Web interface only if the device's License Key is defined with far-end users.

4. Reset the device with a save-to-flash for your settings to take effect; the User Info table now appears in the Web interface.

#### 43.6.2 User Information File for SBC User Database

You can use the SBC User Info table for the following:

- Registering each user to an external registrar server.
- Authenticating (for any SIP request and as a client) each user if challenged by an external server.
- Authenticating as a server incoming user requests (for SBC security).
If the device registers on behalf of users and the users do not perform registration, any SIP request destined to the user is routed to the Proxy Set associated with the user’s IP Group. You can configure up to 3,000 users (table rows) in the SBC User Info table. The SBC User Info table can be configured using any of the following methods:

- Web interface - see "Configuring SBC User Info Table through Web Interface" on page 664
- CLI - see Configuring SBC User Info Table through CLI on page 665
- Loadable User Info file - see "Configuring SBC User Info Table in Loadable Text File" on page 666

### 43.6.2.1 Configuring SBC User Info Table through Web Interface

The following procedure describes how to configure the SBC User Info table through the Web interface.

#### Note:

- To configure the User Info table, make sure that you have enabled the feature, as described in "Enabling the User Info Table" on page 663.
- If you load any User Info file to the device, all previously configured entries are removed from the table in the Web interface and replaced with the entries from the loaded User Info file.

➢ To configure the SBC User Info table through the Web interface:

1. Open the SBC User Info table (Setup menu > Signaling & Media tab > SBC folder > User Information).
2. Click New; the following dialog box appears:

![Figure 43-3: SBC User Info Table - Add Dialog Box](image)

3. Configure a user according to the table below.
4. Click Apply.

To register a user, select the user's table entry, and then from the Action drop-down list, choose Register. To un-register a user, select the user, and then from the Action drop-down list, choose Un-Register.
### Table 43-2: SBC User Info Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[SBCUserInfoTable_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Local User</td>
<td>Defines the user and is used as the Request-URI user part for the AOR in the database.</td>
</tr>
<tr>
<td>[SBCUserInfoTable_LocalUser]</td>
<td>The valid value is a string of up to 10 characters.</td>
</tr>
<tr>
<td>Username</td>
<td>Defines the username for registering the user when authentication is necessary.</td>
</tr>
<tr>
<td>[SBCUserInfoTable_Username]</td>
<td>The valid value is a string of up to 40 characters.</td>
</tr>
<tr>
<td>Password</td>
<td>Defines the password for registering the user when authentication is necessary.</td>
</tr>
<tr>
<td>[SBCUserInfoTable_Password]</td>
<td>The valid value is a string of up to 20 characters.</td>
</tr>
<tr>
<td>IP Group</td>
<td>Assigns an IP Group to the user and is used as the Request-URI source host part for the AOR in the database.</td>
</tr>
<tr>
<td>[SBCUserInfoTable_IPGroupName]</td>
<td>To configure IP Groups, see &quot;Configuring IP Groups&quot; on page 345.</td>
</tr>
<tr>
<td>Status</td>
<td>(Read-only field) Displays the status of the user - &quot;Registered&quot; or &quot;Not Registered&quot;.</td>
</tr>
<tr>
<td>[SBCUserInfoTable_Status]</td>
<td></td>
</tr>
</tbody>
</table>

#### 43.6.2.2 Configuring SBC User Info Table through CLI

The SBC User Info table can be configured in the CLI using the following commands:

- **To add and/or modify a user (example):**
  ```
  # configure voip
  (config-voip)# sip-definition proxy-and-registration
  (sip-def-proxy-and-reg)# user-info sbc-user-info <index, e.g., 1>
  (sbc-user-info-1)# username JohnDee
  (sbc-user-info-1)# <activate | exit>
  ```

- **To delete a specific user, use the no command:**
  ```
  (sip-def-proxy-and-reg)# no user-info sbc-user-info <index, e.g., 1>
  ```

- **To view all table entries:**
  ```
  (sip-def-proxy-and-reg)# user-info sbc-user-info display
  ---- sbc-user-info-0 ----
  local-user (JohnDee)
  username (userJohn)
  password (s3fn+fn=)
  ip-group-id (1)
  status (not-registered)
  ---- sbc-user-info-1 ----
  local-user (SuePark)
  username (userSue)
  password (t6sn+un=)
  ```
To view a specific entry (example):

```plaintext
(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-registered)
```

To search a user by local-user:

```plaintext
(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
```

**Note:** To configure the User Info table, make sure that you have enabled the feature as described in "Enabling the User Info Table" on page 663.

### 43.6.2.3 Configuring SBC User Info Table in Loadable Text File

The SBC User Info table can be configured as a User Info file using a text-based file (*.txt). This file can be created using any text-based program such as Notepad.

You can load the User Info file using any of the following methods:

- Web interface - see "Loading Auxiliary Files" on page 655
- *ini* file, using the UserInfoFileName parameter - see "Auxiliary and Configuration File Name Parameters" on page 843
- 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:

```plaintext
[SBC]
FORMAT LocalUser,UserName,Password,IPGroupID
```

where:

- `[SBC]` indicates that this part of the file is the SBC User Info table
- `LocalUser` is the user and is used as the Request-URI user part for the AOR in the database
- `UserName` is the user's authentication username
- `Password` is the user's authentication password
- `IPGroupID` is the IP Group ID to which the user belongs and is used as the Request-URI source host part for the AOR in the database

**Note:**

- Make sure that there are no spaces between the values.
- To modify the SBC User Info table using a User Info file, you need to load to the device a new User Info file containing your modifications.
Below is an example of a configured User Info file:

```
[ SBC ]
FORMAT LocalUser,UserName,Password,IPGroupID
john,john_user,john_pass,2
sue,sue_user,sue_pass,1
```

### 43.6.3 Viewing the Installed User Info File Name

You can view the name of the User Info file currently installed on the device, through the device's CLI (in Enable mode):
```
# debug auxiliary-files user-info info
```

For example:
```
# debug auxiliary-files user-info info
User Info File Name MyUsers.txt
```

### 43.7 AMD Sensitivity File

The device is shipped with a default, pre-installed AMD Sensitivity file for its Answering Machine Detection (AMD) feature. This file includes the detection algorithms for detecting whether a human or answering machine has answered the call, and is based on North American English. In most cases, the detection algorithms in this file suffice even when your deployment is in a region where a language other than English is spoken. However, if you wish to replace the default file with a different AMD Sensitivity file containing customized detection algorithms, please contact your AudioCodes sales representative for more information.

The AMD Sensitivity file is created in .xml format and then converted to a binary .dat file that can be installed on the device. The XML-to-binary format conversion can be done using AudioCodes DConvert utility. For more information on using this utility, refer to DConvert Utility User's Guide. Only one AMD Sensitivity file can be installed on the device.

To install a new AMD Sensitivity file, use any of the following methods:
- Web interface: Auxiliary Files page - see "Loading Auxiliary Files" on page 655.
- TFTP during initialization: You need to configure the ini file parameter, AMDSensitivityFileName, and then copy the AMD Sensitivity file to the TFTP directory.
- Automatic Update feature: For more information, see Automatic Update Mechanism. For this method, the AMDSensitivityFileUrl parameter must be set through SNMP or ini file.

For more information on the AMD feature, see "Answering Machine Detection (AMD)" on page 200.
This page is intentionally left blank.
44 License Key

The License Key determines the device’s supported features and call capacity, as ordered from your AudioCodes sales representative. You can upgrade or change your device’s supported features and capacity, by purchasing and installing a new License Key that match your requirements.

Note:
- By default, the device is shipped with a pre-installed License Key that enables up to three call sessions. Once you have installed the Mediant Software E-SBC, you need to obtain and activate your License Key, as described in Obtaining License Key for Initial Activation on page 671.
- The availability of certain Web pages depends on the installed License Key.

44.1 Viewing the License Key

The following procedure describes how to view the device’s License Key.

➢ To view the License Key:

■ Open the License Key page:
  - Toolbar: From the Actions drop-down menu, choose License Key.
  - Navigation tree: Setup menu > Administration tab > Maintenance folder > License Key.
The License Key page displays your ordered features and capabilities as well as the following general information:

- **Product Key**: Device's Product Key - for more information, see Viewing the Device's Product Key on page 681.
- **Serial Number**: Device's serial number.
- **Board Type**: AudioCodes internal identification number of the type of your device.
- **Remote License Server / Remote License Server IP**: For more information, see Upgrading SBC Capacity Licenses by License Pool Manager Server on page 679.

### 44.2 Free Product Evaluation for Mediant VE

By default, the product software installation provides you with a free license for a maximum of three concurrent sessions (signaling and media) and three user registrations (far-end users). This allows you to evaluate the product prior to purchasing it with your required capacity and features.

Note that if you want to evaluate the product's transcoding capabilities, you need to do the following:

1. Depending on platform:
   - **OpenStack-KVM / Hyper-V / VMware**: Configure the virtual machine to 2 vCPUs with 4-GB memory.
   - **AWS**: Configure the instance to c4.2xlarge.

2. Once installation is complete, install the following License Key, as described in Section 44.4:
44.3 Obtaining License Key for Initial Activation

The following procedure describes how to obtain and activate your License Key for initial activation of your device.

**Note:**
- License activation is intended **only** for first-time software activation upon product purchase (or if your License Key is "lost", due to whatever reason). For subsequent software feature upgrades, the License Key file is e-mailed to you after your Purchase Order has been processed.
- For HA, each unit has its own Serial Number, Product Key and License Key. Therefore, the instructions in this section must be done for each unit.

➢ **To obtain and activate your License Key for initial activation:**


   ![Figure 44-2: Software License Activation Tool](image_url)
2. Enter the following information:

- **Product Key**: The Product Key identifies your specific device purchase for the purpose of subsequent communication with AudioCodes (for example, for support and software upgrades). The Product Key is provided in the Order Confirmation e-mail sent to you by AudioCodes upon your purchase, as shown in the example below:

  **Figure 44-3: Product Key in Order Confirmation E-mail**

  ![Product Key in Order Confirmation E-mail](image)

  **Note**: For 1+1 High-Availability orders, you are provided with two Product Keys, one for each unit. In such cases, you need to perform license activation twice in order to obtain License Keys for both units.

- **Fingerprint**: The fingerprint is the device’s Serial Number. The Serial Number uniquely identifies the software installation. The Serial Number is displayed in the ‘Serial Number’ field on the Device Information page (Monitor menu > Monitor menu > Summary tab > Device Information).

- **Email**: Provide one or more e-mail addresses to where you want the License Key to be sent.

3. Click **Send** to submit your license activation request.

4. Once AudioCodes processes and completes your license activation, you will receive an e-mail notification with the License Key file attached. Open the file with any text-based program (such as Notepad) and make sure that the serial number (“S/N”) in the License Key is correct and reflects the Serial Number of your device.

**Warning**: Do not modify the contents of the License Key file.

---

### 44.4 Installing a New License Key

This section describes how to install a new License Key on the device.

**Note**: When you install a new License Key, it overwrites the previously installed License Key. Any license-based features that were included in the old License Key, but not included in the new License Key, will no longer be available.
44.4.1 Installing License Key through Web Interface

This section describes how to install a License Key through the Web interface.

When you first load a License Key (before installing it) to the device, the License Key page uses color-coded icons to indicate the changes between the previous License Key and the new License Key, as described in the following table:

### Color-Coded Icons for Newly Installed License Key

<table>
<thead>
<tr>
<th>Icon Color</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Green" /></td>
<td>Indicates new features added by the new License Key.</td>
</tr>
</tbody>
</table>
| ![Orange](image) | Indicates the capacity change of an existing feature. Move your mouse over the icon to view a pop-up describing the capacity change, as shown in the following example for SIPRec Sessions:

```plaintext
VOIP FEATURES

Voice Quality Monitoring
RTCP-XR
Media Enhancement
SIPRec Sessions

Value changed from 6 to 10

10
```

| ![Red](image) | Indicates features from the previous License Key that were not included in the new License Key and are no longer available. |

---

**Note:** After you install the License Key (device reset with a save-to-flash), the icons no longer appear and the License Key page displays only features and capacity of the new License Key.

44.4.1.1 Installing a License Key String

You can install a License Key in encrypted string format, through the Web interface.

**Note:**
- The License Key installation process includes a device reset and therefore, is traffic-affecting. To minimize the disruption of current calls, it is recommended to perform this procedure during periods of low traffic.
- Installation of License Keys in string format is not applicable to devices in HA mode.

➢ **To install a License Key string through Web interface:**

1. Open the License Key page (see Viewing the License Key on page 669).

5. Back up the currently installed License Key, as a precaution. If the new License Key does not comply with your requirements, you can re-load this backed-up License Key to restore the device's original capabilities. For backing up the License Key, see Backing up the License Key on page 681.

2. Copy the License Key string (from the License Key file or e-mail) to your clipboard.
Make sure that you copy only the encrypted string (and not the serial number or any other part of the string), as shown in the example below:

**Figure 44-4: Copying a License Key String**

3. Click **Load By String**; the Load Feature Key By String dialog box appears.

4. In the text box, paste your License Key string, as shown in the following example:

**Figure 44-5: Pasting License Key String in Load Feature Key By String Dialog Box**

5. Click **Apply**; the dialog box closes and the **Apply New License Key** button appears. The License Key page uses color-coded icons to indicate the changes between the previous License Key and the newly loaded License Key (for more information, see Installing License Key through Web Interface on page 673).

6. Click **Apply New License Key**; the following message box appears:

**Figure 44-6: Apply New License Key Message**

7. Click **Reset**; the device begins to save the file to flash memory with a reset and the following progress message box appears:

**Figure 44-7: Reset in Progress for License Key**
When the installation completes, the following message box appears:

**Figure 44-8: Reset and Save-to-Flash Success Message**

Click **Close** to close the message box; you are logged out of the Web interface and prompted to log in again. The features and capabilities displayed on the License Key page now reflect the newly installed License Key.

### 44.4.1.2 Installing a License Key File

You can install a License Key by loading a License Key file through the Web interface. The procedure depends on whether the device is standalone (see Installing on Standalone Devices on page 675) or operating in HA mode (see Installing on HA Devices on page 676).

#### 44.4.1.2.1 Installing on Standalone Devices

When the device operates as a standalone device, the installation of a License Key can also be done from a License Key file (or from a string, as described in Installing a License Key String on page 673).

**Note:** The License Key installation process includes a device reset and is therefore, traffic-affecting. To minimize the disruption of current calls, it is recommended to perform this procedure during periods of low traffic.

➢ **To install a License Key file for standalone devices through Web interface:**

1. Open the License Key page (see Viewing the License Key on page 669).
2. Back up the currently installed License Key, as a precaution. If the new License Key does not comply with your requirements, you can re-load this backed-up License Key to restore the device’s original capabilities. For backing up the License Key, see Backing up the License Key on page 681.
3. Click the **Load By File** button, navigate to the License Key file on your computer, and then select the file to load to the device; the **Apply New License Key** button appears. The License Key page uses color-coded icons to indicate the changes between the previous License Key and the newly loaded License Key (for more information, see Installing License Key through Web Interface on page 673).

**Note:** If want to cancel installation, reset the device without a save to flash. For more information, see Resetting the Device on page 649.

6. Click **Apply New License Key**; the following message box appears:
4. Click **Reset**; the device begins to save the file to flash memory with a reset and the following progress message box appears:

**Figure 44-10: Reset in Progress for License Key**

![Apply New License Key](image)

When installation completes, the following message box appears:

**Figure 44-11: Reset and Save-to-Flash Success Message**

![Apply New License Key](image)

5. Click **Close** to close the message box; you are logged out of the Web interface and prompted to log in again. The features and capabilities displayed on the License Key page now reflect the newly installed License Key.

### 44.4.1.2.2 Installing on HA Devices

When the device operates in HA mode, the installation of a License Key can be done only from a License Key file. You can install the License Key using one of the following methods:

- **Hitless Upgrade**: The License Key is installed on both devices without affecting traffic, by employing the following mechanism:
  1. The License Key file is loaded to the active device.
  2. The active device sends the file to the redundant device.
  3. The redundant device installs the file, by saving it to flash memory with a reset.
  4. The active device saves the file to flash memory (without a reset).
  5. The devices undergo an HA switchover, whereby the active device becomes redundant and the redundant device becomes active. Current calls are maintained and handled by the active device (previously the redundant device).
  6. The redundant device (previously the active device) resets to install the file.
Non-hitless Upgrade: The License Key is installed on both devices simultaneously (both reset at the same time). Therefore, this method is traffic-affecting and current calls are terminated. The installation process is as follows:

1. The License Key file is loaded to the active device.
2. The active device sends the file to the redundant device.
3. Both devices install the file, by saving it to flash memory with a reset.

**Note:**
- The License Key file for HA contains two License Keys - one for the active device and one for the redundant device. Each License Key has a different serial number ("S/N"), which reflects the serial number of each device in the HA system.
- Currently, hitless software downgrade from Version 7.2.150 to an earlier version is not supported (and the non-hitless method must be used).

➢ To install a License Key file for HA through Web interface:

1. Open the License Key page (see Viewing the License Key on page 669).
2. Back up the currently installed License Key, as a precaution. If the new License Key does not comply with your requirements, you can re-load this backed-up License Key to restore the device's original capabilities. For backing up the License Key, see Backing up the License Key on page 681.
3. Click the Load By File button, navigate to the License Key file on your computer, and then select the file to load to the device; the Apply New License Key button appears. The License Key page uses color-coded icons to indicate the changes between the previous License Key and the newly loaded License Key (for more information, see Installing License Key through Web Interface on page 673).

**Note:** If want to cancel installation, reset the device without a save to flash. For more information, see Resetting the Device 649.

4. Click Apply New License Key; the following appears:

   ![License Key page](image)

5. Click one of the following buttons:
• **Hitless Upgrade:** Installs the License Key without affecting traffic by employing the HA switchover mechanism. When you click the button, the process starts and a message box is displayed indicating the installation progress:

![Figure 44-12: Hitless License Key Upgrade - Progress](image1.png)

When installation completes, the following message box appears:

![Figure 44-13: Hitless License Upgrade Successfully Completed](image2.png)

• **Non-Hitless Upgrade:** Installs the License Key simultaneously on both devices where both undergo a reset and therefore, current calls are terminated. When you click the button, the process starts and the following progress message box appears:

![Figure 44-14: Reset in Progress for License Key](image3.png)

When installation completes, the following message box appears:

![Figure 44-15: Reset and Save-to-Flash Success Message](image4.png)

6. **Clock Close** to close the message box; you are logged out of the Web interface and prompted to log in again. The features and capabilities displayed on the License Key page now reflect the newly installed License Key.
44.4.2 Installing License Key through CLI

To install the License Key through CLI, use the following commands:

- To install the License Key:
  
  ```text
  (config-system)# feature-key "string enclosed in double quotation marks"
  ```

- To view the License Key:
  
  ```text
  show system feature-key
  ```

44.4.3 Verifying Installed License Key

Once you have finished installing the new License Key, verify that it has been installed:

- On the License Key page, check that the listed features and capabilities of the newly installed License Key match those that you ordered.
- Access the Syslog server and check that the following message appears:
  
  "S/N<serial number> Key Was Updated. The Board Needs to be Reloaded with ini file\n"

  If the Syslog server indicates that the License Key was unsuccessfully loaded (i.e., the "SN_" line is blank), do the following preliminary troubleshooting procedures:

  1. Open the 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.

44.5 Upgrading SBC Capacity Licenses by License Pool Manager Server

The device can receive SBC capacity (session) licenses from a centralized pool of SBC resources managed by the License Pool Manager Server running on AudioCodes EMS. The License Pool Manager Server can dynamically allocate and de-allocate SBC licenses from the pool to devices in the network to meet capacity demands of each device, whenever required. The License Pool Manager Server holds a pool of customer-ordered SBC capacity (resource) licenses, which includes any of the following license types:

- SBC Sessions (media and signaling)
- SBC Signaling Sessions
- SBC Transcoding Sessions
- SBC Far End Users (FEU) - SBC registrations (number of SIP endpoints that can register with the SBC)

The device can be upgraded by the License Pool Manager Server with any of the above SBC license types.

You can view the SBC licenses allocated by the License Pool Manager Server in the License Key page under the SBC Capacity group:
The number of sessions received from the License Pool Manager Server are displayed under the "remote" column while the sessions from the locally installed License Key are displayed under the "local" column. The far-right column displays the total sessions, which is the summation of the remote and local sessions. However, the total sessions is only updated once the remote license is applied to the device with a reset (initiated by the License Pool Manager or locally on the device by the management user).

Communication between the device and License Pool Manager Server is through HTTPS (port 443) and SNMP. If a firewall exists in the network, make sure that ports for these applications are opened. The connectivity status with the License Pool Manager Server is displayed in the top section of the License Key page, as shown in the example below:

**Figure 44-17: Connectivity Status with License Pool Manager Server**

- **Remote License Server**: Status of License Pool Manager Server:
  - "Enable" - the device has been configured to operate with the License Pool Manager Server.
  - "Disconnected" - the device has not been configured to operate with the License Pool Manager Server.

- **Remote License Server IP**: IP address of the License Server.

The device periodically checks with the License Pool Manager Server for SBC capacity licenses. The License Pool Manager Server identifies the device by serial number. If it has an SBC license for the device, it sends it to the device. If the device's installed License Key already includes SBC capacity figures, the SBC license allocated from the pool is simply added to it (but up to the device's maximum supported capacity capabilities). For standalone devices, you must reset the device with a save to flash for the allocated SBC license to take effect. If communication with the License Pool Manager Server is lost for a long duration, the device discards the allocated SBC license (i.e., expires) and resets with its initial, local SBC license according to the local License Key. This mechanism prevents misuse of SBC licenses allocated by the License Pool Manager Server.

When the device operates in High-Availability (HA) mode, the active and redundant devices are installed with the new SBC license from the License Pool Manager without affecting traffic. This is achieved by a hitless license upgrade mechanism, whereby the License Pool Manager first downloads the license to the redundant device, resets it, and then triggers the devices to perform an HA switchover. It then downloads the same license to the previously active device (now the redundant), resets it, and then triggers another HA switchover. Therefore, currently active calls are maintained throughout the process.

The device sends the following SNMP alarms relating to the allocation/de-allocation of SBC licenses by the License Pool Manager Server:

- acLicensePoolInfraAlarm (OID 1.3.6.1.4.1.5003.9.10.1.21.2.0.106)
- acLicensePoolApplicationAlarm (OID 1.3.6.1.4.1.5003.9.10.1.21.2.0.107)
- acLicensePoolOverAllocationAlarm (OID 1.3.6.1.4.1.5003.9.10.1.21.2.0.125)

For more information on the alarms, refer to the *SNMP Reference Guide*. 
44.4.6 Backing up the License Key

You can make a backup of the currently installed License Key. This may be useful, for example, if you later install a new License Key and you want to revert to the previous License Key.

➢ To back up the License Key:

1. Open the License Key page (see Viewing the License Key on page 670).
2. Click one of the following buttons:
   - ![Backup](backup-icon.png): Saves the License Key as a file to a folder on your computer. By default, the device names the file "license".
   - ![Clipboard](clipboard-icon.png): Copies the License Key as a string to your computer's clipboard. You can then paste the string into any application, for example, an e-mail message.

44.7 Viewing the Device's Product Key

The Product Key identifies a specific purchase of your device installation for the purpose of subsequent communication with AudioCodes (e.g., for support and software upgrades). The Product Key is provided in the order-confirmation email sent to you upon your product purchase and is used for activating your license through AudioCodes Software License Activation tool.

The Product Key is included in the License Key. Once the License Key is installed, you can view the Product Key in the following Web pages:

- **License Key page** (Setup menu > Administration tab > Maintenance folder > License Key). The Product Key is displayed in the read-only 'Product Key' field, as shown in the example below:
Device Information page (see Viewing Device Information on page 729).

If your License Key was purchased in an earlier version (for example, 7.0), the 'Product Key' field may appear empty. In such a scenario, request the Product Key from your AudioCodes sales representative. Once received, do the following:

1. Open the License Key page.
2. Locate the Product Key group:

   Figure 44-19: Empty Product Key Field

   License Key
   empty
   Product Key

3. Click "empty"; the following appears:

   Figure 44-20: Entering Product Key

   License Key
   

4. In the field, enter the Product Key, and then click Submit (or Cancel to discard your entry).
45 Software Upgrade Wizard

The Web interface’s Software Upgrade wizard lets you easily upgrade the device's software version (.cmp file). You can also use the wizard to load an ini file and Auxiliary files (e.g., CPT file). However, you can only use the wizard if you at least load a .cmp file. Once loaded, you can select other file types to load.

You can also use the wizard to upgrade devices in High Availability (HA) mode. You can choose between two optional HA upgrade methods:

- **System Reset Upgrade (non-Hitless):** Both active and redundant devices are upgraded simultaneously. Therefore, this method is traffic-affecting and terminates current calls during the upgrade process. The process is as follows:
  1. The active (current) device loads the .cmp file.
  2. The active device sends the .cmp file to the redundant device.
  3. Both active and redundant devices install and burn the file to flash memory with a reset. In other words, no HA switchover occurs.

- **Hitless Upgrade:** The devices are upgraded without disrupting traffic (i.e., current calls are maintained). The process is as follows:
  1. The active (current) device loads the .cmp file.
  2. The active device sends the .cmp file to the redundant device.
  3. The redundant device installs and burns the file to its flash memory with a reset. The redundant device now runs the new software version.
  4. An HA switchover occurs from active to redundant device. Therefore, current calls are maintained and now processed by the previously redundant device, which is now the active device.
  5. The previously active device (now in redundant mode) installs and burns the file to flash memory with a reset. Therefore, both devices now run the new software version.
  6. An HA switchover occurs from active device (i.e., the initial redundant device) to redundant device (i.e., the initial active device) to return the devices to their original HA state. Only the initial redundant device undergoes a reset to return to redundant state.
Note:

- You can obtain the latest software files from AudioCodes Web site at http://www.audiocodes.com/downloads.
- You can upgrade the device to the latest software version as specified in the installed License Key. If you attempt to upgrade the device to a version that is later than the one specified in the License Key, the device remains at the current software version. For more information, contact your AudioCodes sales representative.
- When upgrading Mediant VE SBC from Version 7.0 to any later version, please refer to the post-upgrade procedure described in Post-Upgrade from Version 7.0 Procedure on page 688.
- 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 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 693).

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>
  ```

- **HA devices:**
  
  - Hitless Software Upgrade:
    ```
    # copy firmware from <URL and file name>
    ```
  
  - Non-Hitless Software Upgrade:
    ```
    # copy firmware from <URL and file name> non-hitless
    ```

If you load the firmware file through CLI, when you initiate the copy command a message is displayed in the console showing the load progress. If other management users are connected to the device through CLI, the message also appears in their CLI sessions, preventing them from performing further actions on the device and disrupting the upload process. For more information, refer to the **CLI Reference Guide**.

- **To upgrade the device using the Software Upgrade wizard:**

  1. Make sure that you have installed a License Key that is compatible with the software version to be installed (see "License Key" on page 669).
  2. It is recommended to enable the Graceful Lock feature (see "Locking and Unlocking the Device" on page 651). The wizard resets the device at the end of the upgrade process, thereby causing current calls to be untimely terminated. To minimize traffic disruption, the Graceful Lock feature prevents the establishment of new calls.
  3. It is recommended to backup the device's configuration to your computer. If an upgrade failure occurs, you can restore your configuration by uploading the backup file to the device. For more information, see "Configuration File" on page 691.
4. Open the Software Upgrade wizard:
   - **Toolbar:** From the **Actions** drop-down menu, choose **Software Upgrade**.
   - **Navigation tree:** **Setup** menu > **Administration** tab > **Maintenance** folder > **Software Upgrade**.

   ![Figure 45-1: Starting Software Upgrade Wizard](image)

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

   ![Load a CMP file from your computer to the device.](image)

   ![Hitless Upgrade](image)
   ![System Reset Upgrade](image)

   **Note:**
   - The Hitless Upgrade and System Reset Upgrade options appear only if the device is configured for HA.
   - 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 and select the .cmp file.
7. Click **Load File**; the device begins to install the .cmp file and a progress bar displays the status of the loading process:

   **Figure 45-2: CMP File Loading Progress Bar**

When the file is loaded, a message is displayed to inform you.

8. If your device is in HA mode, select one of the following upgrade options:
   - Hitless Upgrade (default)
   - System Reset Upgrade

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

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

9. To load additional files, use the **Next** and **Back** buttons to navigate through the wizard to the desired file-load wizard page; otherwise, skip to the next step to load the .cmp file only.

   The wizard page for loading an ini file lets you do one of the following:
   - **Load a new ini file**:
     a. Click **Browse**, and then navigate to and select the new ini file.
     b. Click **Load File**; the device loads the ini file.
   - **Restore configuration to factory defaults**: Clear the 'Use existing configuration' check box.
• **Retain the existing configuration (default):** Select the 'Use existing configuration' check box.

**Figure 45-3: Load an INI File in the Software Upgrade Wizard**

![Image of Software Upgrade Wizard]

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

10. Click **Reset**; a progress bar is displayed, indicating the progress of saving the files to flash and device reset.

**Figure 45-4: Progress Bar Indicating Burning Files to Flash**

![Image of Progress Bar]
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 wizard displays the following, which lists the installed .cmp software version and other files that you may also have installed:

Figure 45-5: Software Upgrade Process Completed (Example)

![Software Upgrade Process Completed](image)

11. Click **End Process** to close the wizard; the Web Login page appears, allowing you to log in to your upgraded device.

### 45.1 Post-Upgrade from Version 7.0 Procedure

If you have upgraded the Mediant VE SBC from **Version 7.0** to any later version, you must add the following configuration parameter and settings to each virtual machine (VM) running the Mediant VE SBC to ensure high performance is maintained:

- **Parameter name:** `monitor_control.halt_desched`
- **Parameter value:** `FALSE`

The following procedure provides an example of how to do this using the VMware vSphere Client program.

➢ **To ensure performance when upgrading from Version 7.0:**

1. Upgrade the device's (virtual machine's) software from Version 7.0 (see "Software Upgrade Wizard" on page 683).
2. Access the host server running the virtual machines (using, for example, VMware vSphere Client).
3. Power off the virtual machine: In the list of virtual machines running on the host, right-click the name of the required virtual machine, and then from the shortcut menu, choose **Power Off**.
4. Add the parameter:
   a. Right-click the required virtual machine again, and then from the shortcut menu, choose **Edit Settings**.
b. Click the **Options** tab; the following appears:

**Figure 45-6: Adding Parameter to VM - Options Tab (Example)**

![Virtual Machine Properties window with Options tab selected](image)

- In the left pane, select **General** under the **Advanced** folder.
d. Click the **Configuration Parameters** button; the following dialog box appears:

![Configuration Parameters](image)

*Figure 45-7: Adding Parameter to VM - Configuration Parameters (Example)*

e. Click the **Add Row** button, and then add a parameter with the following settings:
   - Name: `monitor_control.halt_desched`
   - Value: `FALSE`

f. Click **OK**, and then click **OK** in all opened dialog boxes until you return to the main window listing all the virtual machines.

g. Power on the virtual machine: Right-click the name of the required virtual machine, and then from the shortcut menu, choose **Power On**.

5. Repeat the above steps for each virtual machine.
46 Configuration File

This section describes how to save the device's configuration to a file and how to load a configuration file to the device.

46.1 Saving Configuration to a File

The following procedure describes how to save the device's configuration to a file on your PC. The configuration file can be saved as an ini file or a CLI Script file. Saving a configuration file can serve as a backup of your configuration and if needed, you can later load the file to the device to restore your configuration settings.

Note:
• The saved configuration file includes only parameters whose values you have modified.
• To save the configuration as a CLI script file to a remote server (TFTP or HTTP/S):

  `# write-and-backup to <URL with file name>`

➢ To save configuration to a file on your PC through Web interface:

1. Open the Configuration File page:
   • Toolbar: From the Actions drop-down menu, choose Configuration File
   • Navigation tree: Setup menu > Administration tab > Maintenance folder > Configuration File
2. Depending on the file type that you want to save, click the following button:
   • ini file: Click Save INI File:

     Figure 46-1: Saving Configuration as INI File
     ![Save INI File]

   • CLI Script file: Click Save CLI Script File:

     Figure 46-2: Saving File as CLI Script
     ![Save CLI Script File]
3. Select the 'Save File' option, and then click OK.

46.2 Loading a Configuration File

The following procedure describes how to load a configuration file from a folder on your PC to the device. You can load any of the following configuration file types:

- ini file
- CLI Script file
- CLI Startup Script file
Warning:

- When loading an ini file as described in this section, parameters not included in the ini file are restored to default settings. If you want to keep the device's current configuration settings and also apply the settings specified in the ini file, load the file through the Auxiliary Files page, as described in Loading Auxiliary Files through Web Interface on page 656.
- Regarding device reset when loading a configuration file:
  - ini File: The device automatically resets for the settings to take effect.
  - CLI Script File: The device resets only if the file contains the reload command (on the last line). For more information on this command, refer to the CLI Reference Guide.
  - CLI Startup Script File: The device automatically resets twice for the settings to take effect.

➢ To load a configuration file to the device:

1. Open the Configuration File page:
   - Toolbar: From the Actions drop-down menu, choose Configuration File
   - Navigation tree: Setup menu > Administration tab > Maintenance folder > Configuration File
2. Depending on the file type that you want to load:
   - ini File:
     
     ![Figure 46-3: Loading ini File](image)

     a. Click Browse, and then navigate to and select the file.
     b. Click Load INI File:
   - CLI Script File:
     
     ![Figure 46-4: Loading CLI Script File](image)

     a. Click Browse, and then navigate to and select the file.
     b. Click Load CLI Script File.
   - CLI Startup Script File:
     
     ![Figure 46-5: Loading CLI Startup Script File](image)

     a. Click Browse, and then navigate to and select the file.
     b. Click Load CLI Startup Script:

If the device resets (see note at the beginning of this section), you are prompted to log in again to the Web interface.
47 Automatic Provisioning

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:

<table>
<thead>
<tr>
<th>BootP / TFTP</th>
<th>DHCP</th>
<th>Automatic Update Methods</th>
<th>SNMP (EMS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>67</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>FTP</td>
<td>NFS</td>
<td></td>
</tr>
</tbody>
</table>

47.1.1 DHCP-based Provisioning

A third-party DHCP server can be configured to automatically provide each device, acting as a DHCP client, with a temporary IP address so that individual MAC addresses are not required. The DHCP server can provide additional networking parameters such as subnet mask, default gateway, primary and secondary DNS server, and two SIP server addresses. These network parameters have a time limit, after which the device must 'renew' its lease from the DHCP server.

The device can use a host name in the DHCP request. The host name is set to acl_6nnnn, where 6nnnn 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.

Note:
- When using DHCP to acquire an IP address, the IP Interfaces table, VLANs and other advanced configuration options are disabled.
- For additional DHCP parameters, see "DHCP Parameters" on page 853.

➢ To enable the device as a DHCP client:

1. Open the Network Settings page (Setup menu > IP Network tab > Advanced folder > Network Settings).
2. From the 'Enable DHCP' drop-down list, select Enable.

Figure 47-1: Enabling DHCP Client Functionality

Enable DHCP

Enable

3. Click Apply.
4. To activate the DHCP process, reset the device.
The following shows an example of a configuration file for a Linux DHCP server (dhcpd.conf). The devices are allocated temporary IP addresses in the range 10.31.4.53 to 10.31.4.75. TFTP is assumed to be on the same computer as the DHCP server (alternatively, the "next-server" directive may be used).

```
dDNS-update-style ad-hoc;
default-lease-time 60;
max-lease-time 60;
class "gateways" {  
    match if (substring(hardware, 1, 3) = 00:90:8f);
}
subnet 10.31.0.0 netmask 255.255.0.0 {  
    pool {  
        allow members of "audiocodes";
        range 10.31.4.53 10.31.4.75;
        filename "SIP_F6.60A.217.003.cmp –fb;device.ini";
        option routers 10.31.0.1;
        option subnet-mask 255.255.0.0;
    }
}
```

**Note:**

- 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 693 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.:


For more information on HTTP-based provisioning, see "HTTP/S-Based Provisioning using the Automatic Update Feature" on page 695.

### 47.1.3 FTP-based Provisioning

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 they are changed on the server).

The only difference between FTP-based provisioning and those described in "HTTP-based Provisioning" on page 694 is that the protocol in the URL is "ftp" (instead of "http").

### 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.
Warning: If you use the IniFileURL parameter for the Automatic Update feature, do not use the Web interface to configure the device. If you do configure the device through the Web interface and save (burn) the new settings to the device's flash memory, the IniFileURL parameter is automatically set to 0 and Automatic Updates is consequently disabled. To enable Automatic Updates again, you need to re-load the ini file (using the Web interface or BootP) with the correct IniFileURL settings. As a safeguard to an unintended save-to-flash when resetting the device, if the device is configured for Automatic Updates, the 'Burn To FLASH' field under the Reset Configuration group in the Web interface's Maintenance Actions page is automatically set to No by default.

Note:
- For a description of all the Automatic Update parameters, see "Automatic Update Parameters" on page 844 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:
  - ini File: Contains only ini file parameters and configures all the device's functionalities.
  - CLI Script File: Contains only CLI commands and configures all the device's functionalities (except commands such as show, debug or copy). The file updates the device's configuration only according to the configuration settings in the file. 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
    ```
    To configure the URL of the server where the file is located, use the AUPDClCoiniScriptURL ini file parameter or CLI command, configure system > automatic-update > cli-script <URL>.
  - Startup Script File: Contains only CLI commands and configures all the device's functionalities (except commands such as show, debug or copy). The file updates the device's configuration according to the configuration settings in the file and sets all other parameters that are not included in the file to factory defaults. The file causes two device resets to apply the settings. Therefore, the 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>. 
47.2.2 File Location for Automatic Update

The files for updating the device can be stored on any standard Web (HTTP/S), TFTP, or FTP, server. The files can be loaded periodically to the device using HTTP/S, TFTP, or FTP. This mechanism can be used even when the device is installed behind NAT and firewalls. The Automatic Update feature is done per file and configured by specifying the file name and URL address of the provisioning server where the file is located. For a description of the parameters used to configure URLs per file, see "Automatic Update Parameters" on page 844.

Below are examples for configuring the file names and their URLs for Automatic Update:

- ini File:
  
  IniFileURL = 'http://www.corp.com/configuration.ini'
  CptFileURL = 'http://www.corp.com/call_progress.dat'
  AutoCmpFileUrl = 'http://www.corp.com/SIP_F7.00A.008.cmp'
  FeatureKeyURL = 'https://www.company.com/License_Key.txt'

- 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)# feature-key http://www.company.com/License_Key.txt
  (automatic-update)# auto-firmware http://www.company.com/SIP_F7.00A.008.cmp

Note: For configuration files, the file name in the URL can automatically contain the device's MAC address for enabling the device to download a file unique to the device. For more information, see "MAC Address Placeholder in Configuration File Name" on page 697.

47.2.3 MAC Address Placeholder in Configuration File Name

You can configure the file name of the configuration file in the URL to automatically include the MAC address of the device. As described in "File Location for Automatic Update" on page 697, the file name is included in the configured URL of the provisioning server where the file is located.

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

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

IniFileURL = 'https://www.company.com/config_<MAC>.ini'
(automatic-update)# cli-script
https://company.com/files/cli_script_<MAC>.txt
(automatic-update)# startup-script
https://company.com/files/startup_<MAC>.txt
47.2.4 File Template for Automatic Provisioning

To facilitate automatic provisioning setup, you can use a single template to define the files to download during automatic provisioning. The template uses special keywords to denote the different file types to download and in the URL address of the provisioning server it uses a placeholder for the file names which is replaced by hardcoded file names and extensions according to file type, as described in more detail below.

**Note:**
- Unlike the parameters that define specific URLs for Auxiliary files (e.g., CptFileURL), the file template feature always retains the URLs after each automatic update process. Therefore, with the file template the device always attempts to download the files upon each automatic update process.
- If you configure a parameter used to define a URL for a specific file (e.g., CptFileURL), the settings of the TemplateUrl parameter is ignored for the specific file type (e.g., CPT file).
- Additional placeholders can be used in the file name in the URL, for example, <MAC> for MAC address (see "MAC Address Placeholder in Configuration File Name" on page 697).

➢ **To use a file template for automatic provisioning:**

1. Define the file types to download by the file template, using the AupdFilesList parameter. Use the keywords listed in the table below to specify each file type. For example, to specify ini, License Key, and CPT files:
   - ini File:
     ```plaintext
     AupdFilesList = 'ini', 'fk', 'cpt'
     ```
   - CLI:
     ```plaintext
     # configure system
     (config-system)# automatic update
     (automatic-update)# template-files-list ini,fk,cpt
     ```

2. Define the URL address of the provisioning server on which the files (specified in Step 1) are located for download, using the TemplateUrl parameter. When you configure the URL, you must include the file type placeholder, "<FILE>", which represents the file name. For each file type specified in Step 1, the device sends an HTTP request to the server, where the placeholder in the URL is replaced with the filename and extension, as listed in the below table. For example, if you configure the AupdFilesList parameter as in Step 1 and the TemplateUrl parameter to:
   - ini File:
47. Automatic Provisioning

TemplateUrl = 'http://10.8.8.20/Site1_<FILE>'

- CLI:
  
  ```
  # configure system
  (config-system)# automatic update
  (automatic-update)# template-url http://10.8.8.20/Site1_<FILE>
  ```

The device sends HTTP requests to the following URLs:

- http://10.8.8.20/Site1_device.ini
- http://10.8.8.20/Site1_fk.ini
- http://10.8.8.20/Site1_cpt.data

3. Place the files to download on the provisioning server. Make sure that their file names and extensions are based on the hardcoded string values specific to the file type for the `<FILE>` placeholder (e.g., "Site1_device.ini" for the ini file), as shown in the table below.

<table>
<thead>
<tr>
<th>File Type</th>
<th>Keywords for Template File</th>
<th>Value Replacing <code>&lt;FILE&gt;</code> Placeholder</th>
</tr>
</thead>
<tbody>
<tr>
<td>ini file</td>
<td>ini</td>
<td>device.ini</td>
</tr>
<tr>
<td>CLI Script file</td>
<td>cli</td>
<td>cliScript.txt</td>
</tr>
<tr>
<td>CLI Startup Script file</td>
<td>clis</td>
<td>cliStartupScript.txt</td>
</tr>
<tr>
<td>CMP file based on timestamp</td>
<td>acmp</td>
<td>autoFirmware.cmp</td>
</tr>
<tr>
<td>User Info file</td>
<td>usrinf</td>
<td>userInfo.txt</td>
</tr>
<tr>
<td>CMP file</td>
<td>cmp</td>
<td>firmware.cmp</td>
</tr>
<tr>
<td>License Key file</td>
<td>fk</td>
<td>fk.ini</td>
</tr>
<tr>
<td>Call Progress Tone (CPT) file</td>
<td>cpt</td>
<td>cpt.dat</td>
</tr>
<tr>
<td>Prerecorded Tones (PRT) file</td>
<td>prt</td>
<td>prt.dat</td>
</tr>
<tr>
<td>Dial Plan file</td>
<td>dpln</td>
<td>dialPlan.dat</td>
</tr>
<tr>
<td>Answering Machine Detection (AMD) file</td>
<td>amd</td>
<td>amd.dat</td>
</tr>
<tr>
<td>SSL/TLS Private Key file</td>
<td>sslp</td>
<td>pkey.pem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>pkey&lt;ID&gt;.pem (for multi-certificate system)</td>
</tr>
<tr>
<td>SSL/TLS Root Certificate file</td>
<td>sslr</td>
<td>root.pem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>root&lt;ID&gt;.pem (for multi-certificate system)</td>
</tr>
<tr>
<td>SSL/TLS Certificate file</td>
<td>sslc</td>
<td>cert.pem</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cert&lt;ID&gt;.pem (for multi-certificate system)</td>
</tr>
</tbody>
</table>

47.2.5 Triggers for Automatic Update

The Automatic Update feature can be triggered by the following:

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

a. Open the Gateway Advanced Settings page (Setup menu > Signaling & Media tab > Gateway folder > Gateway Advanced Settings).

b. From the 'Remote Management by Notify' (EnableSIPRemoteReset) drop-down list, select Enable:

   Figure 47-2: Resetting Device by SIP NOTIFY

   ![Image](image.png)

   Remote Management by SIP Notify

   Enable

   c. Click Apply.

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

47.2.6 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:
    ```ini
    AutoCmpFileUrl = 'https://<username>:<password>@<IP address or domain name>/<file name>'
    ```

  - CLI:
    ```bash
    # 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
47. Automatic Provisioning

authentication username and password are configured using the AUPDDigestUsername and AUPDDigestPassword parameters, respectively.

47.2.7 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 700). If authentication succeeds, Step 2 occurs.

2. The device establishes an HTTP/S connection with the URL host (provisioning server). If the connection is HTTPS, the device verifies the certificate of the provisioning server, and presents its own certificate if requested by the server.

3. The device queries the provisioning server for the requested file by sending an HTTP Get request. This request contains the HTTP User-Agent Header, which identifies the device to the provisioning server. By default, the header includes the device's model name, MAC address, and currently installed software and configuration versions. Based on its own dynamic applications for logic decision making, the provisioning server uses this information to check if it has relevant files available for the device and determines which files must be downloaded (working in conjunction with the HTTP If-Modified-Since header, described further on in this section).

You can configure the information sent in the User-Agent header, using the AupdHttpUserAgent parameter or CLI command, configure system > http-user-agent. The information can include any user-defined string or the following supported string variable tags (case-sensitive):

- `<NAME>`: product name, according to the installed License Key
- `<MAC>`: device's MAC address
- `<VER>`: software version currently installed on the device, e.g., "7.00.200.001"
- `<CONF>`: configuration version, as configured by the ini file parameter, INIFileVersion or CLI command, configuration-version

The device automatically populates these tag variables with actual values in the sent header. By default, the device sends the following in the User-Agent header:

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

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

```
User-Agent: MyWorld-Mediant;7.00.200.001(00908F1DD0D3)
```

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

4. If the provisioning server has relevant files available for the device, the following occurs, depending on file type and configuration:
• **File Download upon each Automatic Update process:** This is applicable to software (.cmp) and configuration 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:

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

If the file on the provisioning server was unchanged (not 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 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 704.

**Note:**

• When this method is used, there is typically no need for the provisioning server to check the device’s current firmware version using the HTTP-User-Agent header.

• The Automatic Update feature assumes that the Web server conforms to the HTTP standard. If the Web server ignores the If-Modified-Since header or doesn’t provide the current date and time during the HTTP 200 OK response, the device may reset itself repeatedly. To overcome this problem, modify the update frequency, using the ini file parameter AutoUpdateFrequency or CLI command configure system > automatic update > update-frequency.

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

**Auxiliary Files:**

- **ini:**
  ```plaintext
  CptFileURL = 'https://www.company.com/call_progress.dat'
  FeatureKeyURL = 'https://www.company.com/License_Key.txt'
  ```

- **CLI:**
  ```plaintext
  (config-system)# automatic-update
  (automatic-update)# call-progress-tones
  ```
Software (.cmp) File:

- ini:

  CmpFileUrl = 'https://www.company.com/device/v.7.20A.000.038.cmp'

- CLI:

  (config-system)# automatic-update
  (automatic-update)# firmware
  https://www.company.com/device/v.7.20A.000.038.cmp

**Note:**

- 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 certificate files, it is recommended to use HTTPS with mutual authentication for secure transfer of the SSL Private Key.

- After the device downloads the License Key file (FeatureKeyURL), it checks that the serial number in the file ("S/N <serial number>") is the same as that of the device. If the serial number is the same and the license key is different to the one currently installed on the device, it applies the new License Key. For devices in HA mode, the License Key is applied to both active and redundant units.

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

### 47.2.8 File Download Sequence

Whenever the Automatic Update feature is triggered (see "Triggers for Automatic Update" on page 699), the device attempts to download each file from the configured URLs, in the following order:

1. ini file
2. CLI Script file
3. Startup Script file
4. Periodic software file (.cmp) download
5. One-time software file (.cmp) download
6. 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:

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

Note:

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

47.2.9 Cyclic Redundancy Check on Downloaded Configuration Files

You can enable the device to perform cyclic redundancy checks (CRC) on downloaded configuration files 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
Therefore, the CRC feature can be used to prevent the device from installing such files.

For enabling CRC, use the ini file parameter AUPDCheckIfIniChanged or CLI command, configure system > automatic-update > crc-check regular. By default, CRC is disabled. For more information on the parameter, see "Automatic Update Parameters" on page 844.

47.2.10 Automatic Update Configuration Examples

This section provides a few examples on configuring the Automatic Update feature.

47.2.10.1 Automatic Update for Single Device

This simple example describes how to configure the Automatic Update feature for updating a single device. In this example, the device queries the provisioning server for software, configuration and Auxiliary files every 24 hours.

➢ To set up Automatic Provisioning for single device (example):

1. Set up an HTTP Web server (e.g., http://www.company.com) and place all the required configuration files on this server.

2. Configure the device with the IP address of the DNS server for resolving the domain name (e.g., http://www.company.com) that is used in the URL of the provisioning server. You configure this in the IP Interfaces table:

   • ini File:
     ```ini
     [ 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 network
     (config-network)# interface network-if 0
     (network-if-0)# primary-dns 80.179.52.100
     ```

3. Configure the device with the following Automatic Update settings:

   a. Automatic Update is done every 24 hours (1440 minutes):

      ◆ ini File:
      ```ini
      AutoUpdateFrequency = 1440
      ```

      ◆ CLI:
      ```
      # configure system
      (config-system)# automatic update
      (automatic-update)# update-frequency 1440
      ```

   b. Automatic Update of software file (.cmp):

      ◆ ini File:
      ```ini
      AutoCmpFileUrl = 'https://www.company.com/sw.cmp'
      ```

      ◆ CLI:
# configure system
(config-system)# automatic update
(automatic-update)# auto-firmware 'http://www.company.com/sw.cmp'

c. Automatic Update of Call Progress Tone file:
  ♦ ini File:
    CptFileURL = 'https://www.company.com/call_progress.dat'
  ♦ CLI:
    # configure system
    (config-system)# automatic update
    (automatic-update)# call-progress-tones
    'http://www.company.com/call_progress.dat'

d. Automatic Update of ini configuration file:
  ♦ ini File:
    IniFileURL = 'https://www.company.com/config.ini'
  ♦ CLI:
    # configure system
    (config-system)# automatic update
    (automatic-update)# voice-configuration
    'http://www.company.com/config.ini'

e. Enable Cyclical Redundancy Check (CRC) on downloaded ini file:
  ♦ ini File:
    AUPDCheckIfIniChanged = 1
  ♦ CLI:
    # configure system
    (config-system)# automatic update
    (automatic-update)# crc-check regular

4. Power down and then power up the device.

47.2.10.2 Automatic Update from Remote Servers

This example describes how to configure the Automatic Update feature where files are stored and downloaded from different file server types. The example scenario includes the following:

- FTPS server at ftpserver.corp.com for storing the License Key file. The login credentials to the server are username "root" and password "wheel".
- HTTP server at www.company.com for storing the configuration file.
- DNS server at 80.179.52.100 for resolving the domain names of the provisioning servers (FTPS and HTTP).

➢ To set up Automatic Provisioning for files stored on different server types (example):

1. License Key file:
   a. Set up an FTPS server and copy the License Key file to the server.
   b. Configure the device with the URL path of the License Key file:
      ♦ ini File:
        FeatureKeyURL = 'ftps://root:wheel@ftpserver.corp.com/license_key.txt'
      ♦ CLI:
        # configure system
        (config-system)# automatic update
2. Software (.cmp) and ini files:
   a. Set up an HTTP Web server and copy the .cmp and configuration files to the server.
   b. Configure the device with the URL paths of the .cmp and ini files:
      - ini File:
        ```
        AutoCmpFileUrl = 'http://www.company.com/device/sw.cmp'
        IniFileURL = 'http://www.company.com/device/inifile.ini'
        ```
      - CLI:
        ```
        # configure system
        (config-system)# automatic update
        (automatic-update)# auto-firmware 'http://www.company.com/sw.cmp'
        (automatic-update)# startup-script
        https://company.com/files/startup_script.txt
        ```

3. Configure the device with the IP address of the DNS server for resolving the domain names of the FTPS and HTTP servers:

4. 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
     ```

47.2.10.3 Automatic Update for Mass Deployment

This example describes how to configure the Automatic Update feature for updating multiple devices (i.e., mass deployment) using an HTTP provisioning server. In this example, all the devices are configured to download the same "master" configuration file. This file serves as the configuration template and instructs the devices which files to download and how often to perform the Automatic Update process. In addition, the master file also instructs each device to download an ini configuration file whose file name contains the MAC address of the device.

The example scenario is as follows:

- All devices download a "master" configuration file that contains the following:
  - Common configuration shared by all device's.
Specific configuration that instructs each device to download a specific configuration file based on the device's MAC address, using the special string "<MAC>" in the URL, as described in "MAC Address Placeholder in Configuration File Name" on page 697.

Device queries the provisioning server daily at 24:00 (midnight) for software, configuration and Auxiliary files.

HTTP-based provisioning server at www.company.com for storing the files.

DNS server at 80.179.52.100 for resolving the domain name of the provisioning server.

To set up automatic provisioning for mass provisioning (example):

1. Create a "master" configuration file template named "master_configuration.ini" or "master_startup.txt" with the following settings:

   Common configuration for all devices:
   - ini file:
     AutoUpdatePredefinedTime = '24:00'
     CptFileURL = 'https://www.company.com/call_progress.dat'
     AutoCmpFileUrl = 'https://www.company.com/sw.cmp'
   - CLI:
     # configure system
     (config-system)# automatic update
     (automatic-update)# update-frequency 24:00
     (automatic-update)# call-progress-tones
     https://www.company.com/call_progress.dat
     (automatic-update)# auto-firmware https://www.company.com/sw.cmp

   Configuration per device based on MAC address:
   - ini file:
     IniFileURL = 'http://www.company.com/config_<MAC>.ini'
   - CLI:
     # configure system
     (config-system)# automatic update
     (automatic-update)# cli-script
     http://company.com/files/cli_script_<MAC>.txt
     (automatic-update)# voice-configuration
     http://www.company.com/config_<MAC>.ini

2. Copy the master configuration file that you created in Step 1 as well as the CPT and .cmp files to the HTTP-based provisioning server.

3. Configure each device with the following:
   a. URL of the master configuration file:
      - ini File:
        IniFileURL = 'http://www.company.com/master_configuration.ini'
      - CLI:
        # configure system
        (config-system)# automatic update
        (automatic-update)# voice-configuration
        http://www.company.com/master_configuration.ini
        (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 IP Interfaces 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";

CLI:

# configure network
(config-network)# interface network-if 0
(network-if-0)# primary-dns 80.179.52.100

4. Power down and then power up the device.
This page is intentionally left blank.
48 SBC Configuration Wizard

The SBC Configuration Wizard provides you with a quick-and-easy method for initial configuration of your device. The SBC Configuration Wizard guides you through a sequence of pages, assisting you in defining your device. Once the wizard is complete, your device is sufficiently configured to successfully process and route calls in your deployment.

The SBC Configuration Wizard is based on partially and fully tested interoperability setups between the device and a wide range of vendors, including SIP trunking providers, IP PBXs, and contact centers. Once you have selected the involved vendors and defined basic settings in the SBC Configuration Wizard, it then generates a configuration file based on the matching interoperability configuration template. Alternatively, instead of basing your configuration on specific vendors, you can use the SBC Configuration Wizard to generate a configuration file based on a generic template for commonly used setups. In such cases, you may later need to manually fine-tune your configuration to suit your setup needs.

The SBC Configuration Wizard can automatically load the generated configuration (with a reset) to the device, or you can simply save the generated configuration file to a folder on your PC and then load the file to the device at a later stage.

The generated configuration is a good starting point to enable the successful establishment of basic calls. For complete device configuration, you may need to manually configure additional functionality. For example, you may need to configure security settings (e.g., firewalls and IDS) to ensure that the device is protected from malicious activity and DoS attacks.

For AudioCodes’ full interoperability list, visit AudioCodes website at http://www.audiocodes.com/sbc-interoperability-list.

Note:

- When the SBC Configuration Wizard applies the configuration template to the device, all parameters configured by the SBC Configuration Wizard overwrite the device's existing configuration of those parameter. Parameters not configured by the SBC Configuration Wizard are restored to factory defaults, except basic device settings such as management users (Web and CLI). Some of these basic settings also appear in the SBC Configuration Wizard and their fields are automatically populated with their current settings; if you do modify them in the SBC Configuration Wizard, their new settings are used.

- On some wizard pages, the availability of certain fields depends on the selected application.
48.1 Starting the SBC Configuration Wizard

The following procedure describes how to start the SBC Configuration Wizard. Throughout the wizard, you can get help on the current wizard page, by clicking the ❓ icon, located on the top-right of the page.

➢ To start the SBC Configuration Wizard:

1. Access the SBC Configuration Wizard welcome page:
   - **Toolbar**: Click Actions, and then from the drop-down list, choose Configuration Wizard.
   - **Navigation Tree**: Setup menu > Administration tab > Maintenance folder > Configuration Wizard.

   ![Figure 48-1: SBC Configuration Wizard - Welcome Page](image)

2. If desired, the SBC Configuration Wizard allows you to share usage statistics with AudioCodes in order to help us improve our software. To agree, select the 'Report usage statistics' check box, and then fill in the subsequent fields.

3. The version of the template pack currently installed on the device is displayed in the 'Template pack version' field. The template pack contains the interoperability configuration templates available on the SBC Configuration Wizard. If the template pack is not the latest version (as displayed in the 'Remote Template pack version' field), you can update it by clicking the Update from remote server button. Alternatively, if you have received a template pack file from AudioCodes sales representative, you can install it on the device using the Auxiliary Files page (see Loading Auxiliary Files on page 655).

4. Click **Next**; the General Setup page appears (see General Setup Page on page 713).
48.2 General Setup Page

The General Setup wizard page defines the network topology of the device, which includes the application (e.g., SIP trunk) and the involved third-party vendors, configuration template based on the selected vendor interoperability, and physical network (ports). The wizard displays an illustration of the basic architecture according to your chosen setup.

To define the general setup:

1. Select one of the following 'Application' options:
   - **SIP Trunk (IP-PBX with SIP Trunk)**: The device connects the Enterprise IP PBX with the Internet Telephony Service Provider (ITSP) or SIP Trunk Provider. The IP PBX resides on the Enterprise LAN, while the ITSP resides on the WAN. Only calls between the IP PBX and ITSP traverse the device. Calls between Enterprise phones, users and certain SIP messages (REGISTER, SUBSCRIBE and NOTIFY) are managed locally by the IP PBX and do not traverse the device.
   - **SIP Normalization (two IP-PBX’s)**: The device performs SIP "normalization" of traffic between two IP PBXs. The first IP PBX resides on the Enterprise LAN (co-located with the device) and the second IP PBX resides on the WAN (or at another branch site). Only calls between the IP PBXs traverse the device. Calls between the phones and users of the same IP PBX, and certain SIP messages (REGISTER, SUBSCRIBE and NOTIFY) are managed locally by each IP PBX and do not traverse the device.
   - **Hosted IP-PBX (IP-PBX with Users)**: The device connects phones (users) with the "hosted" IP PBX. The users reside on the Enterprise LAN (co-located with the device) and the IP PBX resides on the WAN (or at the datacenter). All traffic between users and IP PBX traverse the device, including SIP REGISTER, SUBSCRIBE and NOTIFY messages.
   - **Remote Users (IP-PBX with Remote Users)**: The device connects remote phones (users) with the "local" IP PBX. The IP PBX resides on the Enterprise LAN (co-located with the device) and the remote users reside on the WAN (or at the datacenter). All traffic between users and IP PBX traverse the device, including SIP REGISTER, SUBSCRIBE and NOTIFY messages.

2. If you selected the **SIP Trunk** application in Step 1, do the following:
   a. From the 'IP-PBX' drop-down list, select the IP PBX model. If the model is not listed, select **Generic IP-PBX**.
b. From the 'SIP-Trunk' drop-down list, select the SIP trunk provider. If the provider is not listed, select **Generic SIP Trunk**.

c. To generate a configuration template based on the individual properties of the selected IP PBX and SIP Trunk, instead of using the existing template for the specific combination, select the 'Override template' check box.

3. If you selected any application except **SIP Trunk** in Step 1, from the 'Template' drop-down list, select the interoperability configuration template.

4. From the 'Network Setup' drop-down list, select the physical network topology:

- **Two ports: LAN and WAN**: The device connects to the network through two separate physical network links (interfaces). The first interface ("LAN") is connected to the Enterprise LAN (typically, a switch) and has a private IP address. The second interface ("WAN") is connected to the DMZ port of the Enterprise router and has a public (globally routable) IP address. Each link may be accompanied with a backup link for Ethernet link redundancy.

- **One port: LAN**: The device connects to the Enterprise LAN (typically, a switch) through a single physical network link (interface). The interface ("LAN") has a private IP address. You must enable port forwarding on the Enterprise router to forward all VoIP traffic from the ITSP (located on the WAN) to the device. The exact port forwarding configuration is shown on the Conclusion page and consists of the device's address, SIP port (e.g. 5060) and a media port range (e.g. 6000-6999).

- **One port: WAN**: The device connects to the DMZ port of the Enterprise router through a single physical network link (interface). The interface ("WAN") has a public (globally routable) IP address. You must enable port forwarding on the Enterprise router to forward all VoIP traffic from the device to the IP PBX (located on the LAN). The exact port forwarding configuration is shown on the Conclusion page and consists of the IP PBX address, SIP port (e.g. 5060) and a media port range (e.g. 6000-6999).

- **One port: LAN only**: The device connects to the Enterprise LAN (typically, a switch) through a single physical network link (interface). All SIP entities (IP PBX and users) connect to the same LAN. Note that this option is applicable to all applications (see Step 1), except **SIP Trunk**.

5. Click **Next**; the System page appears (see System Page on page 715).
48.3 System Page

The System wizard page configures the device’s basic system settings, including management interface protocol, date and time, and syslog.

Figure 48-3: SBC Configuration Wizard - System Page

To configure system settings:

1. Configure the protocol for accessing the management interfaces:
   - 'Web Interface': Select the type of Web interface protocol (HTTP or HTTPS).
   - 'CLI Interface': Select the type of CLI protocol (Disabled, Telnet, SSH).

2. (Optional) Enable Syslog by selecting the 'Enable Syslog' check box, and then in the 'Syslog IP' field, configuring the Syslog server's IP address. Note that if you enable Syslog, the device generates verbose logs (debug level 5), which adversely affects performance. For more information on Syslog, see Configuring Syslog on page 792.

3. Configure the time and date:
   - 'Time Zone': Select the GMT time zone in which the device is located.
   - 'Primary NTP Server' and 'Secondary NTP Server': Configure the IP address (or hostname) of the primary (and optionally, secondary) NTP server in your network. The NTP server synchronizes the time of the device. This is mandatory when the IP PBX or ITSP uses a TLS connection, as correct time is required for certificate validation.

   For more information on configuring date and time, see Date and Time on page 121.

4. (Optional) Configure a local DNS table, which allows the device to resolve domain names (hostnames) using a locally defined address resolution table. This may be needed when your setup lacks a DNS server and the IP PBX or ITSP require the use of hostnames instead of IP addresses.

   a. Select the 'Apply local DNS' check box, and then configure the following parameters:
- 'Domain Name': Domain name to resolve into an IP address.
- 'First IP address': IP address of the domain name.
- 'Secondary IP address': Second IP address of the domain name (optional).

For more information on configuring the device's DNS table, see Configuring the Internal DNS Table on page 159.

5. Click **Next**; the Interfaces page appears (see Interfaces Page on page 716).

### 48.4 Interfaces Page

The Interfaces wizard page configures the device's LAN and WAN settings.

![SBC Configuration Wizard - Interfaces Page](image)

**Figure 48-4: SBC Configuration Wizard - Interfaces Page**

- **To configure LAN and WAN interface settings:**
  1. From the 'Physical Port' drop-down list, select the Ethernet Group containing the required physical Ethernet port for connecting the device to the 1) Enterprise LAN (typically, a switch) or 2) DMZ port of the Enterprise router for WAN. You may configure two physical ports for Ethernet link redundancy.
  2. Select the VLAN Tagging check box, and then in the 'VLAN ID' field, configure the VLAN ID for the interface. For untagged traffic, clear the check box.
  3. In the 'IP Address' field, configure the device's IP address on the interface.
     - For LAN interfaces: The IP address must be part of the Enterprise LAN and therefore, is typically a private IP address. The address is used for communicating with the IP PBX and/or users that reside on the LAN, as well as for management (OAMP) traffic.
     - For WAN interfaces: The IP address must be a public (globally routable) address and is used for communicating with the ITSP and/or IP PBX that resides on the WAN. If the WAN is the only interface, it is also used for management (OAMP) traffic.
  4. In the 'Subnet Mask' field, configure the subnet mask of the interface.
  5. In the 'Default Gateway' field, configure the default gateway of the interface.
  6. If the device is connected through an Enterprise router that performs NAT, then in the
'NAT Public IP' address, configure the public IP address (of the Enterprise router) used by the device to communicate with the ITSP (for the SIP Trunk application) or IP PBX (for the Hosted IP-PBX application).

7. In the 'Primary DNS Server' (and optionally, 'Secondary DNS Server') field, configure your primary (and optionally, secondary) DNS server in the network. This is mandatory if you use a hostname (FQDN) for ITSP (WAN only) and IP PBX addresses.

8. From the 'OAM Interface' drop-down list, select the device’s interface for management traffic:
   - **LAN**: Management traffic is carried over the regular LAN interface, as defined above.
   - **WAN**: Management traffic is carried over the WAN interface, as defined above.
   - **Additional**: Configure a different interface for management traffic.

9. Click **Next**; the IP-PBX page appears (see IP-PBX Page on page 717).

### 48.5 IP-PBX Page

The IP-PBX wizard page configures the IP PBX settings.

![IP-PBX Page](image)

Figure 48-5: SBC Configuration Wizard - IP-PBX Page

The following fields are read-only:

- **'Network Type':** Displays the IP network interface for communicating with the IP PBX.
- **'NAT Public IP':** Displays the public IP address (of the Enterprise router) for communicating with the IP PBX. The field is applicable only when the device is connected to a router that performs NAT.

➢ **To configure IP PBX settings:**

1. Under the IP-PBX group, configure the following:
- 'Address': Configure the IP address (or hostname) of the IP PBX. Note that for the One port: WAN network topology, when the device is assigned a public IP address, you must use the public IP address (of the Enterprise router) instead of the private address of the IP PBX, and configure the Enterprise router to forward VoIP traffic from the device to the IP PBX.
- 'Backup Address': (Optional) Configure the backup IP address (or hostname) of the IP PBX.
- 'SIP Domain': Configure the SIP domain name used for communication with the IP PBX. The domain name is used in the following SIP message headers:
  - Outbound calls: Request-URI and To headers
  - Inbound calls: From header
- 'Keep Alive': Enable the periodic keep-alive check for multiple IP PBX addresses.

2. Under the Media Ports (Realm) group, configure the media protocol type and ports used by the device for communicating with the IP PBX:
- 'Media Protocol': Configure the media protocol type (RTP or SRTP).
- 'Base Port' Configure the first media port in the port range.
- 'Number Of Sessions': Configure the number of required media sessions. For more information on media port ranges and number of sessions, see Configuring RTP Base UDP Port on page 195.

3. Under the SIP Interface group, configure SIP ports and transport type for communicating with the IP PBX:
- 'Transport Type': Configure the SIP transport type.
- 'Destination Port': Configure the SIP port used by the IP PBX.
- 'Listening Port': Configure the SIP port used by the device when communicating with the IP PBX.

4. Click Next; the SIP Trunk page appears (SIP Trunk Page on page 719).
48.6 SIP Trunk Page

The SIP Trunk wizard page configures the SIP Trunk settings.

Figure 48-6: SBC Configuration Wizard - SIP Trunk Page

The following fields are read-only:

- 'Network Type': Displays the IP network interface for communicating with the SIP Trunk.
- 'NAT Public IP': Displays the public IP address (of the Enterprise router) for communicating with the SIP Trunk. The field is applicable only when the device is connected to a router that performs NAT. Note that the Enterprise router must be configured to "port forward" all VoIP traffic from the SIP Trunk (located on the WAN) to the device. The exact port forwarding configuration is displayed on the Conclusion page and consists of the device's address, SIP listening port (e.g. 5060) and a range of media ports defined below (e.g. 6000-6999).

➢ To configure SIP Trunk settings:

1. Under the SIP Trunk group, configure the following:
   - 'Address': Configures the IP address or hostname of the SIP Trunk.
   - 'Backup Address': (Optional) Configures the backup IP address or hostname of the SIP Trunk.
• 'SIP Domain': Configures the SIP domain name for communicating with the SIP Trunk. The domain name is used in the following SIP message headers:
  ♦ Outbound calls: Request-URI and To headers
  ♦ Inbound calls: From header
• 'Keep Alive': Enables the periodic keep-alive check of multiple SIP Trunk addresses.

2. Under the SIP Interface group, configure the SIP ports and transport type for communicating with the SIP Trunk:
  • 'Transport Type': Configure the SIP transport type.
  • 'Destination Port': Configure the SIP port used by the SIP Trunk.
  • 'Listening Port': Configure the SIP port used by the device for communicating with the SIP Trunk. Note that for the **One port: WAN** network topology, the device must use different Listening Ports when communicating with the IP PBX and SIP Trunk.

3. Under the Media Ports (Realm) group, configure the media protocol type and ports used by the device for communicating with the IP PBX:
  • 'Media Protocol': Configure the media protocol type.
  • 'Base Port': Configure the first media port.
  • 'Number Of Sessions': Configure the number of required media sessions. For more information on media port ranges and number of sessions, see Configuring RTP Base UDP Port on page 195.

4. Under the SIP Account group, configure the device's registration with the SIP Trunk:
  • 'Account Type': Configure whether the device must perform registration or authentication with the SIP Trunk (**None**, **Registration** or **Authentication**).
  • 'Trunk Main Line': Configure the "leading number" assigned by the SIP Trunk. Many SIP Trunks use the same value for Trunk Main Line and Username parameters.
  • 'Username': Configure the SIP authentication username (as provided by the SIP Trunk provider).
  • 'Password': Configure the SIP authentication password (as provided by the SIP Trunk provider).

5. Click **Next**; the Number Manipulation page appears (see Number Manipulation Page on page 721).
48.7 **Number Manipulation Page**

The Number Manipulation wizard page configures caller (source) and callee (destination) number manipulation for outbound and inbound calls.

![Number Manipulation Page](image)

**Figure 48-7: SBC Configuration Wizard - Number Manipulation Page**

➤ **To configure number manipulation:**

1. Under the Outbound Calls and/or Inbound Calls groups, select the required manipulation check box:
   - Destination Number Manipulation: Manipulates the destination number.
   - Source Number Manipulation: Manipulates the source number.

2. In the 'Prefix' field, configure the prefix (digits at the beginning of the number) to which you want to apply manipulation. If configured to "*" (asterisk), manipulation is applied to all numbers.

3. In the 'Remove' field, configure the number of digits to be removed from the beginning of the number. If configured to "0", no digits are removed.

4. In the 'Add' field, configure a new prefix to be added to the beginning of the number. If not configured, no prefix is added.

5. To enable routing by ARM, select the 'Use ARM for call routing' check box, and then configure the following fields:
   - 'Host': IP address or FQDN of the ARM server.
   - 'Username': Username for communication with ARM.
   - 'Password': Password for communication with ARM.

6. Click **Next**; the Remote Users (FEU) page appears (see Remote Users Page on page 722).

   The example below changes the number "+15033311432" to "03311432":
   ```
   Prefix: "+1503"
   ```
48.8 Remote Users Page

The Remote Users (FEU) wizard page configures the remote users settings.

**Note:** This page is applicable only to IP PBXs that support such configuration.

To configure remote users:

1. Select the 'Enable remote users' check box.
2. Under the SIP Interface group, configure the SIP interface for the remote users:
   - 'Transport Type': Configure the SIP transport type.
   - 'Listening Port': Configure the SIP port used by the device for communicating with remote users. For **One Port: LAN** and **One Port: WAN** network topologies, you must configure different listening ports for communication with the IP PBX and remote users.
3. Under the Media Ports (Realm) group, configure the media protocol type and ports used by the device for communicating with the remote users:
   - 'Media Protocol': Configure the media protocol type (RTP or SRTP).
   - 'Base Port': Configure the first media port.
   - 'Number Of Sessions': Configure the number of required media sessions. For more information on media port ranges and number of sessions, see Configuring RTP Base UDP Port on page 195.
4. Click **Next**; the Summary page appears (see Summary Page on page 723).
48.9 Summary Page

The Summary wizard page displays a summary of your configuration.

Figure 48-9: SBC Configuration Wizard - Summary Page

➢ To review your configuration:

1. Review the configuration:
   - To view the configuration in ini-file format, click the INI File tab.
   - To view the configuration in normal format, click the Configuration Summary tab.

2. To save the configuration as an ini file to a folder on your PC, click the Save INI file button. You can later load the file to the device (see Loading an ini File to the Device on page 691).

3. Click Next; the Congratulations page appears (see Congratulations Page on page 724).
48.10 Congratulations Page

The Congratulations wizard page is the last wizard page and allows you to complete configuration.

Figure 48-10: SBC Configuration Wizard - Congratulations Page

Congratulations!

You have successfully completed the SBC Configuration wizard. Click "Apply & Reset" button to activate the new configuration. Note that device will be restarted and it may take up to 4 minutes before it completes activation.

The generated configuration file is a good "starting point" that enables successful establishment of basic calls.

For complete device configuration you may need to configure additional functionality. For example, you may need to add security configuration (e.g. Firewalls, IDS) to ensure that SBC is protected from malicious user activity and DoS attacks.

Refer to the User Manual for more information.

WARNING: Applying this configuration will overwrite all of the existing device configuration.

To complete the SBC Configuration Wizard:

- Click Apply & Reset to apply configuration to the device or click Save INI File to save configuration as an ini file on your PC.
49 Restoring Factory Defaults

This section describes how to restore the device's configuration to factory defaults.

49.1 Restoring Factory Defaults through CLI

You can restore the device to factory defaults through CLI, as described in the following procedure.

➢ To restore factory defaults through CLI:

1. Access the CLI:
   a. Connect the RS-232 serial port of the device to the communication port on your computer. For serial cabling, refer to the Hardware Installation Manual.
   b. Establish serial communication with the device using a serial communication program (such as HyperTerminalTM) 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

49.2 Restoring Factory Defaults through Web Interface

You can restore the device to factory defaults through the Web interface.

Note: When restoring to factory defaults, you can preserve your IP network settings that are configured in the IP Interfaces table (see "Configuring IP Network Interfaces" on page 137), as described in the procedure below. This may be important, for example, to maintain connectivity with the device (through the OAMP interface) after factory defaults have been applied.
To restore factory defaults through Web interface:

1. Open the Configuration File page:
   - Tool bar: From the Actions drop-down menu, choose Configuration File.

   Figure 49-1: Restoring Factory Defaults through Web

2. To keep your current IP network settings, select the Preserve Network Configuration check box. To overwrite all your IP network settings with the default IP network interface, clear the Preserve Network Configuration check box.
3. Click the Restore Defaults button; a message appears requesting you to confirm.
4. Click OK to confirm or Cancel to return to the page.
5. Once the device is restored to factory defaults, reset the device for the settings to take effect.

49.3 Restoring Defaults through ini File

You can restore the device to factory defaults, by loading an empty ini file to the device. This is done using the Web interface’s Configuration File page (see “Configuration File” on page 691). If the ini file includes parameter settings, 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 username and password.
Part X

Status, Performance Monitoring and Reporting
50 System Status

This section describes how to view various system statuses.

50.1 Viewing Device Information

You can view hardware and software information about the device on the Device Information page.

➢ To view device information:

- Open the Device Information page (Monitor menu > Monitor menu > Summary tab > Device Information).

Figure 50-1: Viewing Device Information (Example)

Device Information Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General Settings</td>
<td></td>
</tr>
<tr>
<td>MAC Address</td>
<td>Media access control (MAC) address.</td>
</tr>
<tr>
<td>Serial Number</td>
<td>Serial number of the CPU.</td>
</tr>
<tr>
<td>Product Key</td>
<td>Product Key, which identifies the specific device purchase. For more</td>
</tr>
<tr>
<td></td>
<td>information, see Viewing the Device’s Product Key on page 681.</td>
</tr>
<tr>
<td>Board Type</td>
<td>Product name of the device.</td>
</tr>
</tbody>
</table>

Figure 50-1: Viewing Device Information (Example)
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Device Up Time</strong></td>
<td>Duration that the device has been up and running since the last reset. The duration is displayed in the following format: <code>dd:hh:mm:ss:100th of a second</code></td>
</tr>
<tr>
<td><strong>Device Administrative State</strong></td>
<td>Administrative status (&quot;Unlocked&quot; or &quot;Locked&quot;), as performed in Locking and Unlocking the Device on page 651.</td>
</tr>
<tr>
<td><strong>Device Operational State</strong></td>
<td>Operational status:</td>
</tr>
<tr>
<td></td>
<td>• &quot;Disabled&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;Enabled&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;Error&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;Unknown&quot;</td>
</tr>
<tr>
<td><strong>Flash Size [Mbytes]</strong></td>
<td>Size of the non-volatile storage memory (flash), measured in megabytes.</td>
</tr>
<tr>
<td><strong>RAM Size [Mbytes]</strong></td>
<td>Size of the random access memory (RAM), measured in megabytes.</td>
</tr>
<tr>
<td><strong>CPU Speed [MHz]</strong></td>
<td>Clock speed of the CPU, measured in megahertz (MHz).</td>
</tr>
<tr>
<td><strong>Versions</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Version ID</strong></td>
<td>Software version number.</td>
</tr>
<tr>
<td><strong>DSP Type</strong></td>
<td>Type of DSP.</td>
</tr>
<tr>
<td><strong>DSP Software Version</strong></td>
<td>DSP software version.</td>
</tr>
<tr>
<td><strong>DSP Software Name</strong></td>
<td>DSP software name.</td>
</tr>
<tr>
<td><strong>Flash Version</strong></td>
<td>Flash memory version number.</td>
</tr>
</tbody>
</table>

**Loaded Files**: Displays installed Auxiliary files. You can also delete a file, by clicking the corresponding **Delete** button, as described in Deleting Auxiliary Files on page 657.

### 50.2 Viewing Device Status on Monitor Page

The Web interface's Monitor page provides basic status and information on the device. The page is useful in that it allows you to easily obtain an overview of the device's operating status at a single glance.

➢ **To view device status and information on the Monitor home page:**

- On the Menu bar, click **Monitor** or if you are already in the Monitor menu's Navigation tree, click 🏘️ **Monitor**.

The Monitor page displays the following groups of information:

- **Device Information pane:**
  - 'Address': IP address of the device's OAMP interface
  - 'Firmware': Software version currently running on the device
  - 'Type': Name of the device
  - 'HA Status': High-Availability (HA) status of the device, if configured for HA. For more information, see Viewing HA Status on Monitor Web Page on page 600.

- **SBC Call Statistics:**
  - Active Calls: Total number of SBC calls. The corresponding SNMP performance monitoring MIB is `PM_gwlnINVITEDialogs`. 
• Average Success Rate (ASR): Number of successfully answered calls out of the total number of attempted calls. The corresponding SNMP performance monitoring MIB is PM_gwSBCASR.
• Average Call Duration (ACD): Average call duration in seconds of established calls. The value is refreshed every 15 minutes and therefore, this value reflects the average duration of all established calls made within a 15 minute period. The corresponding SNMP performance monitoring MIB is PM_gwSBCACD.
• Calls per Sec: Total number of new calls per second (CPS).
• Transactions per Sec: Total number of new SIP transactions per second (out-of-dialog transactions such as INVITE and REGISTER, or in-dialog transactions such as UPDATE and BYE). The corresponding SNMP performance monitoring MIB is PM_gwActiveSIPTransactionsPerSecond. The counter is applicable to SBC and Gateway calls.
• Registered Users: Number of users registered with the device. The corresponding SNMP performance monitoring MIB is PM_gwSBCRegisteredUsers.

Figure 50-2: Viewing Call Statistics on Monitor Page

Graphical display of the device with color-coded status icons, as shown in the figure below and described in the subsequent table:

Table 50-1: Description of Graphical Display of Device

<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1      | Displays the highest severity of an active alarm raised (if any) by the device:  
  • Green = No alarms  
  • Red = Critical alarm |
<table>
<thead>
<tr>
<th>Item #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Orange = Major alarm</td>
</tr>
<tr>
<td></td>
<td>Yellow = Minor alarm</td>
</tr>
<tr>
<td></td>
<td>To view active alarms, click this Alarms area to open the Active Alarms page (see Viewing Active Alarms on page 735).</td>
</tr>
<tr>
<td>2</td>
<td>Gigabit Ethernet port status icons:</td>
</tr>
<tr>
<td></td>
<td>![green icon] (green): Ethernet link is working</td>
</tr>
<tr>
<td></td>
<td>![gray icon] (gray): Ethernet link is not connected</td>
</tr>
<tr>
<td></td>
<td>To view detailed Ethernet port information, click these icons to open the Ethernet Port Information page (see Viewing Ethernet Port Information on page 755).</td>
</tr>
</tbody>
</table>
You can obtain information on the percentage of DSP resources utilized by the device, through the SNMP MIB table, acPMDSPUsage. You can also configure low and high DSP utilization thresholds for this MIB, that if crossed, the SNMP trap event, acPerformanceMonitoringThresholdCrossing is sent by the device. For more information on this MIB, refer to the SNMP Reference Guide.
This page is intentionally left blank.
### 52 Viewing Carrier-Grade Alarms

This section describes how to view SNMP alarms raised by the device.

#### 52.1 Viewing Active Alarms

You can view current (active) alarms in the Web interface that have been raised by the device. If an alarm is cleared, it is moved into the History Alarms table (see "Viewing History Alarms" on page 736).

**Note:**
- The alarms in the table are deleted upon a device reset.
- To configure the maximum number of active alarms that can be displayed in the table, see the ini file parameter, ActiveAlarmTableMaxSize.
- For more information on SNMP alarms, refer to the SNMP Reference Guide document.

➢ **To view active alarms:**

- Open the Active Alarms table, by doing one of the following:
  - Navigation tree: **Monitor** menu > **Monitor** tab > **Summary** folder > **Active Alarms**.
  - Monitor home page: Click the "Alarms" area on the graphical display of the device (see "Viewing Device Status on Monitor Page" on page 730).

<table>
<thead>
<tr>
<th>Sequential #</th>
<th>Severity</th>
<th>Source</th>
<th>Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Major</td>
<td>Board#1</td>
<td>Network element admin state change alarm. Gateway is locked</td>
<td>27/11/2010, 12:22:53</td>
</tr>
<tr>
<td>2</td>
<td>Minor</td>
<td>Board#1/EthernetLink#2</td>
<td>Ethernet link alarm. LAN port number 2 is down.</td>
<td>27/11/2010, 12:22:55</td>
</tr>
<tr>
<td>3</td>
<td>Minor</td>
<td>Board#1/EthernetLink#3</td>
<td>Ethernet link alarm. LAN port number 3 is down.</td>
<td>27/11/2010, 12:22:55</td>
</tr>
<tr>
<td>4</td>
<td>Minor</td>
<td>Board#1/EthernetLink#4</td>
<td>Ethernet link alarm. LAN port number 4 is down.</td>
<td>27/11/2010, 12:22:55</td>
</tr>
</tbody>
</table>

**Table 52-1: Active Alarms Table Description**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Number</td>
<td>The number of the alarm. The alarms are numbered sequentially as they are raised by the device. The numbering resets to 1 immediately after a device reset (i.e., the first alarm raised after a reset is assigned the number #1).</td>
</tr>
<tr>
<td>Severity</td>
<td>Severity level of the alarm:</td>
</tr>
<tr>
<td></td>
<td>- Critical (red)</td>
</tr>
<tr>
<td></td>
<td>- Major (orange)</td>
</tr>
<tr>
<td></td>
<td>- Minor (yellow)</td>
</tr>
<tr>
<td>Source</td>
<td>Component of the device from which the alarm was raised.</td>
</tr>
<tr>
<td>Description</td>
<td>Brief description of the alarm.</td>
</tr>
<tr>
<td>Date</td>
<td>Date (DD/MM/YYYY) and time (HH:MM:SS) the alarm was raised.</td>
</tr>
</tbody>
</table>
52.2 Viewing History Alarms

You can view all SNMP alarms, in the Web interface's Alarms History table, that have been raised (active alarms) as well as cleared (resolved). One of the benefits of this is that you can view alarms that may have been raised and then cleared on a continuous basis. For example, such an alarm may be raised due to an Ethernet cable that is not securely attached to the device's Ethernet port, causing the Ethernet link to be sometimes up and sometimes down. This alarm would not be listed in the Active Alarms table due to it being cleared. The Alarms History table displays both the cleared alarm and the alarm for which it was cleared adjacent to one another, as shown in the figure below for alarms #9 and #10.

To configure the maximum number of alarms that can be displayed in the table, use the AlarmHistoryTableMaxSize ini file parameter. If the maximum is reached and a new alarm is added to the table, the oldest alarm in the table is removed from the table to accommodate the new alarm.

Note:
- The alarms in the table are deleted upon a device reset.
- For more information on SNMP alarms, refer to the SNMP Reference Guide document.

➢ To view history alarms:
  - Open the Alarms History table (Monitor menu > Monitor tab > Summary folder > Alarms History).

Figure 52-1: Viewing Alarms History Table

<table>
<thead>
<tr>
<th>Sequential #</th>
<th>Severity</th>
<th>Source</th>
<th>Description</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Major</td>
<td>Board#1</td>
<td>Network element admin state change alarm. Gateway is locked</td>
<td>27/11/2010, 12:22:53</td>
</tr>
<tr>
<td>2</td>
<td>Minor</td>
<td>Board#1/EtherLink#2</td>
<td>Ethernet link alarm. LAN port number 2 is down.</td>
<td>27/11/2010, 12:22:55</td>
</tr>
<tr>
<td>3</td>
<td>Minor</td>
<td>Board#1/EtherLink#3</td>
<td>Ethernet link alarm. LAN port number 3 is down.</td>
<td>27/11/2010, 12:22:55</td>
</tr>
<tr>
<td>4</td>
<td>Minor</td>
<td>Board#1/EtherLink#4</td>
<td>Ethernet link alarm. LAN port number 4 is down.</td>
<td>27/11/2010, 12:22:55</td>
</tr>
<tr>
<td>7</td>
<td>Major</td>
<td>Board#1/ProxyConnections#0</td>
<td>Proxy Set Alarm Proxy Set 0: Proxy lost, looking for another proxy</td>
<td>17/03/2010, 16:30:01</td>
</tr>
<tr>
<td>8</td>
<td>Cleared</td>
<td>Board#1/ProxyConnections#0</td>
<td>Alarm cleared: Proxy Set Alarm Proxy Set 0: Proxy lost, looking for another proxy</td>
<td>17/03/2010, 16:30:01</td>
</tr>
<tr>
<td>9</td>
<td>Major</td>
<td>Board#1/ProxyConnections#1</td>
<td>Proxy Set Alarm Proxy Set 1: Proxy lost, looking for another proxy</td>
<td>17/03/2010, 16:30:01</td>
</tr>
<tr>
<td>10</td>
<td>Cleared</td>
<td>Board#1/ProxyConnections#1</td>
<td>Alarm cleared: Proxy Set Alarm Proxy Set 1: Proxy lost, looking for another proxy</td>
<td>17/03/2010, 16:30:01</td>
</tr>
</tbody>
</table>

Table 52-2: Alarms History Table Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequential Number</td>
<td>The number of the alarm. The alarms are numbered sequentially as they are raised by the device. The numbering resets to 1 immediately after a device reset (i.e., the first alarm raised after a reset is assigned the number #1).</td>
</tr>
<tr>
<td>Severity</td>
<td>Severity level of the alarm:</td>
</tr>
<tr>
<td></td>
<td>- Critical (red)</td>
</tr>
<tr>
<td></td>
<td>- Major (orange)</td>
</tr>
<tr>
<td></td>
<td>- Minor (yellow)</td>
</tr>
<tr>
<td></td>
<td>- Cleared (green)</td>
</tr>
<tr>
<td>Source</td>
<td>Component of the device from which the alarm was raised.</td>
</tr>
<tr>
<td>Description</td>
<td>Brief description of the alarm.</td>
</tr>
</tbody>
</table>
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 Viewing Management User Activity Logs

If you have enabled the reporting of management user activities performed in the device's management interfaces (see "Configuring Reporting of Management User Activities" on page 799), you can view the logged activities in the Web interface, as described in the procedure below.

➢ To view management user activity logs:

- Open the Activity Log table (Monitor menu > Monitor tab > Summary folder > Activity Log).

**Figure 53-1: Viewing Management User Activity Log**

<table>
<thead>
<tr>
<th>ID</th>
<th>TIME</th>
<th>DESCRIPTION</th>
<th>USER</th>
<th>INTERFACE</th>
<th>CLIENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>11/29/2010, 11:40:30</td>
<td>System configuration has be</td>
<td>Admin</td>
<td>WEB</td>
<td>10:13:2:3</td>
</tr>
</tbody>
</table>

**Table 53-1: Activity Log Table Description**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Date (mm/dd/yyyy) and time (hh:mm:ss) that the activity was performed.</td>
</tr>
<tr>
<td>Description</td>
<td>Description of the activity.</td>
</tr>
<tr>
<td>User</td>
<td>Username of the user account that performed the activity.</td>
</tr>
<tr>
<td>Interface</td>
<td>Protocol used for connecting to the management interface (e.g., Telnet, SSH, Web, or HTTP).</td>
</tr>
<tr>
<td>Client</td>
<td>IP address of the client PC from where the user accessed the management interface.</td>
</tr>
</tbody>
</table>
This page is intentionally left blank.
54 Viewing Performance Monitoring

This section describes how to view performance monitoring in the device's Web interface.

54.1 Viewing Call Success and Failure Ratio

You can view success and failure ratio of SIP dialogs in the Web interface's Success/Failure Ratio page. You can filter the display by a specific SRD or IP Group, and by call direction and type of SIP dialog (e.g., INVITEs only). The information is displayed in the following pie charts:

- **Dialog Success Ratio**: Displays the SIP call and subscribe (SUBSCRIBE) dialog success-failed ratio.
- **Dialog Failed Attempts**: Displays failed SIP dialog attempts. This includes the number of calls and subscribes which were successfully and abnormally terminated.
- **Dialog Termination Ratio**: Displays SIP dialog termination by reason (e.g., due to no answer).

➢ **To view success and failed call ratio**:

1. Open the Success/Failure Ratio page (Monitor menu > Monitor tab > Performance Monitoring folder > Success / Failure Ratio).

![Figure 54-1: Viewing Success/Failure Ratio]

2. From the 'SRD/IP Group' drop-down list, select whether you want to view statistic for an SRD or IP Group.

3. From the 'Index' drop-down list, select the SRD or IP Group index.
4. From the 'Direction' 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

If there is no data for the charts, the chart appears gray and "No Data" is displayed to the right of the chart.

- To refresh the charts:
  - Click **Refresh**.

- To reset the counters:
  - Click **Reset Counters**.

### 54.2 Viewing Average Call Duration

You can view the number of currently active calls and the average call duration (ACD) in the Web interface's Average Call Duration page. You can filter display by a specific SRD or IP Group. The page displays the following two graphs:

- **Upper graph**: Displays the number of currently active calls (INVITEs). The x-axis indicates the time (hh:mm:ss) and the y-axis the number of calls.

- **Lower graph**: Displays the ACD. The x-axis indicates the time (hh:mm:ss) and the y-axis the average call duration. The ACD is refreshed every 15 minutes and therefore, this value reflects the average duration of all established calls made within a 15 minute period.
To view number of active calls and average call duration:

1. Open the Average Call Duration page (Monitor menu > Monitor tab > Performance Monitoring folder > Average Call Duration).

   ![Figure 54-2: Viewing Average Call Duration](image)

   - From the 'SRD / IP Group' drop-down list, select the configuration entity (SRD or IP Group).
   - From the 'Index' drop-down list, select the specific 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 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.

54.3 Configuring Performance Profiles

The Performance Profile table lets you configure up to 6,303 Performance Profile rules. A Performance Profile rule defines thresholds of performance monitoring call metrics for Major and Minor severity alarms. If the threshold is crossed, the device raises the corresponding severity alarm. You can configure a Performance Profile rule for all calls (globally), or per SRD or IP Group.

You can configure the alarm thresholds for the following call metrics:

- **Answer Success Ratio or ASR (also known as Answer Seizure Ratio):** The number (in percentage) of answered calls (i.e., number of seizures resulting in an answer signal) out of the total number of attempted calls (seizures). The metric is calculated for the outgoing call leg. The metric includes the following SNMP performance monitoring MIBs:
  - **PM_gwSBCASR:** ASR for all (global) entities (i.e., all IP Groups and SRDs)
Network Effectiveness Ratio (NER): The number (in percentage) of successfully connected calls out of the total number of attempted calls (seizures). The metric measures the ability of the network to deliver a call to the called terminal. In addition to answered calls, the following SIP response codes are regarded as successfully connected calls: 408 (Request Timeout), 480 (Temporarily Unavailable), and 486 (Busy Here). The metric is calculated for the outgoing call leg. The metric includes the following SNMP performance monitoring MIBs:

- PM_gwSBCNER: NER for all (global) entities (i.e., all IP Groups and SRDs)
- PM_gwSBCIPGroupNER: NER per IP Group
- PM_gwSBCSRDNER: NER per SRD

If the configured NER minor or major thresholds are crossed, the device raises the SNMP alarm, AcNERThresholdAlarm (OID 1.3.6.1.4.1.5003.9.10.1.21.2.0.113).

Average Call Duration (ACD): The ACD plus the session disconnect time (SDD) is the duration from when the SIP 200 OK is received to when the SIP Bye message is sent. The metric is calculated for both incoming and outgoing call legs. The metric includes the following SNMP performance monitoring MIBs:

- PM_gwSBCACD: ACD for all (global) entities (i.e., all IP Groups and SRDs)
- PM_gwSBCIPGroupACD: ACD per IP Group
- PM_gwSBCSRDACD: ACD per SRD

If the configured ACD minor or major thresholds are crossed, the device raises the SNMP alarm, acACDThresholdAlarm (OID 1.3.6.1.4.1.5003.9.10.1.21.2.0.112).

At any given time during a call, a voice metric can be in one of the following color-coded quality states (as displayed by SEM):

- Green: Indicates good call quality
- Yellow: Indicates fair call quality
- Red: Indicates poor call quality

When the threshold of a voice metric is crossed, the device changes the alarm severity and corresponding color-coded quality state of the call:

- Minor Threshold (Yellow): Lower threshold that indicates changes from Green or Red to Yellow.
- Major Threshold (Red): Higher threshold that indicates changes from Green or Yellow to Red.

The device also uses hysteresis to determine whether the threshold has indeed being crossed. Hysteresis defines the amount of fluctuation from the 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. Hysteresis is used only for threshold crossings toward a lesser severity (i.e., from Red to Yellow, Red to Green, or Yellow to Green).
The following example is used to explain how the device considers threshold crossings. The example is based on the ASR of a call, where the Major threshold is configured to 70%, the Minor threshold to 90% and the hysteresis for both thresholds to 2%:

![Figure 54-3: Example of Threshold Crossings (ASR)](image)

<table>
<thead>
<tr>
<th>Threshold Crossing</th>
<th>Calculation</th>
<th>Threshold based on Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green to Yellow (Minor alarm)</td>
<td>The change occurs if the measured metric crosses the configured Minor threshold only (i.e., hysteresis is not used).</td>
<td>90%</td>
</tr>
<tr>
<td>Green to Red (Major alarm)</td>
<td>The change occurs if the measured metric crosses the configured Major threshold only (i.e., hysteresis is not used).</td>
<td>70%</td>
</tr>
<tr>
<td>Yellow to Red (Major alarm)</td>
<td>The change occurs if the measured metric crosses the configured Major threshold only (i.e., hysteresis is not used).</td>
<td>70%</td>
</tr>
<tr>
<td>Red to Yellow (Minor alarm)</td>
<td>The change occurs if the measured metric crosses the configured Major threshold with hysteresis.</td>
<td>72% (i.e., 70 + 2)</td>
</tr>
<tr>
<td>Red to Green (alarm cleared)</td>
<td>The change occurs if the measured metric crosses the configured Minor threshold with hysteresis.</td>
<td>92 (i.e., 90 + 2)</td>
</tr>
<tr>
<td>Yellow to Green (alarm cleared)</td>
<td>The change occurs if the measured metric crosses the configured Minor threshold with hysteresis.</td>
<td>92 (i.e., 90 + 2)</td>
</tr>
</tbody>
</table>

**Note:**
- Forwarded calls are not considered in the calculation for ASR and NER.
- If you don't configure thresholds for a specific metric, the device still provides current performance monitoring values of the metric, but does not raise any threshold alarms for it.
- You can configure the device to perform certain actions, for example, reject calls to the IP Group for a user-defined duration, if a threshold is crossed. For more information, see "Configuring Quality of Service Rules" on page 316.
The following procedure describes how to configure Performance Profile rules through the Web interface. You can also configure it through ini file (PerformanceProfile) or CLI (configure system > performance-profile).

➢ To configure a Performance Profile rule:

1. Open the Performance Profile table (Monitor menu > Monitor tab > Performance Monitoring folder > Performance Profile).
2. Click New; the following dialog box appears:

   Figure 54-4: Performance Profile Table - Dialog Box

3. Configure the rule according to the parameters described in the table below.
4. Click Apply.

Table 54-2: Performance Profile Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[PerformanceProfile_Index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Entity</td>
<td>Defines a configuration entity type to which you want to apply the rule.</td>
</tr>
<tr>
<td>[PerformanceProfile_Entity]</td>
<td>[0] Global = (Default) The device calculates call metrics for all calls.</td>
</tr>
<tr>
<td></td>
<td>[1] SRD = Assigns an SRD. To specify the SRD, use the 'SRD' parameter (see below).</td>
</tr>
<tr>
<td></td>
<td>[2] IP Group = Assigns an IP Group. To specify the IP Group, use the 'IP Group' parameter (see below).</td>
</tr>
<tr>
<td>IP Group</td>
<td>Assigns an IP Group to the rule.</td>
</tr>
<tr>
<td>ip-group-name</td>
<td>Note: The parameter is applicable only if you configure the 'Entity' parameter to IP Group.</td>
</tr>
<tr>
<td>[PerformanceProfile_IPGroupName]</td>
<td>Assigns an SRD to the rule.</td>
</tr>
<tr>
<td>srd-name</td>
<td>Note: The parameter is applicable only if you configure the 'Entity' parameter to SRD.</td>
</tr>
<tr>
<td>[PerformanceProfile_SRDName]</td>
<td>Defines the type of performance monitoring metric for which you want to configure thresholds.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pmtype [PerformanceProfile_PMTtype]</td>
<td>- [16] ASR (Default)</td>
</tr>
<tr>
<td></td>
<td>- [17] ACD</td>
</tr>
<tr>
<td></td>
<td>- [18] NER</td>
</tr>
</tbody>
</table>

**Action**

#### Minor Threshold

**minor-threshold [PerformanceProfile_MinorThreshold]**

Defines the Minor threshold (in percentage) of the selected performance monitoring metric, which is the lower threshold located between the Yellow and Green states.

To consider a threshold crossing:

- **Increase in severity (i.e., Green to Yellow):** Only this value is used.
- **Decrease in severity (Red to Green, or Yellow to Green):** This value is used with the hysteresis, configured by the ‘Hysteresis’ parameter (see below).

The valid range is 0 to 100. The default is 0.

#### Major Threshold

**major-threshold [PerformanceProfile_MajorThreshold]**

Defines the Major threshold (in percentage) of the selected performance monitoring metric, which is the upper threshold located between the Yellow and Red states.

To consider a threshold crossing:

- **Increase in severity (i.e., Yellow to Red, or Green to Red):** Only this value is used.
- **Decrease in severity (Red to Yellow):** This value is used with the hysteresis, configured by the ‘Hysteresis’ parameter (see below).

The valid range is 0 to 100. The default is 0.

#### Hysteresis

**hysteresis [PerformanceProfile_Hysteresis]**

Defines the amount of fluctuation (hysteresis) from the configured threshold in order for the threshold to be considered as crossed. Hysteresis is used to avoid false reports being sent by the device. Hysteresis is used only when the severity level decreases (i.e., from Red to Yellow, Yellow to Green, or Red to Green).

The valid value is 0 to 15 (in percentage). The default is 5.

For example, if you configure the 'Major Threshold' parameter to 70% and the 'Hysteresis' parameter to 2%, the device considers a threshold crossing from Red to Yellow only if the ASR crosses 72% (i.e., 70% + 2%).

#### Minimum Samples

**minimum-samples [PerformanceProfile_MinimumSample]**

Defines the minimum number of call sessions (sample) that is required for the device to calculate the performance monitoring metrics (per window size). If the number of call sessions is less than the configured value, no calculation is done.

The default is 10 calls.

**Note:** The calculation also depends on the configured sampling window size (see 'Window Size' parameter). For example, if the parameter is configured to 10 calls, but only 5 calls were processed during the configured sampling window, no calculation is done.

#### Window Size

**window-size**

Defines the time interval (in minutes) during which the device calculates the performance monitoring metrics. For example, if the parameter is configured to five minutes, the calculation
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [PerformanceProfile_WindowSize] | is done for the last five minutes. The default is 5 minutes.  
**Note:** The calculation depends on the configured minimum samples (see 'Minimum Samples' parameter). For example, if the parameter is configured to five minutes, but the number of calls during the interval is less than the configured minimum samples, no calculation is done. |
55 Viewing VoIP Status

This section describes how to view VoIP-related status.

55.1 Viewing SBC Registered Users

You can view SBC users that are registered with the device. For each user, the Address of Record (AOR) and the corresponding contacts are shown. An AOR is a SIP or SIPS URI that points to a domain with a location service that can map the URI to another URI (contact) where the user might be available. A contact is a SIP URI that can be used to contact that specific instance of the user agent for subsequent requests.

➢ To view registered SBC users:

► Web: SBC Registered Users page (Monitor menu > Monitor tab > VoIP Status folder > SBC Registered Users).

Figure 55-1: Viewing Registered SBC Users

<table>
<thead>
<tr>
<th>ADDRESS OF RECORD</th>
<th>CONTACT</th>
</tr>
</thead>
</table>

Table 55-1: SBC Registered Users Table Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address of Record</td>
<td>AOR (e.g., 1000@10.8.5.71)</td>
</tr>
<tr>
<td>Contact</td>
<td>Contacts corresponding to the AOR, for example:</td>
</tr>
<tr>
<td></td>
<td><a href="">sip:1000@10.8.5.71:5060</a>;expires=180; Active status:1</td>
</tr>
<tr>
<td></td>
<td>The contact's registration status is displayed:</td>
</tr>
<tr>
<td></td>
<td>▪ &quot;Active status:1&quot;: The contact has been successfully registered and calls can be routed to it.</td>
</tr>
<tr>
<td></td>
<td>▪ &quot;Active status:0&quot;: 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.</td>
</tr>
</tbody>
</table>

➢ CLI:

- SBC users:
  # show voip register db sbc list
- SBC contacts of a specified AOR:
  # show voip register db sbc user <Address Of Record>
55.2 Viewing Proxy Set Status

You can view the status of Proxy Sets that are used in your call routing topology. Proxy Sets that are not associated with any routing rule are not displayed.

To configure proxy Sets, see Configuring Proxy Sets on page 359.

To view Proxy Set status:
- Open the Active Proxy Set Status page (Monitor menu > Monitor tab > VoIP Status folder > Proxy Sets Status).

![Figure 55-2: Viewing Proxy Sets Status](image)

Table 55-2: Proxy Sets Status Table Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proxy Set ID</td>
<td>Displays the Proxy Set ID.</td>
</tr>
<tr>
<td>Mode</td>
<td>Displays the Proxy Sets' operational mode:</td>
</tr>
<tr>
<td></td>
<td>&quot;Parking&quot; or &quot;Homing&quot;: Redundancy mode, as configured by the ProxySet_ProxyRedundancyMode parameter.</td>
</tr>
<tr>
<td></td>
<td>&quot;Load Balancing&quot;: Proxy load balancing mode, as configured by the ProxySet_ProxyRedundancyMode parameter.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Configuring Proxy Sets.</td>
</tr>
<tr>
<td>Keep Alive</td>
<td>Displays whether the Proxy Keep-Alive feature is enabled (&quot;Enabled&quot;) or disabled (&quot;Disabled&quot;), as configured by the ProxySet_EnableProxyKeepAlive parameter (see Configuring Proxy Sets).</td>
</tr>
<tr>
<td>Address</td>
<td>Displays the IP address of the proxy server. This can be the IP address as configured in dotted-decimal notation for the Proxy Set, or the resolved IP address of a DNS query if an FQDN is configured for the Proxy Set. IP addresses resolved from FQDNs are displayed as &quot;&lt;FQDN name&gt;(&lt;resolved IP address&gt;)&quot;, for example, &quot;abc.com(10.8.6.80)&quot;. The IP address that is currently used for routing is indicated with an asterisk, for example, &quot;10.8.6.89(*)&quot;. If the FQDN failed to be resolved, only the FQDN name is displayed (e.g., &quot;abc.com&quot;).</td>
</tr>
</tbody>
</table>
55. Viewing VoIP Status

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Displays the priority of IP addresses resolved from FQDNs. <strong>Note:</strong> The field is applicable only to Proxy Sets configured with FQDNs.</td>
</tr>
<tr>
<td>Weight</td>
<td>Displays the weight of IP addresses resolved from FQDNs. <strong>Note:</strong> The field is applicable only to Proxy Sets configured with FQDNs.</td>
</tr>
<tr>
<td>Success Count</td>
<td>Displays the total number of successful keep-alive messages (by SIP OPTIONS) sent by the device to the proxy.</td>
</tr>
<tr>
<td>Failure Count</td>
<td>Displays the total number of failed keep-alive messages (by SIP OPTIONS) sent by the device to the proxy.</td>
</tr>
<tr>
<td>Status</td>
<td>Displays the status of the Proxy Set and its' proxy servers.</td>
</tr>
<tr>
<td></td>
<td>• &quot;ONLINE&quot;:</td>
</tr>
<tr>
<td></td>
<td>✓ Proxy Set ID row: At least one proxy is online as determined by the device's keep-alive feature. The status is also &quot;ONLINE&quot; for IP addresses resolved from DNS queries even if keep-alive is disabled.</td>
</tr>
<tr>
<td></td>
<td>✓ Proxy server rows (if multiple addresses): The proxy server is online as determined by the device's keep-alive feature.</td>
</tr>
<tr>
<td></td>
<td>• &quot;OFFLINE&quot;: The proxy is offline as determined by the device's keep-alive feature and the Proxy Set is configured for Homing ('Redundancy Mode' parameter) or enabled for load balancing ('Proxy Load Balancing Method' parameter):</td>
</tr>
<tr>
<td></td>
<td>✓ Homing: The proxy is the main proxy, but the keep-alive has failed.</td>
</tr>
<tr>
<td></td>
<td>✓ Load balancing: The keep-alive for the proxy has failed.</td>
</tr>
<tr>
<td></td>
<td>• &quot;NOT RESOLVED&quot;: Proxy address is configured as an FQDN, but the DNS resolution has failed.</td>
</tr>
<tr>
<td></td>
<td>• Empty field: Keep-alive for the proxy is disabled or the device has yet to send a keep-alive to the proxy.</td>
</tr>
</tbody>
</table>

55.3 Viewing Registration Status

You can view the registration status of the device’s SIP Accounts.

➢ **To view registration status:**

- Open the Registration Status table (Monitor menu > Monitor tab > VoIP Status folder > Registration Status).

<table>
<thead>
<tr>
<th>INDEX</th>
<th>GROUP TYPE</th>
<th>GROUP NAME</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IP Group</td>
<td></td>
<td>NOT REGISTERED</td>
</tr>
</tbody>
</table>

Table 55-3: Registration Status Table Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts Registration Status</td>
<td>Displays the status registration per Account, as configured in the Accounts table (see &quot;Configuring Registration Accounts&quot; on page 375).</td>
</tr>
</tbody>
</table>
### 55.4 Viewing CDR Test Calls

You can view Call Detail Records (CDR) of SBC test calls, configured in "Configuring Test Call Endpoints" on page 811. These CDRs are stored on the device's non-volatile memory. When a new CDR is generated, the device adds it to the top of the table and all existing entries are shifted one row down in the table. The table displays the last 4,096 CDRs. If the table reaches maximum capacity of entries and a new CDR is added, the last CDR entry is removed from the table.

**Note:** If the device is reset, all CDRs are deleted from memory and from the table.

**To view CDRs of test calls:**

- **Web:** Open the Test Call CDR History table (Monitor menu > Monitor tab > VoIP Status folder > Test Call CDR History).

![Figure 55-3: Viewing CDRs of Test Calls](image)

- **CLI:**
  - All CDR history:
    ```
    # show voip calls history test
    ```
  - CDR history for a specific SIP session ID:
    ```
    # show voip calls history test <session ID>
    ```

**Table 55-4: Test Call CDR History Table**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call End Time</td>
<td>Displays the time at which the call ended. The time is displayed in the format, hh:mm:ss, where hh is the hour, mm the minutes and ss the seconds (e.g., 15:06:36).</td>
</tr>
<tr>
<td>End Point</td>
<td>Indicates that this is test call.</td>
</tr>
<tr>
<td>Caller</td>
<td>Displays the URI (user or user@host) of the test endpoint (caller).</td>
</tr>
<tr>
<td>Callee</td>
<td>Displays the destination (called) URI (user@host).</td>
</tr>
<tr>
<td>Direction</td>
<td>Displays the direction of the call:</td>
</tr>
<tr>
<td></td>
<td>&quot;Incoming&quot;</td>
</tr>
<tr>
<td></td>
<td>&quot;Outgoing&quot;</td>
</tr>
<tr>
<td>Remote IP</td>
<td>Displays the IP address of the call party. For an &quot;Incoming&quot; call, this is</td>
</tr>
</tbody>
</table>
55. Viewing VoIP Status

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>the source IP address; for an &quot;Outgoing&quot; call, this is the destination IP address.</td>
<td></td>
</tr>
<tr>
<td>Duration</td>
<td>Displays the duration of the call, displayed in the format hh:mm:ss, where hh is hours, mm minutes and ss seconds. For example, 00:01:20 denotes 1 minute and 20 seconds.</td>
</tr>
<tr>
<td>Termination Reason</td>
<td>Displays the reason for the call being released (ended). For example, &quot;NORMAL_CALL_CLEAR&quot; indicates a normal termination.</td>
</tr>
<tr>
<td>Session ID</td>
<td>Displays the SIP session ID of the call.</td>
</tr>
</tbody>
</table>

55.5 Viewing SBC CDR History

You can view historical Call Detail Records (CDR) of SBC calls in a table. History CDRs are stored on the device’s memory. When a new CDR is generated, the device adds it to the top of the table and all existing entries are shifted one down in the table. The table displays the last 4,096 CDRs. If the table reaches maximum capacity of entries and a new CDR is added, the last CDR entry is removed from the table.

Note: If the device is reset, all CDRs are deleted from memory and from the table.

To view SBC CDR history:

- **Web:** Open the SBC CDR History table (Monitor menu > Monitor tab > VoIP Status folder > SBC CDR History).

  ![Figure 55-4: SBC CDR History Table](image)

  Table 55-5: SBC CDR History Table

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call End Time</td>
<td>Displays the time at which the call ended. The time is displayed in the format, hh:mm:ss, where hh is the hour, mm the minutes and ss the seconds (e.g., 15:06:36).</td>
</tr>
<tr>
<td>IP Group</td>
<td>Displays the IP Group of the leg for which the CDR was generated.</td>
</tr>
<tr>
<td>Caller</td>
<td>Displays the phone number (source URI user@host) of the party who made the call.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Callee</td>
<td>Displays the phone number (destination URI user@host) of the party to whom the call was made.</td>
</tr>
<tr>
<td>Direction</td>
<td>Displays the direction of the call:</td>
</tr>
<tr>
<td></td>
<td>- &quot;Incoming&quot;</td>
</tr>
<tr>
<td></td>
<td>- &quot;Outgoing&quot;</td>
</tr>
<tr>
<td>Remote IP</td>
<td>Displays the IP address of the call party. For an &quot;Incoming&quot; call, this is the source IP address; for an &quot;Outgoing&quot; call, this is the destination IP address.</td>
</tr>
<tr>
<td>Duration</td>
<td>Displays the duration of the call, displayed in the format hh:mm:ss, where hh is hours, mm minutes and ss seconds. For example, 00:01:20 denotes 1 minute and 20 seconds.</td>
</tr>
<tr>
<td>Termination Reason</td>
<td>Displays the reason for the call being released (ended). For example, &quot;NORMAL_CALL_CLEAR&quot; indicates a normal termination.</td>
</tr>
<tr>
<td>Session ID</td>
<td>Displays the SIP session ID of the call.</td>
</tr>
</tbody>
</table>
56 Viewing Network Status

This section describes how to view network-related status.

56.1 Viewing Active IP Interfaces

You can view the device's active IP interfaces that are configured in the IP Interfaces table (see "Configuring IP Network Interfaces" on page 137).

➢ To view active IP network interfaces:
  ■ Open the IP Interface Status page (Monitor menu > Monitor tab > Network Status folder > IP Interface Status).

<table>
<thead>
<tr>
<th>INDEX</th>
<th>APPLICATION TYPE</th>
<th>IP ADDRESS</th>
<th>INTERFACE MODE</th>
<th>PREFIX LENGTH</th>
<th>DEFAULT GATEWAY</th>
<th>INTERFACE NAME</th>
<th>PRIMARY DNS SERVER IP ADDRESS</th>
<th>SECONDARY DNS SERVER IP ADDRESS</th>
<th>UNDERLYING DEVICE</th>
<th>ADDRESS STATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>O+M+C</td>
<td>10.15.7.96</td>
<td>IPv4 Manual</td>
<td>15</td>
<td>10.15.0.1</td>
<td>O+M+C</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>vlan 1</td>
<td>Permanent</td>
</tr>
<tr>
<td>NA</td>
<td>Internal</td>
<td>169.253.254.254</td>
<td>IPv4 Manual</td>
<td>16</td>
<td>0.0.0.0</td>
<td>Internal if</td>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>Internal if</td>
<td>Permanent</td>
</tr>
</tbody>
</table>

56.2 Viewing Ethernet Device Status

You can view the status of configured Ethernet Devices that have been successfully applied. To configure Ethernet Devices, see "Configuring Underlying Ethernet Devices" on page 134.

➢ To view Ethernet Device status:
  ■ Open the Ethernet Device Status page (Monitor menu > Monitor tab > Network Status folder > Ethernet Device Status).

<table>
<thead>
<tr>
<th>INDEX</th>
<th>VLAN ID</th>
<th>UNDERLYING INTERFACE</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>GROUP_1</td>
<td>vlan 1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>GROUP_2</td>
<td>vlan 2</td>
</tr>
</tbody>
</table>

56.3 Viewing Ethernet Port Information

You can view status information of the device's Ethernet ports. To configure Ethernet ports, see "Configuring Underlying Ethernet Devices" on page 134.

Note: If the device is operating in High-Availability mode, you can also view Ethernet port information of the redundant device, by opening the Redundant Ethernet Port Information page (Monitor menu > Monitor tab > Network Status folder > Redundant Ethernet Port Information).

➢ To view Ethernet port information:
  ■ Open the Ethernet Port Information table, by doing one of the following:
    • Navigation tree: Monitor menu > Monitor tab > Network Status folder > Ethernet Port Information.
    • Monitor home page: Click an Ethernet port on the graphical display of the device (see "Viewing Device Status on Monitor Page" on page 730).
Table 56-1: Ethernet Port Information Table Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Name</td>
<td>Displays the name of the port.</td>
</tr>
<tr>
<td>Active</td>
<td>Displays whether the port is active (&quot;Yes&quot;) or not (&quot;No&quot;).</td>
</tr>
<tr>
<td>Speed</td>
<td>Displays the speed of the Ethernet port.</td>
</tr>
<tr>
<td>Duplex Mode</td>
<td>Displays whether the port is half- or full-duplex.</td>
</tr>
<tr>
<td>State</td>
<td>Displays the state of the port:</td>
</tr>
<tr>
<td></td>
<td>• &quot;Forwarding&quot;: Active port (data is being transmitted and received)</td>
</tr>
<tr>
<td></td>
<td>• &quot;Disabled&quot;: Redundancy port</td>
</tr>
<tr>
<td>Group Member</td>
<td>Displays the Ethernet Group to which the port belongs.</td>
</tr>
</tbody>
</table>

56.4 Viewing Static Routes Status

You can view the status of static IP routes, configured in the Static Routes table (see "Configuring Static IP Routing" on page 144) 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 Static Route Status table (Monitor menu > Monitor tab > Network Status folder > Static Route Status).

Figure 56-1: Viewing Static Route Status
### 57 Reporting Information to External Party

This section describes features for reporting various information to an external party.

#### 57.1 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. The 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. 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 through SNMP.

**Note:**
- The RTCP XR feature is available only if the device is installed with a License Key that includes this feature. For installing a License Key, see "License Key" on page 669.
- If the RTCP XR feature is unavailable (not licensed or disabled), the R-factor VoIP metrics are not provided in CDRs (CDR fields, Local R Factor and Remote R Factor) generated by the device. Instead, these CDR fields are sent with the value 127, meaning that information is unavailable.

You can configure the device to send RTCP XR to a specific IP Group. In addition, you can configure the stage of the call at which you want the device to send RTCP XR:
- End of the call.
- Periodically, according to a user-defined interval between consecutive reports.

The device sends RTCP XR in SIP PUBLISH messages. The PUBLISH message contains the following RTCP XR related header values:
- From and To: Telephone extension number of the user.
- Request-URI: IP address and port of the SEM server.
- Event: "vq-rtcpxr"
- Content-Type: "application/vq-rtcpxr"

The type of RTCP XR report event (VQReportEvent) supported by the device is VQSessionReport (SessionReport). The device can include local and remote metrics in the RTCP XR. Local metrics are generated by the device while remote metrics are provided by the remote endpoint. The following table lists the supported voice metrics (parameters) published in the RTCP XR.

**Table 57-1: RTCP XR Published VoIP Metrics**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CallID</td>
<td>-</td>
<td>Call ID - call ID from the SIP dialog</td>
</tr>
<tr>
<td>LocalID</td>
<td>-</td>
<td>Local ID - identifies the reporting endpoint for the media session</td>
</tr>
<tr>
<td>Metric</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RemoteID</td>
<td></td>
<td>Remote ID - identifies the remote endpoint of the media session</td>
</tr>
<tr>
<td>OrigID</td>
<td></td>
<td>Originating ID - Identifies the endpoint which originated the session</td>
</tr>
<tr>
<td>LocalAddr</td>
<td></td>
<td>Local Address - IP address, port, and SSRC of the endpoint/UA which is the receiving end of the stream being measured</td>
</tr>
<tr>
<td>RemoteAddr</td>
<td></td>
<td>Remote Address - IP address, port, and SSRC of the the source of the stream being measured</td>
</tr>
<tr>
<td>LocalGroup</td>
<td></td>
<td>Local Group ID - identification for the purposes of aggregation for the local endpoint</td>
</tr>
<tr>
<td>RemoteGroup</td>
<td></td>
<td>Remote Group ID - identification for the purposes of aggregation for the remote endpoint</td>
</tr>
<tr>
<td>LocalMAC</td>
<td></td>
<td>Media Access Control (MAC) address of the local SIP device</td>
</tr>
<tr>
<td>Timestamps</td>
<td>START</td>
<td>Start time of the media session</td>
</tr>
<tr>
<td></td>
<td>STOP</td>
<td>End time of the media session</td>
</tr>
<tr>
<td>SessionDesc</td>
<td>PT</td>
<td>Payload Type - 'payload type' parameter in the RTP packets (i.e., the codec).</td>
</tr>
<tr>
<td></td>
<td>PD</td>
<td>Payload Description - description of the codec</td>
</tr>
<tr>
<td></td>
<td>SR</td>
<td>Sample Rate - rate at which the voice was sampled</td>
</tr>
<tr>
<td></td>
<td>FD</td>
<td>Frame Duration (msec) - packetization rate</td>
</tr>
<tr>
<td></td>
<td>FO</td>
<td>Frame Octets - number of octets in each frame per RTP packet</td>
</tr>
<tr>
<td></td>
<td>FPP</td>
<td>Frames per Packets - number of frames per RTP packet</td>
</tr>
<tr>
<td></td>
<td>PLC</td>
<td>Packet Loss Concealment - indicates whether a PLC algorithm was used for the session (&quot;0&quot; - unspecified; &quot;1&quot; - disabled; &quot;2&quot; - enhanced; &quot;3&quot; - standard)</td>
</tr>
<tr>
<td></td>
<td>SSUP</td>
<td>Silence Suppression State - indicates whether silence suppression, also known as Voice Activity Detection (VAD) is enabled (&quot;on&quot; or &quot;off&quot;)</td>
</tr>
<tr>
<td>JitterBuffer</td>
<td>JBA</td>
<td>Jitter Buffer Adaptive - indicates the jitter buffer in the endpoint (&quot;0&quot; - unknown; &quot;1&quot; - reserved; &quot;2&quot; - non-adaptive; &quot;3&quot; - adaptive)</td>
</tr>
<tr>
<td></td>
<td>JBR</td>
<td>Jitter Buffer Rate</td>
</tr>
<tr>
<td></td>
<td>JBN</td>
<td>Jitter Buffer Nominal</td>
</tr>
<tr>
<td></td>
<td>JBM</td>
<td>Jitter Buffer Max</td>
</tr>
<tr>
<td></td>
<td>JBX</td>
<td>Jitter Buffer Abs Max</td>
</tr>
<tr>
<td>PacketLoss</td>
<td>NLR</td>
<td>Network Packet Loss Rate</td>
</tr>
<tr>
<td></td>
<td>JDR</td>
<td>Jitter Buffer Discard Rate</td>
</tr>
<tr>
<td>BurstGapLoss</td>
<td>BLD</td>
<td>Burst Loss Density</td>
</tr>
<tr>
<td></td>
<td>BD</td>
<td>Burst Duration</td>
</tr>
<tr>
<td></td>
<td>GLD</td>
<td>Gap Loss Density</td>
</tr>
<tr>
<td></td>
<td>GD</td>
<td>Gap Duration</td>
</tr>
<tr>
<td></td>
<td>GMIN</td>
<td>Minimum Gap Threshold</td>
</tr>
<tr>
<td>Metric</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Delay</td>
<td>RTD</td>
<td>Round Trip Delay (msec)</td>
</tr>
<tr>
<td></td>
<td>ESD</td>
<td>End System Delay (msec)</td>
</tr>
<tr>
<td></td>
<td>OWD</td>
<td>One Way Delay (msec)</td>
</tr>
<tr>
<td></td>
<td>IAJ</td>
<td>Inter-Arrival Jitter (msec)</td>
</tr>
<tr>
<td></td>
<td>MAJ</td>
<td>Mean Absolute Jitter (msec)</td>
</tr>
<tr>
<td>Signal</td>
<td>SL</td>
<td>Signal Level (dB) - ratio of the signal level to a 0 dBm0 reference</td>
</tr>
<tr>
<td></td>
<td>NL</td>
<td>Noise Level (dB) - ratio of the silent period background noise level to a 0 dBm0 reference</td>
</tr>
<tr>
<td></td>
<td>RERL</td>
<td>Residual Echo Return Noise (dB) - ratio between the original signal and the echo level as measured after echo cancellation or suppression has been applied.</td>
</tr>
<tr>
<td>QualityEst</td>
<td>RLQ</td>
<td>Listening Quality R - listening quality expressed as an R factor (0-95 for narrowband calls and 0-120 for wideband calls)</td>
</tr>
<tr>
<td></td>
<td>RLQEstAlg</td>
<td>RLQ Est. Algorithm - name (string) of the algorithm used to estimate RLQ</td>
</tr>
<tr>
<td></td>
<td>RCQ</td>
<td>Conversational Quality R - cumulative measurement of voice quality from the start of the session to the reporting time (R is 0-95 for narrowband calls and 0-120 for wideband calls)</td>
</tr>
<tr>
<td></td>
<td>RCQEstAlg</td>
<td>RCQ Est. Algorithm - name (string) of the algorithm used to estimate RCQ</td>
</tr>
<tr>
<td></td>
<td>EXTRI</td>
<td>External R In - voice quality as measured by the local endpoint for incoming connection on &quot;other&quot; side (R is 0-95 for narrowband calls and 0-120 for wideband calls)</td>
</tr>
<tr>
<td></td>
<td>ExtRIEstAlg</td>
<td>Ext. R In Est. Algorithm - name (string) of the algorithm used to estimate EXTRI</td>
</tr>
<tr>
<td></td>
<td>EXTRO</td>
<td>External R Out - value is copied from RTCP XR received from the remote endpoint on the &quot;other&quot; side of this endpoint (R is 0-95 for narrowband calls and 0-120 for wideband calls)</td>
</tr>
<tr>
<td></td>
<td>ExtROEstAlg</td>
<td>Ext. R Out Est. Algorithm - name (string) of the algorithm used to estimate EXTRO</td>
</tr>
<tr>
<td></td>
<td>MOSLQ</td>
<td>MOS-LQ - estimated mean opinion score for listening voice quality on a scale from 1 to 5, in which 5 represents excellent and 1 represents unacceptable</td>
</tr>
<tr>
<td></td>
<td>MOSLQEstAlg</td>
<td>MOS-LQ Est. Algorithm - name (string) of the algorithm used to estimate MOSLQ</td>
</tr>
<tr>
<td></td>
<td>MOSCQ</td>
<td>MOS-CQ - estimated mean opinion score for conversation voice quality on a scale from 1 to 5, in which 5 represents excellent and 1 represents unacceptable</td>
</tr>
<tr>
<td></td>
<td>MOSCQEstAlg</td>
<td>MOS-CQ Est. Algorithm - name (string) of the algorithm used to estimate MOSCQ</td>
</tr>
<tr>
<td></td>
<td>QoEEstAlg</td>
<td>QoE Est. Algorithm - name (string) of the algorithm used to estimate all voice quality metrics</td>
</tr>
</tbody>
</table>
Below shows an example of a SIP PUBLISH message sent with RTCP XR and QoE information:

```
PUBLISH sip:172.17.116.201 SIP/2.0
Via: SIP/2.0/UDP 172.17.116.201:5060;branch=z9hG4bKac2055925925
Max-Forwards: 70
From: <sip:172.17.116.201>;tag=1c2055916574
To: <sip:172.17.116.201>
Call-ID: 20559160721612201520952@172.17.116.201
CSeq: 1 PUBLISH
Contact: <sip:172.17.116.201:5060>
Allow:
REGISTER,OPTIONS,INVITE,ACK,CANCEL,BYE,NOTIFY,PRACK,REFER,INFO,SUBSCRIBE,UPDATE
Event: vq-rtcpxr
Expires: 3600
User-Agent: device/v.7.20A.000.038
Content-Type: application/vq-rtcpxr
Content-Length: 1066
VQSessionReport
CallID=20328634741612201520943@172.17.116.201
LocalID: <sip:1000@172.17.116.201>
RemoteID: <sip:2000@172.17.116.202;user=phone>
OrigID: <sip:1000@172.17.116.201>
LocalAddr: IP=172.17.116.201 Port=6000 SSRC=0x54c62a13
RemoteAddr: IP=172.17.116.202 Port=6000 SSRC=0x243220dd
LocalGroup: 
RemoteGroup: 
LocalMetrics:
LocalMAC: 00:90:8f:57:d9:71
LocalTimestamps: START=2015-12-16T20:09:45Z STOP=2015-12-16T20:09:52Z
SessionDescription: PT=8 PD=PCMA SR=8000 FD=20 PLC=3 SSUP=Off
JitterBuffer: JBA=3 JBR=0 JBM=7 JBX=300
PacketLoss: NLR=0.00 JDR=0.00
BurstGapLoss: BLD=0.00 BD=0 GLD=0.00 GD=6325 GMIN=16
Delay: RTD=0 ESD=11
QualityEst: RLQ=93 MOSLQ=4.1
MOSCQ=4.10
RemoteMetrics:
RemoteTimestamps: START=2015-12-16T20:09:45Z STOP=2015-12-16T20:09:52Z
JitterBuffer: JBA=3 JBR=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=65535 ESD=0
QualityEst:
DialogID: 20328634741612201520943@172.17.116.201;tag=1c1690611502;from-tag=1c2032864069```

To configure RTCP XR:

1. Open the RTP/RTCP Settings page (Setup menu > Signaling & Media tab > Media folder > RTP/RTCP Settings).
2. Under the RTCP-XR group, configure the following:
   - 'Enable RTCP XR' (VQMonEnable) - enables voice quality monitoring and RTCP XR.
   - '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.
   - '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).

   ![Figure 57-1: Enabling and Configuring RTCP XR](image)

3. Under the RTCP-XR Collection Server group, configure the following:
   - 'Publication IP Group ID' (PublicationIPGroupID): Configures the IP Group to where you want the device to send RTCP XR reports.
   - '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).

   ![Figure 57-2: Configuring RTCP XR Collection Server](image)

4. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.
57.2 Generating Call Detail Records

Call Detail Records (CDR) contains vital statistic information on calls made from the device. The device can generate and report CDRs at various stages of the call - end of call, or only at the start and end of call. In addition, CDRs can be generated for SIP signaling and/or media. The device can send CDRs to any of the following:

- Syslog server. The CDR Syslog message complies with RFC 3164 and is identified by Facility 17 (local1) and Severity 6 (Informational).
- RADIUS server. For CDR in RADIUS format, see "Configuring RADIUS Accounting" on page 777. To configure RADIUS servers for CDR reporting, see "Configuring RADIUS Servers" on page 228.

Note: To view SBC CDRs stored on the device’s memory, see Viewing SBC CDR History on page 753.

57.2.1 CDR Field Description

This section describes the default CDR fields that are generated by the device.

Note: You can customize the default CDR fields if desired. For customizing SBC-related CDRs, see Customizing CDRs for SBC Calls on page 770.

57.2.1.1 CDR Fields for SBC Signaling

The default CDR fields for SBC signaling are listed in the following table.

A typical SBC session consists of two SBC legs. Each leg generates its own signaling CDRs. Each leg generates three different CDR types (SBCReportType), which are sent to the CDR server at different stages of the SIP dialog:

- "CALL_START": CDR is sent upon an INVITE message.
- "CALL_CONNECT": CDR is sent upon a 200 OK response (i.e., call is established).
- "CALL_END": CDR is sent upon a BYE message (i.e., call ends)

The CDR types for SBC signaling and the SIP dialog stages are shown in the following figure:

Figure 57-3: CDR Types for SBC Signaling
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 Leg ID (LegId 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:

- Leg ID (LegId CDR field)
- Source URI (SrcURI CDR field)
- Destination URI (DstURI CDR field)
- Call originator (Orig CDR field) - indicates the call direction (caller)
- Call duration (Durat CDR field) - call duration (elapsed time) from call connect
- Call time is based on SetupTime, ConnectTime and ReleaseTime CDR fields

### Table 57-2: Default CDR Fields for SBC Signaling

<table>
<thead>
<tr>
<th>CDR Field</th>
<th>Description</th>
<th>CDR Report Type (SBCReportType)</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SBCReportType</strong></td>
<td>Report type:</td>
<td></td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>&quot;CALL_START&quot;: CDR sent upon an INVITE message.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;CALL_CONNECT&quot;: CDR sent upon a 200 OK response.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;CALL_END&quot;: CDR sent upon a BYE message.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;DIALOG_START&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;DIALOG_END&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EPTyp</strong></td>
<td>Endpoint type: &quot;SBC&quot;</td>
<td>All</td>
<td>String</td>
</tr>
<tr>
<td><strong>SIPMethod</strong></td>
<td>SIP message type</td>
<td>All</td>
<td>String (up to 10 characters)</td>
</tr>
<tr>
<td><strong>SIPCallId</strong></td>
<td>Unique ID of call</td>
<td>All</td>
<td>String (up to 50 characters)</td>
</tr>
<tr>
<td><strong>SessionId</strong></td>
<td>Unique Session ID</td>
<td>All</td>
<td>String (up to 10 characters)</td>
</tr>
<tr>
<td><strong>LegId</strong></td>
<td>Unique ID number of the call leg within a specific call session.</td>
<td>&quot;CALL_START&quot;, &quot;CALL_CONNECT&quot;, &quot;CALL_END&quot;</td>
<td>String (decimal)</td>
</tr>
</tbody>
</table>

A basic call consists of two legs (incoming leg and outgoing leg) and thus, two leg IDs are generated for the session, one for each leg. For each new call, the device assigns leg ID "1" to the first leg. The device then increments the leg ID for subsequent legs according to the leg sequence in the call session. For example, the device generates leg ID "1" for the incoming leg and leg ID "2" for the outgoing leg. If the call is transferred, the device generates leg ID "3" for the leg belonging to the call transfer target. Another example is a call forking session where the leg ID sequence may be as follows: incoming leg is "1", outgoing leg to user's office phone...
<table>
<thead>
<tr>
<th>CDR Field</th>
<th>Description</th>
<th>CDR Report Type (SBCReportType)</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orig</td>
<td>Call originator:</td>
<td>All</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>• &quot;LCL&quot;: local</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;RMT&quot;: remote</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SourceIP</td>
<td>Source IP address</td>
<td>All</td>
<td>String (up to 20 characters)</td>
</tr>
<tr>
<td>SourcePort</td>
<td>Source UDP port</td>
<td>All</td>
<td>String (up to 10 characters)</td>
</tr>
<tr>
<td>DestIP</td>
<td>Destination IP address</td>
<td>All</td>
<td>String (up to 20 characters)</td>
</tr>
<tr>
<td>DestPort</td>
<td>Destination UDP port</td>
<td>All</td>
<td>String (up to 10 characters)</td>
</tr>
<tr>
<td>TransportType</td>
<td>Transport type:</td>
<td>All</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>• &quot;UDP&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;TCP&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;TLS&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SrcURI</td>
<td>Source URI</td>
<td>All</td>
<td>String (up to 41 characters)</td>
</tr>
<tr>
<td>SrcURIBeforeMap</td>
<td>Source URI before manipulation</td>
<td>All</td>
<td>String (up to 41 characters)</td>
</tr>
<tr>
<td>DstURI</td>
<td>Destination URI</td>
<td>All</td>
<td>String (up to 41 characters)</td>
</tr>
<tr>
<td>DstURIBeforeMap</td>
<td>Destination URI before manipulation</td>
<td>All</td>
<td>String (up to 41 characters)</td>
</tr>
<tr>
<td>Durat</td>
<td>Call duration (in seconds)</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 5 characters)</td>
</tr>
<tr>
<td>TrmSd</td>
<td>Termination side:</td>
<td>&quot;CALL_END&quot;</td>
<td>String</td>
</tr>
<tr>
<td></td>
<td>• &quot;LCL&quot;: local</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• &quot;RMT&quot;: remote</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TrmReason</td>
<td>Termination reason</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 40 characters)</td>
</tr>
<tr>
<td>TrmReasonCategory</td>
<td>Termination reason category:</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 17 characters)</td>
</tr>
<tr>
<td></td>
<td>Calls with duration 0 (i.e., not connected):</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• NO_ANSWER:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ &quot;GWAPP_NORMAL_CALL_CLEAR&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ &quot;GWAPP_NO_USER_RESPONDING&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ &quot;GWAPP_NO_ANSWER_FROM_USER_ALERTED&quot;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>• BUSY:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>✓ &quot;GWAPP_USER_BUSY&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
57. Reporting Information to External Party

<table>
<thead>
<tr>
<th>CDR Field</th>
<th>Description</th>
<th>CDR Report Type (SBCReportType)</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetupTime</td>
<td>Call setup time</td>
<td>All</td>
<td>String (up to 35 characters)</td>
</tr>
<tr>
<td>ConnectTime</td>
<td>Call connect time</td>
<td>&quot;CALL_CONNECT&quot; and &quot;CALL_END&quot;</td>
<td>String (up to 35 characters)</td>
</tr>
<tr>
<td>ReleaseTime</td>
<td>Call release time</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 35 characters)</td>
</tr>
<tr>
<td>RedirectReason</td>
<td>Redirect reason</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 15 characters)</td>
</tr>
<tr>
<td>RedirectURINum</td>
<td>Redirection URI</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 41 characters)</td>
</tr>
<tr>
<td>RedirectURINumBeforeMap</td>
<td>Redirect URI number before manipulation</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 41 characters)</td>
</tr>
<tr>
<td>TxSigIPDiffServ</td>
<td>Signaling IP DiffServ</td>
<td>All</td>
<td>String (up to 15 characters)</td>
</tr>
<tr>
<td>IGroup</td>
<td>IP Group ID and name</td>
<td>All</td>
<td>String (up to 40 characters)</td>
</tr>
<tr>
<td>Srid</td>
<td>SRD ID and name</td>
<td>All</td>
<td>String (up to 29 characters)</td>
</tr>
<tr>
<td>SIPInterfaceId</td>
<td>SIP Interface ID</td>
<td>All</td>
<td>String (up to 15 characters)</td>
</tr>
</tbody>
</table>

- **NO_RESOURCES:**
  - "GWAPP_RESOURCE_UNAVAILABLE_UNSPECIFIED"
  - "RELEASE_BECAUSE_NO_CONFERENCE_RESOURCES_LEFT"
  - "RESOURCE_BECAUSE_NO_TRUNCATING_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.

- **Calls with duration:**
  - "NORMAL_CALL_CLEAR:"
  - "GWAPP_NORMAL_CALL_CLEAR"
  - "ABNORMALLY_TERMINATED": Anything else

- **N/A:** Reasons not belonging to above categories.
<table>
<thead>
<tr>
<th>CDR Field</th>
<th>Description</th>
<th>CDR Report Type (SBCReportType)</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProxySetId</td>
<td>Proxy Set ID</td>
<td>All</td>
<td>String (up to 15 characters)</td>
</tr>
<tr>
<td>IpProfileId</td>
<td>IP Profile ID and name</td>
<td>All</td>
<td>String (up to 34 characters)</td>
</tr>
<tr>
<td>MediaRealmId</td>
<td>Media Realm ID and name</td>
<td>All</td>
<td>String (up to 55 characters)</td>
</tr>
<tr>
<td>DirectMedia</td>
<td>Direct media or traversing SBC:</td>
<td>All</td>
<td>String</td>
</tr>
<tr>
<td>SIPTrmReason</td>
<td>SIP call termination reason (BYE, CANCEL, or SIP error codes, e.g., 404)</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 12 characters)</td>
</tr>
<tr>
<td>SipTermDesc</td>
<td>Description of SIP termination reason:</td>
<td>&quot;CALL_END&quot;</td>
<td>String (up to 26 characters)</td>
</tr>
<tr>
<td>Caller</td>
<td>Name of caller</td>
<td>All</td>
<td>String (up to 36 characters)</td>
</tr>
<tr>
<td>Callee</td>
<td>Name of called party</td>
<td>All</td>
<td>String (up to 36 characters)</td>
</tr>
</tbody>
</table>

Below shows an example of an SBC signaling CDR sent at the end of a call (call was terminated normally):

```
[S=40] | SBCReportType | EPTyp | SIPCallId | SessionId | Orig | SourceIp | SourcePort | DestIp | DestPort | TransportType | SrcURI | SrcURIBeforeMap | DstURI | DstURIBeforeMap | Durat | TrmSd | TrmReason | TrmReasonCategory | SetupTime | ConnectTime | ReleaseTime | RedirectReason | RedirectURINum | RedirectURINumBeforeMap | TxSigIPDiffServ | IGroup (description) | SrId (name) | SIPInterfaceId | ProxySetId | IpProfileId (name) | MediaRealmId (name) | DirectMedia | SIPTrmReason | SIPTermDesc | Caller | Callee |
[S=41] | CALL_END | SBC | 20767593291410201017029@10.33.45.80 | 1871197419 | LCL | 10.33.45.80 | 5060 | 10.33.45.72 | 5060 | UDP | 9001@10.8.8.10 | 9001@10.8.8.10 | 6001@10.33.45.80 | 6001@10.33.45.80 | 15 | LCL | GWAPP_NORMAL_CALL_CLEAR | NORMAL_CALL_CLEAR | 17:00:29.954 | UTC Thu Oct 14 2014
```
57.2.1.2 CDR Fields for SBC Media

The default CDR fields for SBC media are listed in the following table. 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.

There are three different CDR types (SBCReportType), which are sent to the CDR server at different stages of the SIP dialog session:

- "MEDIA_START": CDR is sent upon an INVITE message.
- "UPDATE": CDR is sent upon a re-INVITE message (e.g., the established call is placed on hold by one of the call parties).
- "END": CDR is sent upon a BYE message (i.e., call ends)

The CDR types for SBC media and the SIP dialog stages are shown in the following figure:

Figure 57-4: SBC CDR Types for Media

<table>
<thead>
<tr>
<th>Table 57-3: Default CDR Fields for SBC Media</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CDR Field</strong></td>
</tr>
</tbody>
</table>
| MediaReportType | | Report type:  
| | | - "MEDIA_START": CDR is sent upon  
| | | 200 OK response or early media  
| | | - "UPDATE": CDR is sent upon a re- 
| | | INVITE message  
| | | - "END": CDR sent is upon a BYE  
| | | message |
| SIPCallId | Unique call ID |
| LegId | Unique ID number of the call leg within a  
| | specific call session. The field is  
| | included in all Report Types  
| | (MediaReportType).  
| | A basic call consists of two legs  
| | (incoming leg and outgoing leg) and  
| | thus, two leg IDs are generated for the |
session, one for each leg. For each new call, the device assigns leg ID "1" to the first leg. The device then increments the leg ID for subsequent legs according to the leg sequence in the call session. For example, the device generates leg ID "1" for the incoming leg and leg ID "2" for the outgoing leg. If the call is transferred, the device generates leg ID "3" for the leg belonging to the call transfer target. Another example is a call forking session where the leg ID sequence may be as follows: incoming leg is "1", outgoing leg to user's office phone is "2" and outgoing leg to the user's mobile phone is "3". If the call is then transferred, the leg ID for the transfer leg is "4".

<table>
<thead>
<tr>
<th>CDR Field</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cid</td>
<td></td>
<td>Channel CID</td>
</tr>
<tr>
<td>MediaType</td>
<td></td>
<td>Media type (audio, video, or text)</td>
</tr>
<tr>
<td>Coder</td>
<td></td>
<td>Coder name</td>
</tr>
<tr>
<td>PacketInterval</td>
<td>10 to 200 ms</td>
<td>Coder packet interval</td>
</tr>
<tr>
<td>LocalRtpIp</td>
<td></td>
<td>Local RTP IP address</td>
</tr>
<tr>
<td>LocalRtpPort</td>
<td>0 to 0xFFFF</td>
<td>Local RTP port</td>
</tr>
<tr>
<td>RemoteRtpIp</td>
<td></td>
<td>Remote RTP IP address</td>
</tr>
<tr>
<td>RemoteRtpPort</td>
<td>0 to 0xFFFF</td>
<td>Remote RTP port</td>
</tr>
<tr>
<td>InPackets</td>
<td>0 to 0xffffffff</td>
<td>Number of packets received by the device (local)</td>
</tr>
<tr>
<td>OutPackets</td>
<td>0 to 0xffffffff</td>
<td>Number of packets sent by the device (local)</td>
</tr>
<tr>
<td>LocalPackLoss</td>
<td>0 to 0xffffffff</td>
<td>Number of packet loss of the entire stream (local)</td>
</tr>
<tr>
<td>RemotePackLoss</td>
<td>0 to 0xffffffff</td>
<td>Remote packet loss</td>
</tr>
<tr>
<td>RTPdelay</td>
<td>0 to 10000 ms</td>
<td>Average RTP delay of the entire stream</td>
</tr>
<tr>
<td>RTPjitter</td>
<td>0 to 40000 samples</td>
<td>RTP jitter</td>
</tr>
<tr>
<td>TxRTPssrc</td>
<td>0 to 0xffffffff</td>
<td>Tx RTP SSRC</td>
</tr>
<tr>
<td>RxRTPssrc</td>
<td>0 to 0xffffffff</td>
<td>Local RTP SSRC</td>
</tr>
</tbody>
</table>
| LocalRFactor     | 0 to 120 (127 if information is unavailable) | Local conversation quality  
Note: If the RTCP XR feature is unavailable (not licensed or disabled), this R-factor VoIP metric is not provided. Instead, the device sends the CDR field with the value 127, meaning that |
### 57. Reporting Information to External Party

<table>
<thead>
<tr>
<th>CDR Field</th>
<th>Range</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoteRFactor</td>
<td>0 to 120 (127 if information is unavailable)</td>
<td>Remote conversation quality&lt;br&gt;&lt;br&gt;&lt;strong&gt;Note:&lt;/strong&gt; If the RTCP XR feature is unavailable (not licensed or disabled), this R-factor VoIP metric is not provided. Instead, the device sends the CDR field with the value 127, meaning that information is unavailable.</td>
</tr>
<tr>
<td>LocalMosCQ</td>
<td>10 to 46 (127 if information is unavailable)</td>
<td>Local MOS for conversation</td>
</tr>
<tr>
<td>RemoteMosCQ</td>
<td>10 to 46 (127 if information is unavailable)</td>
<td>Remote MOS for conversation</td>
</tr>
<tr>
<td>TxRTPIPDiffServ</td>
<td>0 to 63</td>
<td>Media IP DiffServ</td>
</tr>
<tr>
<td>LatchedRtpIp</td>
<td></td>
<td>Remote IP address of the incoming RTP stream that the device &quot;latched&quot; on to as a result of the RTP latching mechanism for NAT traversal.</td>
</tr>
<tr>
<td>LatchedRtpPort</td>
<td>0 to 0xFFFF</td>
<td>Remote RTP port of the incoming RTP stream that the device &quot;latched&quot; on to as a result of the RTP latching mechanism for NAT traversal.</td>
</tr>
<tr>
<td>LatchedT38Ip</td>
<td></td>
<td>Latching of a new T.38 stream - new IP address</td>
</tr>
<tr>
<td>LatchedT38Port</td>
<td>0 to 0xFFFF</td>
<td>Latching of a new T.38 stream - new port</td>
</tr>
</tbody>
</table>

#### 57.2.1.3 CDR Fields for SBC Local Storage

The CDR fields for SBC calls that are stored locally (history) on the device are listed in the table below. For storing CDRs locally, see “Storing CDRs on the Device“ on page 775.

<table>
<thead>
<tr>
<th>CDR Field</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Report Type</td>
<td>SBCReportType</td>
</tr>
<tr>
<td>Endpoint Type</td>
<td>EPTyp</td>
</tr>
<tr>
<td>Call Id</td>
<td>SIPCallld</td>
</tr>
<tr>
<td>Session ID</td>
<td>SessionId</td>
</tr>
<tr>
<td>Leg ID</td>
<td>Legld</td>
</tr>
<tr>
<td>Call Orig</td>
<td>Orig</td>
</tr>
<tr>
<td>Source IP</td>
<td>Sourcelp</td>
</tr>
<tr>
<td>Source Port</td>
<td>SourcePort</td>
</tr>
<tr>
<td>Destination IP</td>
<td>Destlp</td>
</tr>
<tr>
<td>CDR Field</td>
<td>Title</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>Destination Port</td>
<td>DestPort</td>
</tr>
<tr>
<td>Transport Type</td>
<td>TransportType</td>
</tr>
<tr>
<td>Source URI</td>
<td>SrcURI</td>
</tr>
<tr>
<td>Source URI Before Manipulation</td>
<td>SrcURIBeforeMap</td>
</tr>
<tr>
<td>Destination URI</td>
<td>DstURI</td>
</tr>
<tr>
<td>Destination URI Before Manipulation</td>
<td>DstURIBeforeMap</td>
</tr>
<tr>
<td>Call Duration</td>
<td>Durat</td>
</tr>
<tr>
<td>Termination Side</td>
<td>TrmSd</td>
</tr>
<tr>
<td>Termination Reason</td>
<td>TrmReason</td>
</tr>
<tr>
<td>Termination Reason Category</td>
<td>TrmReasonCategory</td>
</tr>
<tr>
<td>Setup Time</td>
<td>SetupTime</td>
</tr>
<tr>
<td>Connect Time</td>
<td>ConnectTime</td>
</tr>
<tr>
<td>Release Time</td>
<td>ReleaseTime</td>
</tr>
<tr>
<td>Redirect Reason</td>
<td>RedirectReason</td>
</tr>
<tr>
<td>Redirect URI</td>
<td>RedirectURINum</td>
</tr>
<tr>
<td>Redirect URI Before Manipulation</td>
<td>RedirectURINumBeforeMap</td>
</tr>
<tr>
<td>Signaling IP DiffServ</td>
<td>TxSigIPDiffServ</td>
</tr>
<tr>
<td>IP Group Name</td>
<td>IPGroup</td>
</tr>
<tr>
<td>SRD Name</td>
<td>SrdId</td>
</tr>
<tr>
<td>SIP Interface Name</td>
<td>SIPInterfaceId</td>
</tr>
<tr>
<td>Proxy Set Name</td>
<td>ProxySetId</td>
</tr>
<tr>
<td>IP Profile Name</td>
<td>IpProfileId</td>
</tr>
<tr>
<td>Media Realm Name</td>
<td>MediaRealmId</td>
</tr>
<tr>
<td>Direct Media</td>
<td>DirectMedia</td>
</tr>
<tr>
<td>SIP Termination Reason</td>
<td>SIPTrmReason</td>
</tr>
<tr>
<td>SIP Termination Description</td>
<td>SIPTermDesc</td>
</tr>
<tr>
<td>Caller Display ID</td>
<td>Caller</td>
</tr>
<tr>
<td>Callee Display ID</td>
<td>Callee</td>
</tr>
</tbody>
</table>

### 57.2.2 Customizing CDRs for SBC Calls

The SBC CDR Format table lets you customize SBC-related CDRs that are generated by the device for the following CDR types:

- CDRs (media and SIP signaling) sent in Syslog messages. For CDRs sent in Syslog messages, you can customize the name of the CDR field. The table lets you configure up to 128 Syslog CDR customization rules.

- CDRs related to RADIUS accounting and sent in RADIUS accounting request messages. For RADIUS accounting CDRs, you can customize the RADIUS Attribute’s
prefix name and RADIUS Attribute’s ID, for standard RADIUS Attributes and vendor-specific RADIUS Attributes (VSA). For example, instead of the default VSA name, "h323-connect-time" with RADIUS Attribute ID 28, you can change the name to "Call-Connect-Time" with ID 29. The table lets you configure up to 128 RADIUS-accounting CDR customization rules. For more information on RADIUS accounting, see "Configuring RADIUS Accounting" on page 777.

- CDRs stored locally on the device. For local storage of CDRs, you can customize the name of the CDR field. The table lets you configure up to 64 locally-stored CDR customization rules. For more information on storing CDRs on the device, see Storing CDRs on the Device on page 775.

If you do not configure a CDR customization rule for a specific CDR, the device generates the CDR in a predefined default CDR format (see "CDR Field Description" on page 762).

Note:

- The following standard RADIUS Attributes cannot be customized: 1 through 6, 18 through 20, 22, 23, 27 through 29, 32, 34 through 39, 41, 44, 52, 53, 55, 60 through 85, 88, 90, and 91.
- If the RTCP XR feature is unavailable (not licensed or disabled), the R-factor VoIP metrics are not provided in CDRs (CDR fields, Local R Factor and Remote R Factor) generated by the device. Instead, these CDR fields are sent with the value 127, meaning that information is unavailable.
- The maximum RADIUS packet size is 4,096 bytes (RFC 2865). If the packet size is greater than this due to the inclusion of many CDR fields with long customized title strings, the device removes the last CDR fields and sends the Accounting-Request packet with the CDRs that meet the packet size restriction. The removed CDR fields can be seen in the Syslog.

The following procedure describes how to customize SBC-related CDRs through the Web interface. You can also configure it through ini file (SBCCDRFormat) or CLI (configure troubleshoot > cdr > cdr-format sbc-cdr-format).

➢ To customize SBC-related CDRs:

1. Open the SBC CDR Format table (Troubleshoot menu > Troubleshoot tab > Call Detail Record folder > SBC CDR Format).
2. Click **New**; the following dialog box appears:

   **Figure 57-5: SBC CDR Format Table - Add Dialog Box**

   ![SBC CDR Format Table - Add Dialog Box](image)

3. Configure the CDR according to the parameters described in the table below.

4. Click **Apply**.

   Examples of configured CDR customization rules are shown below:

   **Figure 57-6: Examples of SBC CDR Customization Rules**

   ![Examples of SBC CDR Customization Rules](image)

   **Table 57-5: SBC CDR Format Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[SBCCDRFormat_Index]</td>
<td><strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>CDR Type</td>
<td>Defines the application type for which you want to customize CDRs.</td>
</tr>
<tr>
<td>[SBCCDRFormat_CDRType]</td>
<td><strong>[1] Syslog SBC = (Default) Customizes CDR fields for SIP signaling-related CDRs sent in Syslog messages.</strong></td>
</tr>
<tr>
<td>Field Type</td>
<td><strong>[3] Syslog Media = Customizes CDR fields for media-related CDRs sent in Syslog messages.</strong></td>
</tr>
<tr>
<td>[SBCCDRFormat_FieldType]</td>
<td><strong>[5] Local Storage SBC = Customizes CDR fields that are stored locally on the device. Only signaling-related CDRs are stored locally on the device.</strong></td>
</tr>
<tr>
<td>Title</td>
<td><strong>[7] RADIUS SBC = Customizes CDR fields (i.e., RADIUS Attributes) for CDRs sent in RADIUS accounting request messages.</strong></td>
</tr>
<tr>
<td>[SBCCDRFormat_Template]</td>
<td>Defines the CDR field (column) that you want to customize.</td>
</tr>
<tr>
<td>Field Type</td>
<td>The applicable CDR field depends on the settings of the 'CDR Type' parameter:</td>
</tr>
<tr>
<td>[SBCCDRFormat_FieldType]</td>
<td><strong>For all types:</strong> [300] CDR Type (default); [301] Call ID; [302] Session ID; [303] Report Type; [304] Media Type; [305] Accounting Status Type; [306] H323 ID; [307] RADIUS Call ID;</td>
</tr>
</tbody>
</table>
### 57. Reporting Information to External Party

#### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[308] Blank; [309] Global Session ID; [310] Leg ID.</td>
<td></td>
</tr>
</tbody>
</table>

**Title**

**title**

[SBCCDRFormat_Title] Defines a new name for the CDR field (for Syslog or local storage) or for the RADIUS Attribute prefix name (for RADIUS accounting) that you selected in the 'Column Type' parameter.

The valid value is a string of up to 31 characters. You can configure the name to be enclosed by apostrophes (single or double). For example, if you want the CDR field name to appear as 'Phone Duration', you must configure the parameter to 'Phone Duration'. You can also configure the CDR field name with an equals (=) sign, for example "call-connect-time=".

**Note:**
### 57.2.3 Configuring CDR Reporting

To enable and configure CDR reporting, perform the following procedure. For detailed descriptions of the parameters, see "Syslog, CDR and Debug Parameters" on page 858.

➢ **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

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>▪ For VSA's that do not require a prefix name, leave the parameter undefined.</td>
<td></td>
</tr>
<tr>
<td>▪ The parameter's value is case-sensitive. For example, if you want the CDR field name to be Phone-Duration (i.e., upper case &quot;P&quot; and &quot;D&quot;).</td>
<td></td>
</tr>
<tr>
<td>RADIUS Attribute Type&lt;br&gt;<code>radius-type</code>&lt;br&gt;<code>[SBCCDRFormat_RadiusType]</code></td>
<td>Defines whether the RADIUS Attribute of the CDR field is a standard or vendor-specific attribute.</td>
</tr>
<tr>
<td>▪ [0] Standard = (Default) For standard RADIUS Attributes.</td>
<td></td>
</tr>
<tr>
<td>▪ [1] Vendor Specific = For vendor-specific RADIUS Attributes (VSA).</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> The parameter is applicable only for RADIUS accounting (i.e., 'CDR Type' parameter configured to RADIUS SBC).</td>
<td></td>
</tr>
<tr>
<td>RADIUS Attribute ID&lt;br&gt;<code>radius-id</code>&lt;br&gt;<code>[SBCCDRFormat_RadiusID]</code></td>
<td>Defines an ID for the RADIUS Attribute. For VSAs, this represents the VSA ID; for standard Attributes, this represents the Attribute ID (first byte of the Attribute). The valid value is 0 to 255 (one byte). The default is 0.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td>▪ The parameter is applicable only for RADIUS accounting (i.e., 'CDR Type' parameter configured to RADIUS SBC).</td>
<td></td>
</tr>
<tr>
<td>▪ For VSA's (i.e., 'RADIUS Attribute Type' parameter configured to Vendor Specific), the parameter must be configured to any value other than 0.</td>
<td></td>
</tr>
<tr>
<td>▪ For standard RADIUS Attributes (i.e., 'RADIUS Attribute Type' parameter configured to Standard), the value must be a &quot;known&quot; RADIUS ID (per RFC for RADIUS). However, if you configure the ID to 0 (default) for any of the RADIUS Attributes (configured in the 'Column Type' parameter) listed below and then apply your rule (Click Apply), the device automatically replaces the value with the RADIUS Attribute's ID according to the RFC:</td>
<td></td>
</tr>
<tr>
<td>✓ Destination Username: 30</td>
<td></td>
</tr>
<tr>
<td>✓ Source Username: 31</td>
<td></td>
</tr>
<tr>
<td>✓ Accounting Status Type: 40</td>
<td></td>
</tr>
<tr>
<td>✓ Local Input Octets: 42</td>
<td></td>
</tr>
<tr>
<td>✓ Local Output Octets: 43</td>
<td></td>
</tr>
<tr>
<td>✓ Call Duration: 46</td>
<td></td>
</tr>
<tr>
<td>✓ Local Input Packets: 47</td>
<td></td>
</tr>
<tr>
<td>✓ Local Output Packets: 48</td>
<td></td>
</tr>
<tr>
<td>If you configure the value to 0 and the RADIUS Attribute is not any of the ones listed above, the configuration is invalid.</td>
<td></td>
</tr>
</tbody>
</table>
2. Open the Call Detail Record Settings page (Troubleshoot menu > Troubleshoot tab > Call Detail Record folder > Call Detail Record Settings).

![Figure 57-7: CDR Parameters in Advanced Parameters Page](image)

3. Configure the following parameters:
   - In the 'CDR Server IP Address' field (CDRSyslogServerIP), enter the IP address of the server to where you want the CDRs sent.
   - From the 'CDR Report Level' drop-down list (CDRReportLevel), select the stage of the call at which you want CDRs to be generated and sent.
   - (Applicable only to SBC) From the 'Media CDR Report Level' drop-down list (MediaCDRReportLevel), select the stage of the call at which you want CDRs to be generated and sent.
   - From the 'CDR Syslog Sequence Number' drop-down list (CDRSyslogSeqNum), enable or disable the inclusion of the sequence number (S=) in CDR Syslog messages.

4. Click **Apply**.

---

**Note:**
- If you do not configure an IP address for a CDR server, the device sends CDRs to the Syslog server, as configured in 'Enabling Syslog' on page 797.
- The device sends CDRs only for dialog-initiating INVITE messages (call start), 200 OK responses (call connect) and BYE messages (call end). For SBC calls only: If you want to enable the generation of CDRs for non-call SIP dialogs (such as SUBSCRIBE, OPTIONS, and REGISTER), use the EnableNonCallCdr parameter.

**57.2.4 Storing CDRs on the Device**

The CDRs generated by the device can also be stored locally on the device (hard disk of server platform).

You can specify the calls (configuration entities) for which you wish to create CDRs and store locally. This is done using Logging Filter rules in the Logging Filters table. For example, you can configure a rule to create CDRs for traffic belonging only to IP Group 2 and store the CDRs locally.

The locally stored CDRs are saved in a comma-separated values file (*.csv), where each CDR is shown on a dedicated row. An example of a CSV file with two CDRs are shown below:

- CSV file viewed in Excel:
To view the CDR column headers corresponding to the CDR data in the CSV file, run the following CLI command:

**SBC CDRs:**
```plaintext
(config-system)# cdr
(cdr)# cdr-format show-title local-storage-sbc
session id,report type,call duration, call end time, call connect time,call start time, call originator, termination reason, call id, srce uri, dest uri
```

You can do the following with locally saved CDR files (*.csv), through the CLI (root menu):

**View stored CDR files:**
- View all stored CDR files:
  ```plaintext
  # show storage-history
  ```
- View all stored, unused CDR files:
  ```plaintext
  # show storage-history unused
  ```

**Delete stored CDR files:**
- Delete all stored files:
  ```plaintext
  # clear storage-history cdr-storage-history all
  ```
- Delete all stored, unused CDR files:
  ```plaintext
  # clear storage-history cdr-storage-history unused
  ```

**Save stored CDR files to an external destination:**
```plaintext
# copy storage-history cdr-storage-history <filename> to <protocol://destination>
```
Where:
- **filename:** name you want to assign the file. Any file extension name can be used, but as the file content is in CSV format, it is recommended to use the .csv file extension.
- **protocol:** protocol over which the file is sent (tftp, http, or https).

For example:
```plaintext
# copy storage-history cdr-storage-history my_cdrs.csv to tftp://company.com/cdrs
```

The following procedure describes how to configure local CDR storage through the Web interface.

**To configure local CDR storage:**

1. Open the Call Detail Record Settings page (Troubleshoot menu > Troubleshoot tab > Call Detail Record folder > Call Detail Record Settings).
2. Configure the following parameters:
   - 'Local Storage Max File Size' (CDRLocalMaxFileSize): Enter the maximum size (in kilobytes) of the CDR file. Once the file size is reached, the device creates a new file for subsequent CDRs, and so on.
57. Reporting Information to External Party

- 'Local Storage Max Number of Files' (CDRLocalMaxNomOfFiles): Enter the maximum number of CDR files. Once the maximum is reached, a subsequent CDR file replaces the oldest created file.
- 'Local Storage File Creation Interval' (CDRLocalInterval): Enter the time (in minutes) for how often the device creates a new CDR file. For example, if configured to 60, it creates a new file every hour even if the maximum file size has not yet been reached.
- For a detailed description of each parameter, see "Syslog, CDR and Debug Parameters" on page 858.

3. Open the Logging Filters table (see "Configuring Log Filter Rules" on page 787), and then configure a log filtering rule with the following settings:
   - 'Filter Type' and 'Value': (as desired)
   - 'Log Destination': Local Storage
   - 'Log Type': CDR Only
   - 'Mode': Enable

**Note:**
- If you have enabled the CDR storage feature and you later decide to change the maximum number of files (CDRLocalMaxNomOfFiles) to a lower value (e.g., from 50 to 10), the device stores the remaining files (e.g., 40) in its memory (i.e., unused files).
- When the device operates in High-Availability mode, stored CDRs are deleted upon device switchover.
- For customizing CDR fields for SBC calls, see Customizing CDRs for SBC Calls on page 770.

57.3 Configuring RADIUS Accounting

The device can send accounting data of SIP calls as call detail records (CDR) to a RADIUS Accounting server. CDR-based accounting messages can be sent upon call release, call connection and release, or call setup and release. This section lists the CDR attributes for RADIUS accounting.

The following figure shows the interface between the device and the RADIUS server, based on the RADIUS Accounting protocol. For each CDR that the device sends to the RADIUS server, it sends an Accounting-Request Stop with all the CDR attributes. When the RADIUS server successfully receives all the CDR attributes, it responds with an Accounting-Response Stop ACK message to the device. If the device does not receive the Accounting-Response ACK message, it can resend the Accounting-Request Stop with all CDR attributes again, up to a user-defined number of re-tries (see "Configuring RADIUS Packet Retransmission" on page 230).
There are two types of data that can be sent to the RADIUS server. The first type is the accounting-related attributes and the second type is the vendor specific attributes (VSA):

- **Standard RADIUS attributes (per RFC):** A typical standard RADIUS attribute is shown below. The RADIUS attribute ID depends on the attribute.

  ![Figure 57-9: Typical Standard RADIUS Attribute](image)

  The following figure shows a standard RADIUS attribute collected by Wireshark. The bottom pane shows the RADIUS attribute information as sent in the packet; the upper pane is Wireshark's interpretation of the RADIUS information in a more readable format. The example shows the attribute in numeric format (32-bit number in 4 bytes).

  ![Figure 57-10: Example of Standard RADIUS Attribute Collected by Wireshark](image)

- **Vendor-specific RADIUS attributes:** RADIUS attributes that are specific to the device (company) are referred to as Vendor-specific attributes (VSA). The CDR of VSAs are sent with a general RADIUS ID of 26 to indicate that they are vendor-specific (non-standard). In addition, the company's registered vendor ID (as registered with the Internet Assigned Numbers Authority or IANA) is also included in the packet. The device's default vendor ID is 5003, which can be changed (see "Configuring the RADIUS Vendor ID" on page 231). The VSA ID is also included in the packet.

  ![Figure 57-11: Example of a Vendor-Specific Attribute](image)

  The following figure shows a vendor-specific RADIUS attribute collected by Wireshark. The bottom pane shows the RADIUS attribute information as sent in the packet; the
upper pane is Wireshark's interpretation of the RADIUS information in a more readable format. The example shows the attribute in string-of-characters format.

Figure 57-12: Example of Vendor-Specific RADIUS Attribute Collected by Wireshark

Note: You can customize the prefix title of the RADIUS attribute name and the ID. For more information, see Customizing CDRs for SBC Calls on page 770.

To configure the address of the RADIUS Accounting server, see "Configuring RADIUS Servers" on page 228. For all RADIUS-related configuration, see "RADIUS-based Services" on page 227.

➢ To configure RADIUS accounting:
1. Open the Call Detail Record Settings page (Troubleshoot menu > Troubleshoot tab > Call Detail Record folder > Call Detail Record Settings).
2. Configure the following parameters:
   - From the 'Enable RADIUS Access Control' drop-down list (EnableRADIUS), select Enable.
   - From the 'RADIUS Accounting Type' drop-down list (RADIUSAccountingType), select the stage of the call that RADIUS accounting messages are sent to the RADIUS accounting server.
   - From the 'AAA Indications' drop-down list (AAAIndications), select whether you want Authentication, Authorization and Accounting (AAA) indications.
   For a detailed description of the parameters, see "RADIUS Parameters" on page 940.

Figure 57-13: Configuring RADIUS Accounting

3. Click Apply, and then reset the device with a save-to-flash for your settings to take effect.

The table below lists the RADIUS Accounting CDR attributes included in the communication packets transmitted between the device and a RADIUS server.
<table>
<thead>
<tr>
<th>Attribute ID</th>
<th>Attribute Name</th>
<th>Vendor-Specific Attribute (VSA) ID</th>
<th>Description</th>
<th>Value Format</th>
<th>Example</th>
<th>AAA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request Attributes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>user-name</td>
<td>(Standard)</td>
<td>Account number or calling party number or blank</td>
<td>String</td>
<td>5421385747</td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>4</td>
<td>nas-ip-address</td>
<td>(Standard)</td>
<td>IP address of the requesting device</td>
<td>Numeric</td>
<td>192.168.14.43</td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>6</td>
<td>service-type</td>
<td>(Standard)</td>
<td>Type of service requested</td>
<td>Numeric</td>
<td>1: login</td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-incoming-conf-id</td>
<td>1</td>
<td>SIP call identifier</td>
<td>Up to 32 octets</td>
<td>h323-incoming-conf-id=38393530</td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-remote-address</td>
<td>23</td>
<td>IP address of the remote gateway</td>
<td>Numeric</td>
<td>-</td>
<td>Stop Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-conf-id</td>
<td>24</td>
<td>H.323/SIP call identifier</td>
<td>Up to 32 octets</td>
<td></td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-setup-time</td>
<td>25</td>
<td>Setup time in NTP format</td>
<td>String</td>
<td>h323-setup-time=09:33:26.621 Mon Dec 2014</td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-call-origin</td>
<td>26</td>
<td>Originator of call:</td>
<td>String</td>
<td>h323-call-origin=answer</td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-call-type</td>
<td>27</td>
<td>Protocol type or family used on this leg of the call. The value is always &quot;VOIP&quot;.</td>
<td>String</td>
<td>h323-call-type=VOIP</td>
<td>Start Acc Stop Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-connect-time</td>
<td>28</td>
<td>Connect time in NTP format</td>
<td>String</td>
<td>h323-connect-time=09:33:37.657 UTC Mon Dec 08 2015</td>
<td>Stop Acc</td>
</tr>
<tr>
<td>Attribute ID</td>
<td>Attribute Name</td>
<td>Vendor-Specific Attribute (VSA) ID</td>
<td>Description</td>
<td>Value Format</td>
<td>Example</td>
<td>AAA</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------</td>
<td>-----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
<td>----------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>26</td>
<td>h323-disconnect-time</td>
<td>29</td>
<td>Disconnect time in NTP format</td>
<td>String</td>
<td>-</td>
<td>Stop  Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-disconnect-cause</td>
<td>30</td>
<td>Disconnect cause code (Q.850)</td>
<td>Numeric</td>
<td>h323-disconnect-cause=16</td>
<td>Stop  Acc</td>
</tr>
<tr>
<td>26</td>
<td>h323-gw-id</td>
<td>33</td>
<td>Name of the gateway</td>
<td>String</td>
<td>h323-gw-id=&lt;SIP ID string&gt;</td>
<td>Start  Acc Stop  Acc</td>
</tr>
<tr>
<td>26</td>
<td>sip-call-id</td>
<td>34</td>
<td>SIP Call ID</td>
<td>String</td>
<td>sip-call-id=<a href="mailto:abcde@ac.com">abcde@ac.com</a></td>
<td>Start  Acc Stop  Acc</td>
</tr>
<tr>
<td>26</td>
<td>call-terminator</td>
<td>35</td>
<td>Terminator of the call:</td>
<td>String</td>
<td>call-terminator=yes</td>
<td>Stop  Acc</td>
</tr>
<tr>
<td>26</td>
<td>terminator</td>
<td>37</td>
<td>Terminator of the call:</td>
<td>String</td>
<td>terminator=originate</td>
<td>Stop  Acc</td>
</tr>
<tr>
<td>30</td>
<td>called-station-id</td>
<td>(Standard)</td>
<td>Destination URI</td>
<td>String</td>
<td>8004567145</td>
<td>Start  Acc</td>
</tr>
<tr>
<td>31</td>
<td>calling-station-id</td>
<td>(Standard)</td>
<td>Source URI</td>
<td>String</td>
<td>5135672127</td>
<td>Start  Acc Stop  Acc</td>
</tr>
<tr>
<td>40</td>
<td>acct-status-type</td>
<td>(Standard)</td>
<td>Account Request Type - start (1) or stop (2).</td>
<td>Numeric</td>
<td>1</td>
<td>Start  Acc Stop  Acc</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> 'start' isn't supported on the Calling Card application.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>acct-delay-time</td>
<td>(Standard)</td>
<td>No. of seconds tried in sending a particular record</td>
<td>Numeric</td>
<td>5</td>
<td>Start  Acc Stop  Acc</td>
</tr>
<tr>
<td>Attribute ID</td>
<td>Attribute Name</td>
<td>Description</td>
<td>Value Format</td>
<td>Example</td>
<td>AAA</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------------</td>
<td>------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>acct-input-octets</td>
<td>(Standard) Number of octets received for that call duration</td>
<td>Numeric</td>
<td>-</td>
<td>Stop Acc</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>acct-output-octets</td>
<td>(Standard) Number of octets sent for that call duration</td>
<td>Numeric</td>
<td>-</td>
<td>Stop Acc</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>acct-session-id</td>
<td>(Standard) A unique accounting identifier - match start &amp; stop</td>
<td>String</td>
<td>34832</td>
<td>Start AccStop Acc</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>acct-session-time</td>
<td>(Standard) For how many seconds the user received the service</td>
<td>Numeric</td>
<td>-</td>
<td>Stop Acc</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>acct-input-packets</td>
<td>(Standard) Number of packets received during the call</td>
<td>Numeric</td>
<td>-</td>
<td>Stop Acc</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>acct-output-packets</td>
<td>(Standard) Number of packets sent during the call</td>
<td>Numeric</td>
<td>-</td>
<td>Stop Acc</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>nas-port-type</td>
<td>(Standard) Physical port type of device on which the call is active</td>
<td>String</td>
<td>0: Asynchronous</td>
<td>Start AccStop Acc</td>
<td></td>
</tr>
</tbody>
</table>

**Response Attributes**

<table>
<thead>
<tr>
<th>26</th>
<th>h323-return-code</th>
<th>103 The reason for failing authentication (0 = ok, other number failed)</th>
<th>Numeric</th>
<th>0 Request accepted</th>
<th>Stop Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>44</td>
<td>acct-session-id</td>
<td>(Standard) A unique accounting identifier – match start &amp; stop</td>
<td>String</td>
<td>-</td>
<td>Stop Acc</td>
</tr>
</tbody>
</table>

Below is an example of RADIUS Accounting, where non-standard parameters are preceded with brackets:

```
Accounting-Request (4)
user-name = 111
acct-session-id = 1
nas-ip-address = 212.179.22.213
nas-port-type = 0
acct-status-type = 2
```
acct-session-time = 1
acct-input-packets = 122
acct-output-packets = 220
called-station-id = 201
calling-station-id = 202

// Accounting non-standard parameters:
(4923 33) h323-gw-id =
(4923 23) h323-remote-address = 212.179.22.214
(4923 1) h323-ivr-out = h323-incoming-conf-id:02102944 600a1899
3fd61009 0e2f3cc5
(4923 30) h323-disconnect-cause = 22 (0x16)
(4923 27) h323-call-type = VOIP
(4923 26) h323-call-origin = Originate
(4923 24) h323-conf-id = 02102944 600a1899 3fd61009 0e2f3cc5
This page is intentionally left blank.
Part XI

Diagnostics
58 Syslog and Debug Recording

For debugging and troubleshooting, you can use the device's Syslog and/or Debug Recording capabilities:

- **Syslog**: Syslog is an event notification protocol that enables a device to send event notification messages across IP networks to event message collectors, also known as Syslog servers. The device contains an embedded Syslog client, which sends error reports / events that it generates to a remote Syslog server using the IP / UDP protocol. This information is a collection of error, warning, and system messages that records every internal operation of the device.

- **Debug Recording**: The device can send debug recording packets to a debug capturing server. When the debug recording is activated, the device duplicates all messages that are sent and/or received by it and then sends them to an external server defined by IP address. The debug recording can be done for different types of traffic such as RTP/RTCP, T.38, and SIP. Debug recording is used for advanced debugging when you need to analyze internal messages and signals. Debug recording is also useful for recording network traffic in environments where hub or port mirroring is unavailable and for recording internal traffic between two endpoints on the same device.

**Note**: You can include Syslog messages in debug recording (see "Configuring Log Filter Rules" on page 787).

58.1 Configuring Log Filter Rules

The Logging Filters table lets you configure up to 60 rules for filtering debug recording packets, Syslog messages, and Call Detail Records (CDR). The log filter determines the calls for which you want to generate debug recording packets, Syslog messages or CDRs. For example, you can add a rule to generate Syslog messages only for calls belonging to IP Groups 2 and 4, or for calls belonging to all IP Groups except IP Group 3. You can also configure log filters for generating CDRs only and saving them on the device (local storage). Debug recording log filters can include signaling information (such as SIP messages), Syslog messages, CDRs, media (RTP, RTCP, and T.38), and pulse-code modulation (PCM).

If you don't configure any rules in the Logging Filters table and you have globally enabled debug recording (by configuring the Debug Recording server's address - see Note below), Syslog (global parameter - see Note below), and/or CDR generation (global parameter for enabling Syslog - see Note below), logs are generated for all calls. Thus, the benefit of log filtering is that it allows you to create logs per specific calls, eliminating the need for additional device resources (CPU consumption) otherwise required when logs are generated for all calls.

You can enable and disable configured Log Filter rules. Enabling a rule activates the rule, whereby the device starts generating the debug recording packets, Syslog messages, or CDRs. Disabling a rule is useful, for example, if you no longer require the rule, but may need it in the future. Thus, instead of deleting the rule entirely, you can simply disable it.
Note:

- If you want to configure a Log Filter rule that logs Syslog messages to a Syslog server (i.e., not to a Debug Recording server), you must enable Syslog functionality, using the 'Enable Syslog' (EnableSyslog) parameter (see "Enabling Syslog" on page 797). Enabling Syslog functionality is not required for rules that include Syslog messages in the debug recording sent to the Debug Recording server.
- To configure the Syslog server's address, see "Configuring the Syslog Server Address" on page 797. To configure additional, global Syslog settings, see "Configuring Syslog" on page 792.
- To configure the Debug Recording server's address, see "Configuring the Debug Recording Server Address" on page 802.
- To configure additional, global CDR settings such as at what stage of the call the CDR is generated (e.g., start and end of call), see "Configuring CDR Reporting" on page 774.

The following procedure describes how to configure Log Filter rules through the Web interface. You can also configure it through ini file (LoggingFilters) or CLI (configure troubleshoot > logging logging-filters).

➢ To configure a Log Filter rule:
  1. Open the Logging Filters table (Troubleshoot menu > Troubleshoot tab > Logging folder > Logging Filters).
  2. Click New; the following dialog box appears:

      Figure 58-1: Logging Filters Table - Add Dialog Box

      [Diagram of Logging Filters Table]

      3. Configure a Log Filtering rule according to the parameters described in the table below.

      4. Click Apply.

      Table 58-1: Logging Filters Table Parameter Descriptions

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row.</td>
</tr>
<tr>
<td>[LoggingFilters_index]</td>
<td>Note: Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Filter Type</td>
<td>Defines the filter type criteria.</td>
</tr>
<tr>
<td>filter-type</td>
<td>- [1] Any (default)</td>
</tr>
<tr>
<td></td>
<td>- [9] SRD = Filters log by SRD. To configure SRDs, see Configuring SRDs on page 327.</td>
</tr>
<tr>
<td></td>
<td>- [10] Classification = Filters log by Classification rule. To configure Classification rules, see Configuring Classification Rules on page 493.</td>
</tr>
<tr>
<td></td>
<td>- [12] User = Filters log by user. The user is defined by username or username@hostname in the Request-URI of the SIP Request-Line. For example, &quot;2222@10.33.45.201&quot;, which represents the following INVITE:</td>
</tr>
<tr>
<td></td>
<td>INVITE sip:2222@10.33.45.201;user=phone SIP/2.0</td>
</tr>
<tr>
<td></td>
<td>- [13] IP Trace = Filters log by an IP network trace, Wireshark-like expression. For more information, see &quot;Filtering IP Network Traces&quot; on page 791.</td>
</tr>
<tr>
<td></td>
<td>- [14] SIP Interface = Filters log by SIP Interface. To configure SIP Interfaces, see Configuring SIP Interfaces on page 337.</td>
</tr>
<tr>
<td>Value</td>
<td>Defines the value for the filtering type configured in the 'Filter Type' parameter.</td>
</tr>
<tr>
<td>value</td>
<td>The value can include the following:</td>
</tr>
<tr>
<td></td>
<td>- A single value.</td>
</tr>
<tr>
<td></td>
<td>- A range, using a hyphen &quot;-&quot; between the two values. For example, to specify IP Groups 1, 2 and 3, configure the parameter to &quot;1-3&quot; (without apostrophes).</td>
</tr>
<tr>
<td></td>
<td>- Multiple, non-contiguous values, using commas &quot;,&quot; between each value. For example, to specify IP Groups 1, 3 and 9, configure the parameter to &quot;1,3,9&quot; (without apostrophes).</td>
</tr>
<tr>
<td></td>
<td>- The exclamation (!) wildcard character can be used for excluding a specific configuration entity from the filter. For example, to include all IP Groups in the filter except IP Group ID 2, configure the 'Filter Type' parameter to IP Group and the 'Value' parameter to &quot;!2&quot; (without apostrophes). Note that a Logging Filter rule applies to the entire session, which is both legs (i.e., not per leg). For example, a call between IP Groups 1 and 2 are logged for both legs even if the 'Value' parameter is configured to &quot;!2&quot;.</td>
</tr>
<tr>
<td></td>
<td>- Any to indicate all.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can use the index number or string name to specify the configuration entity for the following 'Filter Types': IP Group, SRD, Classification, IP-to-IP Routing, or SIP Interface. For example, to specify IP Group at Index 2 with the name &quot;SIP Trunk&quot;, configure the parameter to either &quot;2&quot; or &quot;SIP Trunk&quot; (without apostrophes).</td>
</tr>
<tr>
<td></td>
<td>For IP trace expressions, see &quot;Filtering IP Network Traces&quot; on page 791.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Log Destination</td>
<td>Defines where the device sends the log file.</td>
</tr>
</tbody>
</table>
| log-dest [LoggingFilters_LogDestination] | - [0] Syslog Server = The device generates Syslog messages based on the configured log filter and sends them to a user-defined Syslog server. The Syslog messages can contain one of the following types of information, depending on the settings of the 'Log Type' parameter (described later):<br>  - Not configured (default): Syslog messages include regular syslog information.<br>  - CDR Only: Syslog messages include only CDRs (no system information and alerts).<br>- [1] Debug Recording Server = (Default) The device generates debug recording packets based on the configured log filter and sends them to a user-defined Debug Recording server.<br>- [2] Local Storage = The device generates CDRs based on the configured log filter and stores them locally on the device. For more information on local CDR storage, see Storing CDRs on the Device on page 775. 
Note: |
|   | - If the 'Filter Type' parameter is configured to IP Trace, you must configure the parameter to Debug Recording Server.<br> - If you configure the parameter to Local Storage, you must configure the 'Log Type' parameter to CDR Only.<br> - If you configure the parameter to Syslog Server and the debug level (GwDebugLevel) is configured to No Debug (see "Configuring Syslog Debug Level" on page 798), the Syslog messages include only system Warnings and Errors.<br> - If you configure the parameter to Debug Recording Server, you can also include Syslog messages in the debug recording packets sent to the debug recording server. To include Syslog messages, configure the 'Log Type' parameter (see below) to the relevant option. |
| Log Type | Defines the type of messages to include in the log file. |
| log-type [LoggingFilters_CaptureType] | - [0] = (Default) Not configured. The option is applicable only for sending Syslog messages to a Syslog server (i.e., 'Log Destination' parameter is configured to Syslog Server).<br> - [1] Signaling = The option is applicable only to debug recording (i.e., 'Log Destination' parameter is configured to Debug Recording Server). The debug recording includes signaling information such as SIP signaling messages, Syslog messages, CDRs, and the device's internal processing messages.<br> - [2] Signaling & Media = The option is applicable only to debug recording (i.e., 'Log Destination' parameter is configured to Debug Recording Server). The debug recording includes signaling, Syslog messages, and media (RTP/RTCP/T.38).<br> - [3] Signaling & Media & PCM = The option is applicable only to debug recording (i.e., 'Log Destination' parameter is configured to Debug Recording Server). The debug recording includes signaling, Syslog messages, media, and PCM.<br> - [5] CDR Only = Only CDRs are generated. The option is applicable only if the 'Log Destination' parameter is configured to Syslog Server or Local Storage. When configured to Syslog Server, the CDRs are sent to the device for local storage. |

User’s Manual 790 Document #: LTRT-41877
58. Syslog and Debug Recording

### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server</td>
<td>Server, only CDRs are included in the Syslog messages (excluding all system logs and alerts) sent to the Syslog server.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ If you configure the 'Log Destination' parameter to Local Storage, the 'Log Type' parameter must be configured to CDR Only.</td>
</tr>
<tr>
<td></td>
<td>▪ The parameter is not applicable when the 'Filter Type' parameter is configured to <strong>IP Trace</strong>.</td>
</tr>
<tr>
<td></td>
<td>▪ To include Syslog messages in debug recording, it is unnecessary to enable Syslog functionality.</td>
</tr>
<tr>
<td>Mode</td>
<td>Enables and disables the rule.</td>
</tr>
<tr>
<td>[LoggingFilters_Mode]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>▪ [0] Disable</td>
</tr>
<tr>
<td></td>
<td>▪ [1] Enable (default)</td>
</tr>
</tbody>
</table>

#### 58.1.1 Filtering IP Network Traces

You can filter Syslog and debug recording messages for IP network traces, by configuring 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>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip.src, ip.dst</td>
<td>Source and destination IP address</td>
</tr>
<tr>
<td>ip.addr</td>
<td>IP address - up to two IP addresses can be entered</td>
</tr>
<tr>
<td>ip.proto</td>
<td>IP protocol type (PDU) entered as an enumeration value (e.g., 1 is ICMP, 6 is TCP, 17 is UDP)</td>
</tr>
<tr>
<td>udp, tcp, icmp, sip, ldap, http, https</td>
<td>Single expressions for protocol type</td>
</tr>
<tr>
<td>udp.port, tcp.port</td>
<td>Transport layer</td>
</tr>
<tr>
<td>udp.srcport, tcp.srcport</td>
<td>Transport layer for source port</td>
</tr>
<tr>
<td>udp.dstport, tcp.dstport</td>
<td>Transport layer for destination port</td>
</tr>
<tr>
<td>and, &amp;&amp;, ==, &lt;, &gt;</td>
<td>Between expressions</td>
</tr>
</tbody>
</table>

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' parameter is undefined, the device records all IP traffic types.
- You cannot use ip.addr or udp/tcp.port together with ip.src/dst or udp/tcp.srcport/dstport. For example, "ip.addr==1.1.1.1 and ip.src==2.2.2.2" is an invalid configuration value.

58.2 Configuring Syslog

This section describes how to configure Syslog. To filter Syslog messages, see "Configuring Log Filter Rules" on page 787.

58.2.1 Syslog Message Format

The Syslog message is sent from the device to a Syslog server as an ASCII (American Standard Code for Information Interchange) message. Syslog uses UDP as its underlying transport layer mechanism. By default, UDP port 514 is assigned to Syslog, but this can be changed (see "Enabling Syslog" on page 797).

Syslog includes two types of log messages:
- SIP call session logs: Logs relating to call sessions (e.g., call established). These logs are identified by a session ID ("SID"), described in detail in the table below. The following is an example of a SIP-session related Syslog message:
  

- Board logs: Logs relating to the operation of the device (infrastructure) that are non-call session related (e.g., device reset or Web login). These logs are identified by a board ID ("BID"), described in detail in the table below. The following is an example of a board Syslog message:
  

The format of the Syslog message is described in the following table below:

<table>
<thead>
<tr>
<th>Message Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message Types</td>
<td>Syslog generates the following types of messages:</td>
</tr>
<tr>
<td></td>
<td>• ERROR: Indicates that a problem has been identified that requires immediate handling.</td>
</tr>
<tr>
<td></td>
<td>• WARNING: Indicates an error that might occur if measures are not taken to prevent it.</td>
</tr>
<tr>
<td></td>
<td>• NOTICE: Indicates that an unusual event has occurred.</td>
</tr>
<tr>
<td></td>
<td>• INFO: Indicates an operational message.</td>
</tr>
<tr>
<td></td>
<td>• DEBUG: Messages used for debugging.</td>
</tr>
</tbody>
</table>
### Message Item

<table>
<thead>
<tr>
<th>Note:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- The INFO and DEBUG messages are required only for advanced debugging and by default, they are not sent by the device.</td>
</tr>
<tr>
<td>- When viewing Syslog messages in the Web interface, these message types are color coded.</td>
</tr>
</tbody>
</table>

### Message Sequence Number

[S=number]

By default, Syslog messages are sequentially numbered in the format [S=number], for example, "[S=643]". A skip in the number sequence of messages indicates a loss of message packets. For example, in the below Syslog, messages 238 through 300 were not received. In other words, 63 Syslog messages were lost (the sequential numbers are indicated below in bold font):

```
```

You can disable the inclusion of the message sequence number in Syslog messages, by setting the ‘CDR Syslog Sequence Number’ parameter to **Disable** (see “Configuring Syslog” on page 797).

### Log Number

(lgr)(number)

Ignore this number; it has been replaced by the Message Sequence Number (described previously).

### Session ID (SID)

Unique SIP call session and device identifier. The device identifier facilitates debugging by clearly identifying the specific device that sent the log message, especially useful in deployments consisting of multiple devices. In addition, the benefit of unique numbering is that it enables you to filter the information (such as SIP, Syslog, and media) according to device or session ID.

The syntax of the session and device identifiers are as follows:

[SID=<unique number generated during software installation>:<number of times device has reset>:<unique SID counter indicating the call session; increments consecutively for each new session; resets to 1 after a device reset>]

For example:

```
```

Where:
**Message Item**

- 2ed1c8 is the device's MAC address.
- 96 is the number of times the device has reset.
- 5 is a unique SID session number (in other words, this is the fifth call session since the last device reset).
  - A session includes both the outgoing and incoming legs, where both legs share the same session number.
  - Forked legs and alternative legs share the same session number.

**Board ID (BID)**

Unique non-SIP session related (e.g., device reset) and device identifier. The device identifier facilitates debugging by clearly identifying the specific device that sent the log message, especially useful in deployments consisting of multiple devices. In addition, the benefit of unique numbering is that it enables you to filter the information according to device.

The syntax of the BID is as follows:

[BID=<unique number generated during software installation>:<number of times device has reset>]

For example:

14:32:52.062: 10.33.8.70: WARNING: [S=9399] [BID=2ed1c8:96] invalid Physical index

Where:
- 2ed1c8 is the device's MAC address.
- 96 is the number of times the device has reset.

**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.

---

### 58.2.1.1 Event Representation in Syslog Messages

The Syslog message events that the device sends are denoted by unique abbreviations. The following example shows an abbreviated event in a Syslog message indicating packet loss (PL):

Apr 4 12:00:12 172.30.1.14 PL:5 [Code:3a002] [CID:3294] [Time: 20:17:00]

The table below lists the unique event abbreviations:

<table>
<thead>
<tr>
<th>Error Abbreviation</th>
<th>Error Name Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA</td>
<td>Invalid Accumulated Packets Counter</td>
</tr>
<tr>
<td>AC</td>
<td>Invalid Channel ID</td>
</tr>
<tr>
<td>AL</td>
<td>Invalid Header Length</td>
</tr>
<tr>
<td>AO</td>
<td>Invalid Codec Type</td>
</tr>
<tr>
<td>Error Abbreviation</td>
<td>Error Name Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>AP</td>
<td>Unknown Aggregation Payload Type</td>
</tr>
<tr>
<td>AR</td>
<td>Invalid Routing Flag Received</td>
</tr>
<tr>
<td>AT</td>
<td>Simple Aggregation Packets Lost</td>
</tr>
<tr>
<td>CC</td>
<td>Command Checksum Error</td>
</tr>
<tr>
<td>CE</td>
<td>Invalid Cell Coder Code</td>
</tr>
<tr>
<td>CS</td>
<td>Command Sequence Error</td>
</tr>
<tr>
<td>ES</td>
<td>8 sec Timeout Before Disconnect</td>
</tr>
<tr>
<td>HO</td>
<td>Host Received Overrun</td>
</tr>
<tr>
<td>IA</td>
<td>Invalid AMR Payload</td>
</tr>
<tr>
<td>IC</td>
<td>Invalid CID Error</td>
</tr>
<tr>
<td>IG</td>
<td>Invalid G723 Code</td>
</tr>
<tr>
<td>IP</td>
<td>Invalid payload length</td>
</tr>
<tr>
<td>IR</td>
<td>Invalid RTCP Packet</td>
</tr>
<tr>
<td>IS</td>
<td>Invalid SID Length</td>
</tr>
<tr>
<td>LC</td>
<td>Transmitter Received Illegal Command</td>
</tr>
<tr>
<td>LF</td>
<td>Lost Fax Frames In High Speed Mode</td>
</tr>
<tr>
<td>LM</td>
<td>Lost Modem Frames In High Speed Mode</td>
</tr>
<tr>
<td>MI</td>
<td>Misalignment Error</td>
</tr>
<tr>
<td>MR</td>
<td>Modem Relay Is Not Supported</td>
</tr>
<tr>
<td>OR</td>
<td>DSP JB Overrun</td>
</tr>
<tr>
<td>PH</td>
<td>Packet Header Error</td>
</tr>
<tr>
<td>PL</td>
<td>RTP Packet Loss</td>
</tr>
<tr>
<td>RB</td>
<td>Counts the number of BFI Frames Received From The Host</td>
</tr>
<tr>
<td>RD</td>
<td>No Available Release Descriptor</td>
</tr>
<tr>
<td>RO</td>
<td>RTP Reorder</td>
</tr>
<tr>
<td>RP</td>
<td>Unknown RTP Payload Type</td>
</tr>
<tr>
<td>RS</td>
<td>RTP SSRC Error</td>
</tr>
<tr>
<td>UF</td>
<td>Unrecognized Fax Relay Command</td>
</tr>
<tr>
<td>AA</td>
<td>Invalid Accumulated Packets Counter</td>
</tr>
<tr>
<td>AC</td>
<td>Invalid Channel ID</td>
</tr>
<tr>
<td>AL</td>
<td>Invalid Header Length</td>
</tr>
<tr>
<td>AO</td>
<td>Invalid Codec Type</td>
</tr>
<tr>
<td>AP</td>
<td>Unknown Aggregation Payload Type</td>
</tr>
<tr>
<td>AR</td>
<td>Invalid Routing Flag Received</td>
</tr>
</tbody>
</table>
58.2.1.2 Identifying AudioCodes Syslog Messages using Facility Levels

The device’s Syslog messages can easily be identified and distinguished from Syslog messages from other equipment, by setting its Facility level. The Facility levels of the device’s Syslog messages are numerically coded with decimal values. Facility level may use any of the "local use" facilities (0 through 7), according to RFC 3164. Implementing Facility levels is useful, for example, if you collect the device’s as well as other equipments’ Syslog messages on the same server. Therefore, in addition to filtering Syslog messages by IP address, you can filter messages by Facility level.

- To configure the Facility level:
  - Configure the SyslogFacility ini file parameter to one of the following options:

<table>
<thead>
<tr>
<th>Numerical Value</th>
<th>Facility Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 (default)</td>
<td>local use 0 (local0)</td>
</tr>
<tr>
<td>17</td>
<td>local use 1 (local1)</td>
</tr>
<tr>
<td>18</td>
<td>local use 2 (local2)</td>
</tr>
<tr>
<td>19</td>
<td>local use 3 (local3)</td>
</tr>
<tr>
<td>20</td>
<td>local use 4 (local4)</td>
</tr>
<tr>
<td>21</td>
<td>local use 5 (local5)</td>
</tr>
<tr>
<td>22</td>
<td>local use 6 (local6)</td>
</tr>
<tr>
<td>23</td>
<td>local use 7 (local7)</td>
</tr>
</tbody>
</table>

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)

58.2.1.3 Syslog Fields for Answering Machine Detection (AMD)

The Syslog message can include information relating to the Answering Machine Detection (AMD) feature. AMD is used to detect whether a human (including a fax machine), an answering machine, silence, or answering machine beeps have answered the call on the remote side.

- AMDSignal – the field can acquire one of the following values:
  - voice (V)
  - answer machine (A)
  - silence (S)
  - unknown (U)

- AMDDecisionProbability – probability (in %) success that correctly detects answering type

Below is an example of such a Syslog message with AMD information:

CallMachine:EVENT_DETECTED_EV - AMDSignal = <type = V/A/S/U>,
AMDDecisionProbability = <percentage> %

If there is no AMD detection, the AMDSignal field is shown empty (i.e. AMDSignal = ).
For more information on the AMD feature, see "Answering Machine Detection (AMD)" on page 200.

### 58.2.1.4 SNMP Alarms in Syslog Messages

SNMP alerts are sent to the Syslog server using the following formats:

- **Raised Alarms**: RAISE-ALARM: <Alarm Name>; Textual Description: <Textual Description>; Severity <Alarm Severity>; Source <Alarm Source>; Unique ID: <Alarm Unique ID>.
  
  If additional information exists in the alarm, then these are also added: Additional Info1:/ Additional Info2:/ Additional Info3

  The Messages’ Severity is as follows:

<table>
<thead>
<tr>
<th>ITU Perceived Severity (SNMP Alarm’s Severity)</th>
<th>AudioCodes’ Syslog Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Critical</td>
<td>RecoverableMsg</td>
</tr>
<tr>
<td>Major</td>
<td>RecoverableMsg</td>
</tr>
<tr>
<td>Minor</td>
<td>RecoverableMsg</td>
</tr>
<tr>
<td>Warning</td>
<td>Notice</td>
</tr>
<tr>
<td>Indeterminate</td>
<td>Notice</td>
</tr>
<tr>
<td>Cleared</td>
<td>Notice</td>
</tr>
</tbody>
</table>

- **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:

### 58.2.2 Enabling Syslog

The following procedure describes how to enable Syslog.

- **To enable Syslog:**
  1. Open the Syslog Settings page (Troubleshoot menu > Troubleshoot tab > Logging folder > Syslog Settings).
  2. From the ‘Enable Syslog’ drop-down list, select Enable.

  ![Figure 58-2: Enabling Syslog](image)

  3. Click Apply.

### 58.2.3 Configuring the Syslog Server Address

The following procedure describes how to configure the Syslog server’s address to where the device sends Syslog messages.
To configure the Syslog server address:

1. Open the Syslog Settings page (Troubleshoot menu > Troubleshoot tab > Logging folder > Syslog Settings).
2. In the 'Syslog Server IP' field (SyslogServerIP), enter the IP address of the Syslog server.
3. In the 'Syslog Server Port' field, enter the port of the Syslog server.

![Figure 58-3: Configuring the Syslog Server Address](image)

4. Click Apply.

### 58.2.4 Configuring Syslog Debug Level

You can configure the amount of information (debug level) to include in Syslog messages. You can also enable the device to send multiple Syslog messages bundled into a single packet, and enable a protection mechanism that automatically lowers the debug level when the device's CPU resources become low, ensuring sufficient CPU resources are available for processing voice traffic.

To configure the Syslog debug level:

1. Open the Syslog Settings page (Troubleshoot menu > Troubleshoot tab > Logging folder > Syslog Settings).

![Figure 58-4: Configuring Syslog Debug Level](image)

2. From the 'Debug Level' (GwDebugLevel) drop-down list, select the debug level of Syslog messages:
   - **No Debug**: Disables Syslog and no Syslog messages are sent.
   - **Basic**: Sends debug logs of incoming and outgoing SIP messages.
   - **Detailed**: Sends debug logs of incoming and outgoing SIP message as well as many other logged processes.

3. From the 'Syslog Optimization' (SyslogOptimization) drop-down list, select whether you want the device to accumulate and bundle multiple debug messages into a single UDP packet before sending it to a Syslog server. The benefit of the feature is that it reduces the number of UDP Syslog packets, thereby improving (optimizing) CPU utilization. The size of the bundled message is configured by the MaxBundleSyslogLength parameter.

4. From the 'Syslog CPU Protection' (SyslogCpuProtection) drop-down list, select whether you want to enable the protection feature for the device's CPU resources during debug reporting, ensuring voice traffic is unaffected. If CPU resources drop (i.e., high CPU usage) to a critical level (user-defined threshold), the device automatically lowers the debug level to free up CPU resources that were required for the previous debug-level functionality. When CPU resources become available again, the device increases the debug level to its' previous setting. For example, if you set the 'Debug Level' to **Detailed** and CPU resources decrease to the defined threshold, the device automatically changes the level to **Basic**, and if that is not enough, it...
changes the level to **No Debug**. Once CPU resources are returned to normal, the device automatically changes the debug level back to its original setting (i.e., **Detailed**). The threshold is configured by the `DebugLevelHighThreshold` parameter.

5. Click **Apply**.

### 58.2.5 Configuring Reporting of Management User Activities

The device can report operations (activities) performed in the device's management interfaces (e.g., Web and CLI) by management users, in Syslog messages. The Syslog message indicates these logs with the string "Activity Log". Each logged user activity includes the following information:

- **Username** (e.g., "Admin") of the user that performed the action
- **IP address** of the client PC from where the Web user accessed the management interface
- **Protocol** used for the session (e.g., SSH or HTTP)

The following example shows a Web-user activity log (indicating a login action) with the above-mentioned information:

<table>
<thead>
<tr>
<th>Time</th>
<th>IP Address</th>
<th>Operation Details</th>
<th>User</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:07:46.300</td>
<td>10.15.7.95</td>
<td><strong>Activity Log</strong>: WEB: Successful login at 10.15.7.95:80. User: Admin. Session: HTTP (10.13.22.54)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The device can report the following user activities:

- Modifications of individual parameters, for example:

<table>
<thead>
<tr>
<th>Time</th>
<th>IP Address</th>
<th>Operation Details</th>
<th>User</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:33:00.162</td>
<td>10.15.7.95</td>
<td><strong>Activity Log</strong>: Max Login Attempts was changed from '3' to '2'. User: Admin. Session: HTTP (10.13.22.54)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Modifications of table fields, and addition and deletion of table rows, for example:

<table>
<thead>
<tr>
<th>Time</th>
<th>IP Address</th>
<th>Operation Details</th>
<th>User</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>14:42:48.334</td>
<td>10.15.7.95</td>
<td><strong>Activity Log</strong>: Classification - remove line 2. User: Admin. Session: HTTP (10.13.22.54)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Entered CLI commands (modifications of security-sensitive commands are logged without the entered value).
- Configuration file load (reported without per-parameter notifications).
- Auxiliary file load and software update.
- Device reset and burn to flash memory.
- Access to unauthorized Web pages according to the Web user's access level.
- Modifications of "sensitive" parameters.
- Login and logout.
- Actions that are not related to parameter changes (for example, file uploads, file delete, lock-unlock maintenance actions, LDAP clear cache, register-unregister, and start-stop trunk. In the Web, these actions are typically done by clicking a button (e.g., the LOCK button).

For more information on each of the above listed options, see "Syslog, CDR and Debug Parameters" on page 858.

The following procedure describes how to configure management user activity logging through the Web interface. You can also configure it through ini file (`ActivityListToLog`) or CLI (`configure troubleshoot > activity-log`).

#### To configure reporting of Web user activities:

1. Open the Syslog Settings page (**Troubleshoot** tab > **Troubleshoot** menu > **Logging**
folder > **Syslog Settings**).

2. Under the Activity Types to Report group, select the Web actions to report to the Syslog server:

   ![Figure 58-5: Configuring Web Activities to Report to Syslog](image)

3. Click **Apply**.

   **Note:**
   - You can also view logged user activities in the Web interface (see "Viewing Web User Activity Logs" on page 739).
   - Logging of CLI commands can only be configured through CLI or ini file.
   - You can configure the device to send an SNMP trap each time a user performs a Web activity. For more information, see "Configuring SNMP Community Strings" on page 89.

### 58.2.6 Viewing Syslog Messages

You can receive and view Syslog messages generated by the device using any of the following Syslog server types:

- **Wireshark**: Third-party, network protocol analyzer (http://www.wireshark.org).

  **Note:** When debug recording is enabled and Syslog messages are also included in the debug recording, to view Syslog messages using Wireshark, you must install AudioCodes' Wireshark plug-in (acsyslog.dll). Once the plug-in is installed, the Syslog messages are decoded as "AC SYSLOG" and displayed using the "acsyslog" filter (instead of the regular "syslog" filter). For more information on debug recording, see "Debug Recording" on page 801.

- **Third-party, Syslog Server**: Any third-party, Syslog server program that enables filtering of messages according to parameters such as priority, IP sender address, time, and date.

- **Device's CLI Console**: The device sends error messages (e.g., Syslog messages) to the CLI as well as to the configured destination. Use the following commands:
  - To start debug recording:
    ```
    debug log
    ```
58. Syslog and Debug Recording

- To stop debug recording:
  
  no debug log

- To stop all debug recording:
  
  no debug log all

**Device's Web Interface:** The device provides an embedded Syslog server, which is accessed through the Web interface (Troubleshoot tab > Troubleshoot menu > Message Log). This provides limited Syslog server functionality.

**Figure 58-6: Viewing Message Log in Web Interface**

The displayed logged messages are color-coded as follows:

- **Green:** Opening messages at start of message log
- **Black:** Notice messages
- **Blue:** Warning (error) messages

To stop and clear the Message Log, close the Message Log page by accessing any another page in the Web interface.

**Note:**

- 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 displayed on the 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.

**Persistent Logging Storage on the Device:** For more information, see Viewing Persistent-Logging Storage on page 807.

### 58.3 Configuring Debug Recording

This section describes how to configure debug recording and how to collect debug recording packets.
58.3.1 Configuring the Debug Recording Server Address

The procedure below describes how to configure the address of the debug recording server to where the device sends the captured traffic. Once you configure an address, the device generates debug recording packets for all calls. However, you can configure the device to generate debug recording packets for specific calls, using Logging Filter rules in the Logging Filters table (see "Configuring Log Filter Rules" on page 787).

➢ To configure the debug recording server's address:

1. Open the Logging Settings page (Troubleshoot tab > Troubleshoot menu > Logging folder > Logging Settings).

2. In the 'Debug Recording Destination IP' field, configure the IP address of the debug capturing server.

3. In the 'Debug Recording Destination Port' field, configure the port of the debug capturing server.

4. Click Apply.

58.3.2 Collecting Debug Recording Messages

To collect debug recording packets, use the open source packet capturing program, Wireshark. AudioCodes proprietary plug-in files for Wireshark are required.

➢ 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.


3. Copy the plug-in files to the directory in which you installed Wireshark, as follows:
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:
59 Debugging Web Services

If you have configured remote Web services (see Remote Web Services on page 268), you can enable debugging of the remote HTTP clients. You can configure the debug level from 1 to 3, where 3 is the most detailed. The debug messages are sent to the Syslog server.

➢ To configure debugging of Web services:

1. Open the Web Service Settings page (Setup menu > IP Network tab > Web Services folder > Web Service Settings).

2. In the 'Debug Level' field (RestDebugMode), enter the debug level (or disable debugging by configuring it to 0):

   ![Debug Level](image)

3. Click Apply.
This page is intentionally left blank.
# Viewing Persistent-Logging Storage

The device automatically sends logged system event messages to its storage, where they persist even if the device undergoes a reset or powers off. Persistent logging is done by default (and cannot be disabled). Persistent logging eliminates the need for sending logged messages to third-party, logging servers (such as a Syslog server) and provides an efficient tool for troubleshooting the device. When the device's persistent logging storage is full, older messages are overwritten by new messages.

**Note:** When the device operates in HA mode, the Persistent Logging storage is not copied to the Redundant device upon an HA switchover.

The device indexes the stored logged messages and organizes them into the following groups (categories):

- **"conf"** - log messages relating to device boot up and application initialization, including configuration file parsing and management user activities (see Configuring Reporting of Management User Activities)
- **"err"** - log messages relating to warnings, errors, and critical severity levels
- **"ha"** - log messages relating to High Availability (HA)
- **"init"** - log messages relating to device startup
- **"sip"** - log messages relating to SIP messages
- **"other"** - log messages that do not belong to any category above or that are system logs

Each logged message has the following basic structure:

```
<Index>|<Timestamp>|TPApp: [S=<Message Sequence Number>][BID=<Device Identifier>] <Message Description>
```

Where:

- **Index** is the unique number representing the index of the logged message entry in the device’s Persistent Logging storage.
- **Timestamp** is the date and time that the logged event occurred. The timestamp is in the format `YYYY-MM-DD HH-MM-SS`, where `YYYY` is the year (e.g., 2017), `MM` the month (e.g., 01), `DD` the day (e.g., 20), `HH` the hour (e.g., 14), `MM` the minutes (e.g., 59) and `SS` the seconds (e.g., 30).
- **Message Sequence Number** is the sequence number of the logged message (messages are sequentially numbered, e.g., S=4).
- **Device Identifier** is a unique non-SIP session related (e.g., device reset) and device identifier. The BID number is in the format, BID=<last 6 characters in MAC>:<number of times device has reset>. For example, BID=2ed1c8:96 indicates a MAC address of "2ed1c8" and number of device resets of 96.
- **Message Description** is a description of the logged message event.

For example:

```
```

You can view all logged messages or filter the view by category, date range, and number of most recently logged messages. This is done through CLI:
To view persistent logged messages:

```
# debug persistent-log show [category-list {conf|err|ha|init|other|sip}] [start-date <YYYY-MM-DD> end-date <YYYY-MM-DD>] [count <Number of Logs to Display>] [offset <Display Count From Logged Index>]
```

For example, to view two logged messages from logged message at index 120:

```
# debug persistent-log show count 2 offset 120
```

You can view statistics of the persistent logging database, which displays the following:

- "Number of received logs": Number of logs that were sent to the Persistent Logging storage.
- "Number of logs sent to DB": Number of logs that were successfully saved to the Persistent Logging storage.
- "Number of dropped logs": Difference between "Number of received logs" and "Number of logs sent to DB". Dropped logs (typically, due to high load) indicates that the information in the Persistent Logging storage may be inconsequential or missing.

To view statistics of persistent logged messages:

```
# debug persistent-log show stats
```

For example:

```
# debug persistent-log show stats
Number of logs sent to DB: 767
Number of received logs: 767
Number of dropped logs: 0
```
61  Creating Core Dump and Debug Files upon Device Crash

For debugging, you can configure the device to create a core dump file and/or debug file. The files may assist you in identifying the cause of the crash. The core dump can either be included in or excluded from the debug file, or alternatively, sent separately to a TFTP server. You can then provide the files to AudioCodes support team for troubleshooting.

- **Core Dump File:** You can enable the device to send a core dump file to a remote destination upon a device crash. The core dump is a copy of the memory image at the time of the crash. It provides a powerful tool for determining the root cause of the crash. When enabled, the core dump file is sent to a user-defined TFTP server (IP address). If no address is configured, the core dump file is saved to the device’s flash memory (if it has sufficient memory). Each time the device crashes, the new core dump file replaces the previous core dump file, if exists.

  The core dump file is saved as a binary file with the following file name: "core_<device name>_ver_<firmware version>_mac_<MAC address>_ <date>_ <time>". For example, core_acMediant_ver_700-8-4_mac_00908F099096_1-02-2015_3-29-29.

- **Debug File:** You can manually retrieve the debug file from the device and save it to a folder on your local PC. The debug file contains the following information:
  - Exception information, indicating the specific point in the code where the crash occurred and a list of up to 50 of the most recent SNMP alarms that were raised by the device before it crashed.
  - Latest log messages that were recorded prior to the crash.
  - Core dump. The core dump is included only if core dump generation is enabled, no IP address has been configured, and the device has sufficient memory on its flash memory.
  - May include additional application-proprietary debug information.

  The debug file is saved as a zipped file with the following file name: "debug_<device name>_ver_<firmware version>_mac_<MAC address>_ <date>_ <time>". For example, debug_acMediant_ver_700-8-4_mac_00908F099096_1-03-2015_3-29-29.

The following procedure describes how to configure core dump file creation through the Web interface.

- **To enable core dump file generation:**
  1. Set up a TFTP server to where you want to send the core dump file.
  2. Open the Debug Files page (Troubleshoot menu > Troubleshoot tab > Debug folder > Debug Files).

**Figure 61-1: Debug Files Page**

<table>
<thead>
<tr>
<th>CORE DUMP SETTINGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Core Dump</td>
</tr>
<tr>
<td>Core Dump Destination IP</td>
</tr>
</tbody>
</table>

3. From the 'Enable Core Dump' drop-down list, select **Enable**.
4. (Optional) In the 'Core Dump Destination IP' field, enter an IP address of the remote server to where you want the file to be sent.
5. Click **Apply**, and then reset the device with a save-to-flash for your settings to take effect.
You can also delete the core dump file through CLI, as described in the following procedure:

- **To delete the core dump file:**
  - Navigate to the root CLI directory (enable mode), and then enter the following command:
    ```
    # clear debug-file
    ```

The following procedure describes how to retrieve the debug file from the device through the Web interface.

- **To save the debug file:**
  - In the Debug Files page, click the **Save Debug File** button.
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 Rules table lets you test SIP signaling (setup and registration) and media (DTMF signals) of calls between a simulated phone on the device and a remote IP 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 endpoint can be defined as an IP Group or IP address. 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 License Key. For more information, contact your AudioCodes sales representative.

The following procedure describes how to configure test calls through the Web interface. You can also configure it through ini file (Test_Call) or CLI (configure troubleshoot > test-call test-call-table).

➢ To configure a test call:

1. Open the Test Call Rules table (Troubleshooting tab > Troubleshooting menu > Test Call folder > Test Call Rules).
2. Click **New**; the following dialog box appears:

**Figure 62-1: Test Call Rules Table - Add Dialog Box**

3. Configure a test call according to the parameters described in the table below.

4. Click **Apply**, and then save your settings to flash memory.

**Table 62-1: Test Call Rules Table Parameter Descriptions**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Common</strong></td>
<td></td>
</tr>
<tr>
<td>Index</td>
<td>Defines an index number for the new table row. <strong>Note:</strong> Each row must be configured with a unique index.</td>
</tr>
<tr>
<td>Endpoint URI</td>
<td>Defines the endpoint's URI. This can be defined as a user or user@host. The device identifies this endpoint only by the URI's user part. The URI's host part is used in the SIP From header in REGISTER requests. The valid value is a string of up to 150 characters. By default, the parameter is not configured. <strong>Note:</strong> The parameter is mandatory.</td>
</tr>
<tr>
<td>Called URI</td>
<td>Defines the destination (called) URI (user@host). The valid value is a string of up to 150 characters. By default, the parameter is not configured.</td>
</tr>
<tr>
<td>Route By</td>
<td>Defines the type of routing method. This applies to incoming and outgoing calls.</td>
</tr>
<tr>
<td>[1] IP Group = (Default) Calls are matched by (or routed to) an IP Group. To specify the IP Group, see the 'IP Group' parameter in the table.</td>
<td></td>
</tr>
</tbody>
</table>
## Parameter | Description
--- | ---
- **[2]** Dest Address = Calls are matched by (or routed to) a destination IP address. To configure the address, see the 'Destination Address' parameter in the table.

### Note:
- If configured to **Dest Address**, you must assign a SIP Interface (see the 'SIP Interface' parameter in the table).
- For REGISTER messages:
  - ✓ If configured to **IP Group**, only Server-type IP Groups can be used.

### IP Group
- **ip-group-id**
  - Assigns an IP Group. This is the IP Group that the test call is sent to or received from.
  - By default, no value is defined.
  - To configure IP Groups, see "Configuring IP Groups" on page 345.

### Destination Address
- **dst-address**
  - Defines the destination host.
  - The valid value is an IP address[:port] or DNS name[:port].

### SIP Interface
- **sip-interface-name**
  - Assigns a SIP Interface. This is the SIP Interface to which the test call is sent and received from.
  - By default, no value is defined.
  - To configure SIP Interfaces, see Configuring SIP Interfaces on page 337.

### Application Type
- **application-type**
  - Defines the application type for the endpoint. This associates the IP Group and SRD to a specific SIP interface.
  - [2] SBC = SBC application

### Destination Transport Type
- **dst-transport**
  - Defines the transport type for outgoing calls.
  - [-1] = Not configured (default)
  - [0] UDP
  - [1] TCP
  - [2] TLS

### QoE Profile
- **qoe-profile**
  - Assigns a QoE Profile to the test call.
  - By default, no value is defined.
  - To configure QoE Profiles, see "Configuring Quality of Experience Profiles" on page 307.

### Bandwidth Profile
- **bandwidth-profile**
  - Assigns a Bandwidth Profile to the test call.
  - By default, no value is defined.
  - To configure Bandwidth Profiles, see "Configuring Bandwidth Profiles"
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Test_Call_BWProfile]</td>
<td>Profiles&quot; on page 312.</td>
</tr>
</tbody>
</table>

### Authentication

**Note:** These parameters are applicable only if the 'Call Party' parameter (see below) is configured to **Caller**.

#### Auto Register

**Note:** Enables automatic registration of the endpoint. The endpoint can register to the device itself or to the 'Destination Address' or 'IP Group' parameter settings (see above).

- [0] Disable (default)
- [1] Enable

#### Username

Define the authentication username. By default, no username is defined.

#### Password

Define the authentication password. By default, no password is defined.

### Test Setting

Define whether the test endpoint is the initiator (caller) or receiving side (called) of the test call.

- [0] Caller (default)
- [1] Called

#### Maximum Channels for Session

Define the maximum number of concurrent channels for the test session. For example, if you have configured an endpoint "101" and you configure the parameter to "3", the device automatically creates three simulated endpoints - "101", "102" and "103" (i.e., consecutive endpoint URIs are assigned).

The default is 1.

#### Call Duration

Define the call duration (in seconds).

**Note:** The valid value is -1 to 100000. The default is 20. A value of 0 means infinite. A value of -1 means that the parameter value is automatically calculated according to the values of the 'Calls per Second' and 'Maximum Channels for Session' parameters.

**Note:** The parameter is applicable only if you configure 'Call Party' to **Caller**.

#### Calls per Second

Define the number of calls per second.

**Note:** The parameter is applicable only if you configure 'Call Party' to **Caller**.

#### Test Mode

Define 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
62. Testing SIP Signaling Calls

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Test Duration</td>
<td>Defines the test duration (in minutes). The valid value is 0 to 100000. The default is 0 (i.e., unlimited).</td>
</tr>
<tr>
<td></td>
<td>Note: The parameter is applicable only if you configure 'Call Party' to Caller.</td>
</tr>
<tr>
<td>Schedule Interval</td>
<td>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).</td>
</tr>
<tr>
<td></td>
<td>Note: The parameter is applicable only if you configure 'Call Party' to Caller.</td>
</tr>
</tbody>
</table>

### 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 Rules table, select the required test call entry.
  2. From the **Action** drop-down list, choose the required command:
     - **Dial**: Starts the test call (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.
62.3 Viewing Test Call Status

You can view the status of test call rules in the 'Test Status' field of the Test Call Rules table. The status can be one of the following:

<table>
<thead>
<tr>
<th>Status</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;Idle&quot;</td>
<td>Test call is not active.</td>
</tr>
<tr>
<td>&quot;Scheduled&quot;</td>
<td>Test call is planned to run (according to the 'Schedule Interval' parameter).</td>
</tr>
<tr>
<td>&quot;Running&quot;</td>
<td>Test call has been started (i.e., by clicking Dial from the 'Action' drop-down list).</td>
</tr>
<tr>
<td>&quot;Receiving&quot;:</td>
<td>Test call has been automatically activated by calls received from the remote endpoint for the test call endpoint (when all these calls end, the status returns to &quot;Idle&quot;).</td>
</tr>
<tr>
<td>&quot;Terminating&quot;</td>
<td>Test call is in the process of terminating currently established calls (when Drop Call is clicked from the 'Action' drop-down list to stop the test).</td>
</tr>
<tr>
<td>&quot;Done&quot;</td>
<td>Test call has successfully completed (or was prematurely stopped by clicking the Drop Call from the 'Action' drop-down list).</td>
</tr>
</tbody>
</table>

62.4 Viewing Test Call Statistics

You can view statistical information on the test call.

**Note:**
- On the receiving side, when the first call is accepted in "Idle" state, statistics are reset.
- The device also generates CDRs for test calls if you have enabled CDR generation (see "Configuring CDR Reporting" on page 774). To view CDRs of test calls, see Viewing CDR Test Calls on page 752.

➢ To view statistics of a test call:

1. In the Test Call Rules table, select the required test call row.
2. Scroll down the page to the area below the table. Statistics of the selected test call are displayed under the Statistics group, as shown in the example below:

**Figure 62-2: Viewing Test Call Statistics**

<table>
<thead>
<tr>
<th>Statistics Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Calls</td>
<td>Number of currently established test calls.</td>
</tr>
<tr>
<td>Call Attempts</td>
<td>Number of calls that were attempted.</td>
</tr>
<tr>
<td>Total Established Calls</td>
<td>Total number of calls that were successfully established.</td>
</tr>
<tr>
<td>Total Failed Attempts</td>
<td>Total number of call attempts that failed.</td>
</tr>
<tr>
<td>Remote Disconnections Count</td>
<td>Number of calls that were disconnected by the remote side.</td>
</tr>
<tr>
<td>Average CPS</td>
<td>Average calls per second.</td>
</tr>
<tr>
<td>Elapsed Time</td>
<td>Duration of the test call since it was started (or restarted).</td>
</tr>
<tr>
<td>Test Status</td>
<td>Status (brief description) as displayed in the 'Test Status' field (see &quot;Viewing Test Call Status&quot; on page 816).</td>
</tr>
<tr>
<td>Detailed Status</td>
<td>Displays a detailed description of the test call status:</td>
</tr>
<tr>
<td></td>
<td>&quot;Idle&quot;: Test call is currently not active.</td>
</tr>
<tr>
<td></td>
<td>&quot;Scheduled - Established Calls: &lt;number of established calls&gt;, ASR: &lt;ASR&gt;%&quot;: Test call is planned to run (according to 'Schedule Interval' parameter settings) and also shows the following summary of completed test calls:</td>
</tr>
<tr>
<td></td>
<td>☑ Total number of test calls that were established.</td>
</tr>
<tr>
<td></td>
<td>☑ Number of successfully answered calls out of the total number of calls attempted (ASR).</td>
</tr>
<tr>
<td></td>
<td>&quot;Running (Calls: &lt;number of active calls&gt;, ASR: &lt;ASR&gt;%)&quot;: Test call has been started (i.e., the Dial command was clicked) and shows the following:</td>
</tr>
</tbody>
</table>
### Statistics Field

<table>
<thead>
<tr>
<th>Statistics Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Number of currently active test calls.</td>
<td>✓ Number of successfully answered calls out of the total number of calls attempted (Answer Success Ratio or ASR).</td>
</tr>
<tr>
<td>✓ &quot;Receiving (&lt;number of active calls&gt;): 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 &quot;Idle&quot;.</td>
<td></td>
</tr>
<tr>
<td>✓ &quot;Terminating (&lt;number of active calls&gt;): 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.</td>
<td></td>
</tr>
<tr>
<td>✓ &quot;Done - Established Calls: &lt;number of established calls&gt;, ASR: &lt;ASR&gt;%&quot;: Test call has been successfully completed (or was prematurely stopped by clicking the Drop Call command) and shows the following:</td>
<td>✓ Total number of test calls that were established.</td>
</tr>
<tr>
<td>✓ Number of successfully answered calls out of the total number of calls attempted (ASR).</td>
<td></td>
</tr>
</tbody>
</table>

#### 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.

## 62.5 Configuring DTMF Tones for Test Calls

By default, the device plays the DTMF signal tone "3212333" to remote tested endpoints for answered calls (incoming and outgoing). For basic test calls (as described in Configuring Basic Test Calls on page 819), the device can play only the configured DTMF tones (or none, if not configured). For test call endpoints that are configured in the Test Call Rules table, you can configure the device to play either DTMF tones or a tone from an installed PRT file (Test Call Tone). For more information, see Configuring Test Call Endpoints on page 811.

**Note:**
- You can configure the DTMF signaling type (e.g., out-of-band or in-band) using the 'DTMF Transport Type' parameter. For more information, see Configuring DTMF Transport Types.
- To generate DTMF tones, the device's DSP resources are required.

- To configure played DTMF signal to answered test call:
  1. Open the Test Call Settings page (Troubleshooting tab > Troubleshooting menu > Test Call folder > Test Call Settings).
2. In the 'Test Call DTMF String' field, enter the DTMF string (up to 15 digits):

![Configuring Played DTMF Tone for Test Calls](image)

3. Click **Apply**.

### 62.6 Configuring Basic Test Calls

The Basic Test Call feature tests incoming calls from remote SIP (IP) endpoints to a simulated test endpoint on the device. The only required configuration is to assign a prefix number (test call ID) to the simulated endpoint. Incoming calls with this called (destination) prefix number are identified by the device as test calls and sent to the simulated endpoint. The figure below displays a basic test call example:

![Incoming Test Call Example](image)

- **To configure basic call testing:**
  1. Open the Test Call Settings page (**Troubleshooting** tab > **Troubleshooting** menu > **Test Call** folder > **Test Call Settings**).
  2. In the 'Test Call ID' field, enter a prefix for the simulated endpoint:

![Configuring Basic Test Calls](image)

3. Click **Apply**.

**Note:**
- The device can play DTMF tones to the remote endpoint. For more information, see Configuring DTMF Tones for Test Calls on page 818.
- Test calls are done on all SIP Interfaces.

### 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.

![Single Test Call Example]

- Test Call Rules table configuration:
  - Endpoint URI: "101"
  - Called URI: "201"
  - Route By: **Dest Address**
  - Destination Address: "10.30.40.01"
  - SIP Interface: SIPInterface_0
  - Call Party: **Caller**
  - Test Mode: **Once**

**Batch Test Call Scenario:** This example describes the configuration of a batch test call setup for scheduled and continuous call testing of multiple endpoints. The test call is done between two AudioCodes devices - Device A and Device B - with simulated test endpoints. This eliminates the need for phone users, who would otherwise need to answer and end calls many times for batch testing. The calls are initiated from Device A, where Device B serves as the remote answering endpoint.

![Batch Test Call Example]

- Test Call Rules table configuration at Device A:
  - Endpoint URI: "101"
  - Called URI: "201"
  - Route By: **Dest Address**
  - Destination Address: "10.13.4.12"
  - SIP Interface: SIPInterface_0
  - Call Party: **Caller**
  - Maximum Channels for Session: "3" (configures three endpoints - "101", "102" and "103")
62. Testing SIP Signaling Calls

- Call Duration: "5" (seconds)
- Calls per Sec: "1"
- Test Mode: **Continuous**
- Test Duration: "3" (minutes)
- Schedule Interval: "180" (minutes)

- Test Call Rules table configuration at Device B:
  - Endpoint URI: "201"
  - Maximum Channels for Session: "3" (configures three endpoints - "201", "202" and "203")

**Registration Test Call Scenario:** This example describes the configuration for testing the registration and authentication (i.e., username and password) process of a simulated test endpoint on the device with an external proxy/registrar server. This is useful, for example, for verifying that endpoints located in the LAN can register with an external proxy and subsequently, communicate with one another.

**Figure 62-8: Test Call Registration Example**

This example assumes that you have configured your device for communication between LAN phone users such as IP Groups to represent the device (10.13.4.12) and the proxy server, and IP-to-IP routing rules to route calls between these IP Groups.

- Test Call Rules table configuration:
  - Endpoint URI: "101"
  - Called URI: "itsp"
  - Route By: **Dest Address**
  - Destination Address: "10.13.4.12" (this is the IP address of the device itself)
  - SIP Interface: SIPInterface_0
  - Auto Register: **Enable**
  - User Name: "testuser"
  - Password: "12345"
  - Call Party: **Caller**
This page is intentionally left blank.
63 Pinging a Remote Host or IP Address

You can verify the network connectivity with a remote host or IP address by pinging the network entity.

- IPv4: The ping to an IPv4 address can be done from any of the device's VoIP interfaces that is configured with an IPv4 address. The ping is done using the following CLI command:

  # ping <IPv4 ip address or host name> source [voip] interface

For a complete description of the ping command, refer to the CLI Reference Guide.
This page is intentionally left blank.
## 64 Dialing Plan Notation for Routing and Manipulation

The device supports flexible dialing plan notations for denoting the prefix and/or suffix source and/or destination numbers and SIP URI user names in the routing and manipulation tables.

**Note:** When configuring phone numbers or prefixes in the Web interface, enter them only as digits without any other characters. For example, if you wish to enter the phone number 555-1212, it must be entered as 5551212 without the hyphen (-). If the hyphen is entered, the entry is invalid.

### Table 64-1: Dialing Plan Notations for Prefixes and Suffixes

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x (letter &quot;x&quot;)</td>
<td>Wildcard that denotes any single digit or character.</td>
</tr>
</tbody>
</table>
| # (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

**Note:**
- 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: |
<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
</table>
| ✓ [5551200-5551300]# | To depict prefix numbers from 123100 to 123200:  
| ✓ 123[100-200]# | To depict prefix and suffix numbers together:  
| ✓ 03(100): for any number that starts with 03 and ends with 100.  
| ✓ [100-199](100,101,105): for a number that starts with 100 to 199 and ends with 100, 101 or 105.  
| ✓ 03(abc): for any number that starts with 03 and ends with abc.  
| ✓ 03(5xx): for any number that starts with 03 and ends with 5xx.  
| ✓ 03(400,401,405): for any number that starts with 03 and ends with 400 or 401 or 405. |

**Note:**  
- 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 received number +49 (7303) 165-xxxx to +49 \287303\29 165-xxxx, 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., xx165xxxx#); the prefix to add to the number would include the HEX values (e.g., +49 \287303\29 165-).  

Below is a list of common ASCII characters and their corresponding HEX values:  

<table>
<thead>
<tr>
<th>ASCII Character</th>
<th>HEX Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>\2a</td>
</tr>
<tr>
<td>(</td>
<td>\28</td>
</tr>
<tr>
<td>Notation</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>)</td>
<td>\29</td>
</tr>
<tr>
<td>\</td>
<td>\5c</td>
</tr>
<tr>
<td>/</td>
<td>\2f</td>
</tr>
</tbody>
</table>
This page is intentionally left blank.
65 Configuration Parameters Reference

The device's configuration parameters, default values, and their descriptions are documented in this section.

Note: Parameters and values enclosed in square brackets [...] represent the ini file parameters and their enumeration values.

65.1 Management Parameters

This section describes the device's management-related parameters.

65.1.1 General Parameters

The general management parameters are described in the table below.

Table 65-1: General Management Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [WebLoginBlockAutoComplete]        | Disables autocompletion when entering the management login username in the "Username" field of the device's Web interface. Disabling autocompletion may be useful for security purposes by hiding previously entered usernames and thereby, preventing unauthorized access to the device's management interface.  
  • [0] Disable = (Default) Autocompletion is enabled and the 'Username' field automatically offers previously logged in usernames.  
  • [1] Enable = Autocompletion is disabled. |
| [EnforcePasswordComplexity]        | Enables the enforcement of management login-password complexity requirements to ensure strong passwords.  
  • [0] Disable (default)  
  • [1] Enable  
  For more information on password complexity requirements, see the 'Password' parameter in Configuring Management User Accounts on page 73. |

Access List Table

configure network > access-list [WebAccessList_x]

This table configures up to ten IP addresses that are permitted to access the device's Web interface and Telnet interfaces. Access from an undefined IP address is denied. When no IP addresses are defined in this table, this security feature is inactive (i.e., the device can be accessed from any IP address).

The default is 0.0.0.0 (i.e., the device can be accessed from any IP address).

For example:

`WebAccessList_0 = 10.13.2.66`
`WebAccessList_1 = 10.13.77.7`

For a description of the parameter, see "Configuring Web and Telnet Access List" on page 81.

configure troubleshoot

Defines the console access mode.
Parameter | Description
--- | ---
> startup-n-recovery > system-console-mode [SystemConsoleMode] | • [0] = (Default) VGA  
• [1] = RS-232  
**Note:** For the parameter to take effect, a device reset is required.

[CustomerSN] | Defines a serial number (S/N) for the device.  
**Note:** The device's original S/N is automatically added at the end of the configured S/N. For example, if the original S/N is 8906721 and the configured S/N is "abc123", the resultant S/N is "abc1238906721".

Product Key configure system > product-key [ProductKey] | Defines the device's Product Key.  
The valid value is a string of up to 40 characters.

**Local Users Table**

Local Users configure system > user [WebUsers] | The table defines management users.  
The format of the ini file table parameter is as follows:  
[WebUsers]  
FORMAT WebUsers_Index = WebUsers_Username, WebUsers_Password, WebUsers_Status, WebUsers_PwAgeInterval, WebUsers_SessionLimit, WebUsers_SessionTimeout, WebUsers_BlockTime, WebUsers_UserLevel, WebUsers_SSHPublicKey;

For more information, see Configuring Management User Accounts on page 73.

**Additional Management Interfaces Table**

Additional Management Interfaces configure system > additional-mgmt-if [AdditionalManagementInterfaces] | The table defines additional management interfaces.  
The format of the ini file table parameter is as follows:  
[AdditionalManagementInterfaces]  
FORMAT AdditionalManagementInterfaces_Index = AdditionalManagementInterfaces_InterfaceName, AdditionalManagementInterfaces_TLSContextName, AdditionalManagementInterfaces_HTTPSOnly;

For more information, see Configuring Additional Management Interfaces on page 71.

### 65.1.2 Web Parameters

The Web parameters are described in the table below.

**Table 65-2: Web Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Enable web access from all interfaces web-access-from-all-interfaces [EnableWebAccessFromAllInterfaces] | Enables Web access from any of the device's IP network interfaces. The feature applies to HTTP and HTTPS protocols.  
• [0] = (Default) Disable – Web access is only through the |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OAMP interface.</td>
<td><strong>[1] = Enable</strong> - Web access is through any network interface.</td>
</tr>
</tbody>
</table>
| **Note:**                        | • For the parameter to take effect, a device reset is required.  
• Instead of using this parameter, you can use the Additional Management Interfaces table to assign specific IP network interfaces for management interfaces (as well as assign them TLS Contexts). For more information, see Configuring Additional Management Interfaces on page 71. |
| Password Change Interval         | Defines the duration (in minutes) of the validity of Web login passwords. When this duration expires, the password of the Web user must be changed.  
The valid value is 0 to 100000, where 0 means that the password is always valid. The default is 1140.  
**Note:** The parameter is applicable only when using the Local Users table, where the default value of the 'Password Age' parameter in the Local Users table inherits the parameter's value. |
| User Inactivity Timer            | 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 Security Administrator or Master user.  
The valid value is 0 to 10000, where 0 means inactive. The default is 90.  
**Note:** The parameter is applicable only when using the Local Users table. |
| Session Timeout                  | Defines the duration (in minutes) of inactivity of a logged-in user in the Web interface, after which the user is automatically logged off the Web session. In other words, the session expires when the user has not performed any operations (activities) in the Web interface for the configured duration.  
The valid value is 0-100000, where 0 means no timeout. The default is 15.  
**Note:** You can also configure the functionality per user in the Local Users table (see "Configuring Management User Accounts" on page 73), which overrides this global setting. |
| Deny Access On Fail Count        | 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. |
<p>| Deny Authentication Timer        | 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 |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
| Display Last Login Information [DisplayLoginInformation] | Enables display of user's login information on each successful login attempt.  
- [0] Disable (default)  
- [1] Enable |
| [EnableMgmtTwoFactorAuthentication] | Enables Web login authentication using a third-party, smart card.  
- [0] = Disable (default)  
- [1] = Enable  
When enabled, the device retrieves the Web user’s login username from the smart card, which is automatically displayed (read-only) in the Web Login screen; the user is then required to provide only the login password. Typically, a TLS connection is established between the smart card and the device’s Web interface, and a RADIUS server is implemented to authenticate the password with the username. Thus, this feature implements a two-factor authentication - what the user has (the physical card) and what the user knows (i.e., the login password). |
| http-port [HTTPport] | Defines the LAN HTTP port for Web management. To enable Web management from the LAN, configure the desired port. The default is 80.  
**Note:** For the parameter to take effect, a device reset is required. |
| [DisableWebConfig] | Determines whether the entire Web interface is read-only.  
- [0] = (Default) Enables modifications of parameters.  
- [1] = Web interface is read-only.  
When in read-only mode, parameters can't be modified and the following pages can't be accessed: Web User Accounts, TLS Contexts, Time and Date, Maintenance Actions, Load Auxiliary Files, Software Upgrade Wizard, and Configuration File.  
**Note:** For the parameter to take effect, a device reset is required. |
| [ResetWebPassword] | Enables the device to restore the default management users:  
- Security Administrator user (username "Admin"; password "Admin")  
- Monitor user (username "User"; password "User")  
In addition, all other users that may have been configured (in the Local Users table) are deleted.  
- [0] = (Default) Disabled. Currently configured users (usernames and passwords) are retained. |

**Note:** For the parameter to take effect, a device reset is required.
### Parameter

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>[1]</strong> = Default users are restored (see description above) and all other configured users are deleted.</td>
</tr>
</tbody>
</table>

**Note:**
- For the parameter to take effect, a device reset is required.
- In addition to the ini file (see above), you can also restore the default user accounts through the following management platforms:
  - SNMP (restores default users and retains other configured users):  
    1. Set `acSysGenericNILine` 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`

### Customizing Web GUI

**[WelcomeMessage]**

configure system > welcome-msg

Defines a welcome message displayed on the Web interface’s Web Login page.

The format of the ini file table parameter is:

```
[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 = "**********************************" ;
```

For more information, see Creating a Login Welcome Message on page 71.

**Note:**
- Each index row represents a line of text. Up to 20 lines (or rows) of text can be defined.
- The configured text message must be enclosed in double quotation marks (i.e., "...").
- If the parameter is not configured, no Welcome message is displayed.

**[UseProductName]**

Enables the option to customize the name of the device (product) that appears in the management interfaces.

- **[0]** = Disabled (default).
- **[1]** = Enables the display of a user-defined name, which
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>is configured by the UserProductName parameter. For more information, see Customizing the Product Name on page 69.</strong></td>
<td></td>
</tr>
<tr>
<td><strong>[UserProductName]</strong></td>
<td>Defines a name for the device instead of the default name. The value can be a string of up to 29 characters. For more information, see Customizing the Product Name on page 69. <strong>Note</strong>: To enable customization of the device name, see the UseProductName parameter.</td>
</tr>
</tbody>
</table>
| **[UseWebLogo]**          | Defines whether the Web interface displays a logo image or text.  
  - [0] = (Default) The Web interface displays a logo image, configured by the LogoFileName parameter.  
  For more information, see Replacing the Corporate Logo on page 66.                                                                                                                                 |
| **[WebLogoText]**         | Defines the text that is displayed instead of the logo in the Web interface. The valid value is a string of up to 15 characters. For more information, see Replacing the Corporate Logo with Text on page 68. **Note**: The parameter is applicable only when the UseWebLogo parameter is configured to 1. |
| **[LogoWidth]**           | Defines the width (in pixels) of the logo image that you want displayed in the Web interface instead of the default logo. The valid value is 0 to 199. The default is 145. For more information, see Replacing the Corporate Logo with an Image on page 67. **Notes**:  
  - The optimal setting depends on your screen resolution.  
  - If the width of the loaded image is greater than the maximum value, the device automatically resizes the image to the default width size.  
  - The height is limited to 24 pixels.  
  - The parameter is applicable only when the UseWebLogo parameter is configured to 0.  
  - To define the image file, see the LogoFileName parameter.                                                                                                                                 |
| **[LogoFileName]**        | Defines the name of the image file that you want loaded to the device. This image is displayed as the logo in the Web interface (instead of AudioCodes logo). The file name can be up to 47 characters. For more information, see Replacing the Corporate Logo with an Image on page 67. **Notes**:  
  - The image file type can be one of the following: GIF, PNG, JPG, or JPEG.тика.                                                                                                                                 |
The size of the image file can be up to 64 Kbytes. The parameter is applicable only when the UseWebLogo parameter is configured to 0.

### 65.1.3 Telnet Parameters

The Telnet parameters are described in the table below.

**Table 65-3: Telnet Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded Telnet Server enable</td>
<td>Enables the device's embedded Telnet server. Telnet is disabled by default for security.</td>
</tr>
<tr>
<td>Telnet Server TCP Port</td>
<td>Defines the port number for the embedded Telnet server. The valid range is all valid port numbers. The default port is 23.</td>
</tr>
<tr>
<td>Telnet Server Idle Timeout</td>
<td>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.</td>
</tr>
<tr>
<td>Maximum Telnet Sessions</td>
<td>Defines the maximum number of permitted, concurrent Telnet/SSH sessions. The valid range is 1 to 5 sessions. The default is 2.</td>
</tr>
<tr>
<td>[CLIPrivPass]</td>
<td>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 &quot;Admin&quot;.</td>
</tr>
<tr>
<td>Default Terminal Window Height</td>
<td>Defines the number (height) of output lines displayed in the CLI terminal window. This applies to all new CLI sessions and is preserved after device resets. The valid value range is -1 (default) and 0-65535:</td>
</tr>
</tbody>
</table>
  - A value of -1 means that the parameter is disabled and the settings of the CLI command window-height is used. |
  - A value of 0 means that all the CLI output is displayed in the window. |
### Parameter Description

- A value of 1 or greater displays that many output lines in the window and if there is more output, the “—MORE—” prompt is displayed. For example, if you configure the parameter to 4, up to four output lines are displayed in the window and if there is more output, the “—MORE—” prompt is displayed (at which you can press the spacebar to display the next four output lines).

**Note:** You can override this parameter for a specific CLI session and configure a different number of output lines, by using the `window-height` CLI command in the currently active CLI session.

---

#### 65.1.4 ini File Parameters

The parameters relating to ini-file management are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| `[INIPasswordsDisplayType]` | Defines how passwords are displayed in the ini file.  
  - [0] = (default) Disable. Passwords are obscured (“encoded”). The passwords are displayed in the following syntax: $1$<obscured password> (e.g., $1$S3p+fno=).  
  - [1] = Enable. All passwords are hidden and replaced by an asterisk (*). |

---

#### 65.1.5 SNMP Parameters

The SNMP parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Disable SNMP configure system > snmp settings > disable [DisableSNMP] | Enables and disables SNMP.  
  - [0] No = (Default) SNMP is enabled.  
  - [1] Yes = SNMP is disabled. |
| configure system > snmp settings > port [SNMPPort] | Defines the device’s local (LAN) UDP port used for SNMP Get/Set commands.  
  The range is 100 to 3999. The default port is 161.  
  **Note:** For the parameter to take effect, a device reset is required. |
| [ChassisPhysicalAlias] | Defines the ‘alias’ name object for the physical entity as specified by a network manager, and provides a non-volatile ‘handle’ for the physical entity.  
  The valid range is a string of up to 255 characters. |
| [ChassisPhysicalAssetID] | Defines the user-assigned asset tracking identifier object for the device’s chassis as specified by an EMS, and provides non-volatile storage of this information.  
  The valid range is a string of up to 255 characters. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ifAlias]</td>
<td>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.</td>
</tr>
</tbody>
</table>
| configure system > snmp trap > auto-send-keep-alive [SendKeepAliveTrap] | Enables the device to send NAT keep-alive traps to the port of the SNMP network management station (e.g., AudioCodes EMS). This is used for NAT traversal, and allows SNMP communication with AudioCodes EMS management platform, located in the WAN, when the device is located behind NAT. It is needed to keep the NAT pinhole open for the SNMP messages sent from EMS to the device. The device sends the trap periodically - every 9/10 of the time configured by the NATBindingDefaultTimeout parameter. The trap that is sent is acKeepAlive. For more information on the SNMP trap, refer to the SNMP Reference Guide.  
  - [0] = (Default) Disable  
  - [1] = Enable  
  To configure the port number, use the KeepAliveTrapPort parameter.  
  **Note:** For the parameter to take effect, a device reset is required. |
| [KeepAliveTrapPort] | Defines the port of the SNMP network management station to which the device sends keep-alive traps.  
  The valid range is 0 - 65534. The default is port 1161.  
  To enable NAT keep-alive traps, use the SendKeepAliveTrap parameter. |
| [PM_EnableThresholdAlarms] | Enables the sending of the SNMP trap event, acPerformanceMonitoringThresholdCrossing which is sent every time the threshold (high and low) of a Performance Monitored object (e.g., acPMMediaRealmAttributesMediaRealmBytesTxHighThreshold) is crossed.  
  - [0] = (Default) Disable  
  - [1] = Enable |
| configure system > snmp settings > sys-oid [SNMPSysOid] | Defines the SNMP MIB OID for the base product system.  
  The default is 1.3.6.1.4.1.5003.8.1.1.  
  **Note:**  
  - For the parameter to take effect, a device reset is required.  
  - The device automatically adds the device’s unique product identifier number at the end of your OID. |
| [SNMPTrapEnterpriseOid] | Defines the SNMP MIB OID for the Trap Enterprise.  
  The default is 1.3.6.1.4.1.5003.9.10.1.21.  
  **Note:**  
  - For the parameter to take effect, a device reset is required.  
  - The device automatically adds the device’s unique product identifier number at the end of your OID. |
<p>| [acUserInputAlarmDescription] | Defines the description of the input alarm. |
| [acUserInputAlarmSeverity] | Defines the severity of the input alarm. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[AlarmHistoryTableMaxSize]</td>
<td>Defines the maximum number of rows in the Alarm History table. The parameter can be controlled by the Config Global Entry Limit MIB (located in the Notification Log MIB). The valid range is 50 to 1000. The default is 500. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
</tbody>
</table>
| [ActiveAlarmTableMaxSize]         | Defines the maximum number of currently active alarms that can be displayed in the Active Alarms table. When the table reaches this user-defined maximum capacity (i.e., full), the device sends the SNMP trap event, acActiveAlarmTableOverflow. If the table is full and a new alarm is raised by the device, the new alarm is not displayed in the table. The valid range is 100 to 1000. The default is 250. For more information on the Active Alarms table, see Viewing Active Alarms on page 735. **Note:**  
  - For the parameter to take effect, a device reset is required.  
  - To clear the acActiveAlarmTableOverflow trap, you must reset the device. The reset also deletes all the alarms in the Active Alarms table. |
| configure system > snmp settings > 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  
  **Note:**  
  - For the parameter to take effect, a device reset is required.  
  - Before setting the parameter, all SNMPv3 users must be deleted; otherwise, the parameter setting is ignored.  
  - If the supplied key does not pass validation of the 12 Hex values input or it is set with the default value, the engine ID is generated according to RFC 3411. |

**SNMP Trap Destination Parameters** (configure system > snmp trap destination)  
**Note:** Up to five SNMP trap managers can be defined.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 |
<p>| IP Address ip-address [SNMPManagerTableIP_x] | Defines the IP address of the remote host used as an SNMP Manager. The device sends SNMP traps to this IP address. Enter the IP address in dotted-decimal notation, e.g., 108.10.1.255. |
| Trap Port                         | Defines the port number of the remote SNMP Manager. The |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>port</td>
<td>device sends SNMP traps to this port. The valid SNMP trap port range is 100 to 4000. The default port is 162.</td>
</tr>
<tr>
<td>[SNMPManagerTrapPort_x]</td>
<td>Enables the sending of traps to the corresponding SNMP manager.</td>
</tr>
<tr>
<td>Trap Enable send-trap</td>
<td>Enables the sending of traps to the corresponding SNMP manager.</td>
</tr>
<tr>
<td>[SNMPManagerTrapSendingEnable_x]</td>
<td>Enables the sending of traps to the corresponding SNMP manager.</td>
</tr>
<tr>
<td>Trap User trap-user</td>
<td>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.</td>
</tr>
<tr>
<td>[SNMPManagerTrapUser_x]</td>
<td>Defines an FQDN of the remote host used as an SNMP manager to receive traps sent by the device. The device sends the traps to the DNS-resolved IP address. The valid range is a string of up to 99 characters. For more information, see &quot;Configuring an SNMP Trap Destination with FQDN&quot; on page 93.</td>
</tr>
<tr>
<td>Trap Manager Host Name manager-host-name</td>
<td>Defines an FQDN of the remote host used as an SNMP manager to receive traps sent by the device. The device sends the traps to the DNS-resolved IP address. The valid range is a string of up to 99 characters. For more information, see &quot;Configuring an SNMP Trap Destination with FQDN&quot; on page 93.</td>
</tr>
<tr>
<td>[SNMPTrapManagerHostName]</td>
<td>Defines an FQDN of the remote host used as an SNMP manager to receive traps sent by the device. The device sends the traps to the DNS-resolved IP address. The valid range is a string of up to 99 characters. For more information, see &quot;Configuring an SNMP Trap Destination with FQDN&quot; on page 93.</td>
</tr>
<tr>
<td>Activity Trap configure troubleshoot &gt; activity-trap [EnableActivityTrap]</td>
<td>Enables the device to send an SNMP trap to notify of Web user activities in the Web interface. The activities to report are configured by the ActivityListToLog parameter.</td>
</tr>
<tr>
<td>[SNMPV3 Users Table]</td>
<td>Enables the device to send an SNMP trap to notify of Web user activities in the Web interface. The activities to report are configured by the ActivityListToLog parameter.</td>
</tr>
<tr>
<td>SNMP Community String Parameters</td>
<td>Enables the device to send an SNMP trap to notify of Web user activities in the Web interface. The activities to report are configured by the ActivityListToLog parameter.</td>
</tr>
<tr>
<td>[SNMPTrustedMgr_x]</td>
<td>The table defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. For more information, see &quot;Configuring SNMP Trusted Managers&quot; on page 93.</td>
</tr>
<tr>
<td>SNMP V3 Users Table</td>
<td>The table defines up to five IP addresses of remote trusted SNMP managers from which the SNMP agent accepts and processes SNMP Get and Set requests. For more information, see &quot;Configuring SNMP Trusted Managers&quot; on page 93.</td>
</tr>
</tbody>
</table>
### Parameter Configuration

#### SNMP V3 Users

The table defines SNMP v3 users. The format of the ini file table parameter is:

```
[SNMPUsers]
FORMAT SNMPUsers_Index = SNMPUsers_Username, SNMPUsers_AuthProtocol, SNMPUsers_PrivProtocol, SNMPUsers_AuthKey, SNMPUsers_PrivKey, SNMPUsers_Group;
```

For example:

```
SNMPUsers 1 = v3admin1, 1, 0, myauthkey, -, 1;
```

The example above configures user 'v3admin1' with security level authNoPriv(2), authentication protocol MD5, authentication text password 'myauthkey', and ReadWriteGroup2.

For more information, see "Configuring SNMP V3 Users" on page 94.

### 65.1.6 Serial Parameters

The serial interface parameters are described in the table below.

**Table 65-6: Serial Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>- [0] = Enabled</td>
</tr>
<tr>
<td></td>
<td>- [1] = (Default) Disabled</td>
</tr>
<tr>
<td></td>
<td>The RS-232 serial port can be used to change the networking parameters and view error/notification messages. To establish serial communication with the device, see &quot;Establishing a CLI Session&quot; on page 86.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>[SerialBaudRate]</td>
<td>Defines the serial communication baud rate.</td>
</tr>
<tr>
<td></td>
<td>The valid values include the following: 1200, 2400, 9600, 14400, 19200, 38400, 57600, or 115200 (default).</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>[SerialData]</td>
<td>Defines the serial communication data bit.</td>
</tr>
<tr>
<td></td>
<td>- [7] = 7-bit</td>
</tr>
<tr>
<td></td>
<td>- [8] = (Default) 8-bit</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>[SerialParity]</td>
<td>Defines the serial communication polarity.</td>
</tr>
<tr>
<td></td>
<td>- [0] = (Default) None</td>
</tr>
<tr>
<td></td>
<td>- [1] = Odd</td>
</tr>
<tr>
<td></td>
<td>- [2] = Even</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
</tbody>
</table>
65.1.7 Auxiliary and Configuration File Name Parameters

The table below lists the ini file parameters associated with the Auxiliary files. For more information on Auxiliary files, see "Loading Auxiliary Files" on page 655.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **[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:** The 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**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Call Progress Tones File**  
**[CallProgressTonesFilename]** | Defines the name of the file containing the Call Progress Tones definitions.  
  For the ini file, the name must be enclosed by single apostrophes, for example, 'cpt_us.dat'.  
  For more information on how to create and load this file, refer to DConvert Utility User's Guide.  
  **Note:** For the parameter to take effect, a device reset is required. |
| **Prerecorded Tones File**  
**[PrerecordedTonesFilename]** | Defines the name of the file containing the Prerecorded Tones.  
  **Note:** For the parameter to take effect, a device reset is required. |
| **Dial Plan File**  
**[DialPlanFilename]** | Defines the name of the Dial Plan file. This file should be created using AudioCodes DConvert utility (refer to DConvert Utility User’s Guide).  
  For the ini file, the name must be enclosed by single apostrophes, for example, 'dial_plan.dat'. |
### 65.1.8 Automatic Update Parameters

The automatic update of software and configuration files parameters are described in the table below.

**Table 65-8: Automatic Update of Software and Configuration Files Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[UserInfoFileName]</td>
<td>Defines the name of the file containing the User Information data. For the ini file, the name must be enclosed by single apostrophes, for example, 'userinfo_us.dat'.</td>
</tr>
</tbody>
</table>
| [AutoUpdateCmpFile]        | Enables the Automatic Update mechanism for the cmp file.  
  - [0] = (Default) The Automatic Update mechanism doesn't apply to the cmp file.  
  - [1] = The Automatic Update mechanism includes the cmp file.  
  **Note:** For the parameter to take effect, a device reset is required. |
| [AutoUpdateFrequency]      | Defines the interval (in minutes) that the device waits between consecutive automatic updates. The default is 0 (i.e., the update at fixed intervals mechanism is disabled).  
  **Note:** For the parameter to take effect, a device reset is required. |
| [AutoUpdatePredefinedTime] | Defines schedules (time of day) for performing automatic updates. The format syntax of the parameter is 'hh:mm', where hh denotes the hour and mm the minutes. The value must be enclosed in single apostrophes. For example, '20:18'.  
  **Note:**  
  - For the parameter to take effect, a device reset is required.  
  - The actual update time is randomized by five minutes to reduce the load on the Web servers. |
| [AupdHttpUserAgent]        | Defines the information sent in the HTTP User-Agent header in the HTTP Get requests sent by the device to the provisioning server for the Automatic Update mechanism.  
  The valid value is a string of up to 511 characters. The information can include any user-defined string or the following string variable tags (case-sensitive):  
  - <NAME>: product name, according to the installed License Key  
  - <MAC>: device’s MAC address  
  - <VER>: software version currently installed on the device, e.g., "7.00.200.001"  
  - <CONF>: configuration version, as configured by the ini file parameter, INIFileVersion or CLI command, configuration-version  
  The device automatically populates these tag variables with actual values in the sent header. By default, the device sends the following in the User-Agent header: |

---

**User's Manual** 844  
Document #: LTRT-41877
### Parameter Configuration Reference

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>auto-firmware</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>aupd-verify-cert</strong></td>
<td>Determines whether the Automatic Update mechanism verifies the TLS certificate received from the provisioning server when the connection is HTTPS.</td>
</tr>
</tbody>
</table>
|                                  | - **[AUPDVerifyCertificates]** 0 = Disable (default)  
|                                  | 1 = Enables TLS certificate verification when the connection with the provisioning server is based on HTTPS. The device verifies the authentication of the certificate received from the provisioning server. The device authenticates the certificate against its trusted root certificate store (see "Configuring SSL/TLS Certificates" on page 105) and if ok, allows communication with the provisioning server. If authentication fails, the device denies communication (i.e., handshake fails). |
| **[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.                                                      |
| **crc-check regular**            | Enables the device to perform cyclic redundancy checks (CRC) on downloaded configuration files 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. |

**Example:**

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

```
User-Agent: MyWorld-Mediant;7.00.200.001(00908F1DD0D3)
```

**Note:**

- The variable tags are case-sensitive.
- If you configure the parameter with the `<CONF>` variable tag, you must reset the device with a save-to-flash for your settings to take effect.
- The tags can be defined in any order.
- The tags must be defined adjacent to one another (i.e., no spaces).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| If the check values are different, the device installs the downloaded file and applies the new configuration settings.  
  - [0] = (Default) Disable - the device does not perform CRC and installs the downloaded file regardless.  
  - [1] = Enable CRC for the entire file, including line order (i.e., same text must be on the same lines). If there are differences between the files, the device installs the downloaded file. If there are no differences, the device discards the newly downloaded file.  
  - [2] = Enable CRC for individual lines only. Same as option [1], except that the CRC ignores the order of lines (i.e., same text can be on different lines). |                                                                                                                                                                                                           |
| 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.  
  **Note:**  
  - 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 the parameter set to 1 is loaded.  
  **Note:** If you use the parameter in an ini file for periodic automatic provisioning with non-HTTP (e.g., TFTP) and without CRC, the device resets upon every file download. |                                                                                                                                                                                                           |

**Software/Configuration File URL Path for Automatic Update Parameters**  
CLI path: configure system > automatic-update

| firmware [CmpFileURL] | Defines the name of the cmp file and the URL address (IP address or FQDN) of the server on which the file is located.  
  For example: http://192.168.0.1/filename  
  **Note:**  
  - For the parameter to take effect, a device reset is required.  
  - When the 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. |                                                                                                                                                                                                           |
<p>| voice-configuration [IniFileURL] | Defines the name of the ini file and the URL address (IP address or FQDN) of the server on which the file is located. |                                                                                                                                                                                                           |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td><a href="http://192.168.0.1/filename">http://192.168.0.1/filename</a></td>
</tr>
<tr>
<td></td>
<td><a href="http://192.8.77.13/config">http://192.8.77.13/config</a>_&lt;MAC&gt;.ini</td>
</tr>
<tr>
<td></td>
<td>https://&lt;username&gt;:&lt;password&gt;@&lt;IP address&gt;/&lt;file name&gt;</td>
</tr>
<tr>
<td>Note:</td>
<td>For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td></td>
<td>When using HTTP or HTTPS, the date and time of the ini file are validated. Only more recently dated ini files are loaded.</td>
</tr>
<tr>
<td></td>
<td>The case-sensitive string, &quot;&lt;MAC&gt;&quot; can be used in the file name for instructing the device to replace it with the device's MAC address. For more information, see &quot;MAC Address Placeholder in Configuration File Name&quot; on page 697. This option allows the loading of specific configurations for specific devices.</td>
</tr>
<tr>
<td></td>
<td>The maximum length of the URL address is 99 characters.</td>
</tr>
<tr>
<td>cli-script &lt;URL&gt;</td>
<td>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.</td>
</tr>
<tr>
<td>[AUPDCliScriptURL]</td>
<td>Note: The case-sensitive string, &quot;&lt;MAC&gt;&quot; can be used in the file name for instructing the device to replace it with the device's MAC address. For more information, see MAC Address Placeholder in Configuration File Name on page 697.</td>
</tr>
<tr>
<td>startup-script &lt;URL&gt;</td>
<td>Defines the URL address of the server where the CLI Startup Script file containing the device's configuration is located. This file is used for automatic provisioning.</td>
</tr>
<tr>
<td>[AUPDStartupScriptURL]</td>
<td>Note: The case-sensitive string, &quot;&lt;MAC&gt;&quot; can be used in the file name for instructing the device to replace it with the device's MAC address. For more information, see MAC Address Placeholder in Configuration File Name on page 697.</td>
</tr>
<tr>
<td>prerecorded-tones</td>
<td>Defines the name of the Prerecorded Tones (PRT) file and the URL address (IP address or FQDN) of the server on which the file is located.</td>
</tr>
<tr>
<td>Note:</td>
<td>The maximum length of the URL address is 99 characters.</td>
</tr>
<tr>
<td>call-progress-tones</td>
<td>Defines the name of the CPT file and the URL address (IP address or FQDN) of the server on which the file is located.</td>
</tr>
<tr>
<td>Note:</td>
<td>The maximum length of the URL address is 99 characters.</td>
</tr>
<tr>
<td>tls-root-cert</td>
<td>Defines the name of the TLS trusted root certificate file and the URL address of the server on which the file is located.</td>
</tr>
<tr>
<td>[TLSRootFileUrl]</td>
<td>Note: For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>tls-cert</td>
<td>Defines the name of the TLS certificate file and the URL address of the server on which the file is located.</td>
</tr>
<tr>
<td>[TLSCertFileUrl]</td>
<td>Note: For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>tls-private-key</td>
<td>Defines the URL address of the server on which the TLS private key file is located.</td>
</tr>
<tr>
<td>[TLSPkeyFileUrl]</td>
<td></td>
</tr>
</tbody>
</table>
### 65.2 Networking Parameters

This subsection describes the device's networking parameters.

#### 65.2.1 Ethernet Parameters

The Ethernet parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Physical Ports Table</strong></td>
<td>The table configures the physical Ethernet ports. The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>Physical Ports configure network &gt; physical-port [PhysicalPortsTable]</td>
<td>[ PhysicalPortsTable ]</td>
</tr>
<tr>
<td></td>
<td>FORMAT PhysicalPortsTable_Index = PhysicalPortsTable_Port, PhysicalPortsTable_Mode, PhysicalPortsTable_SpeedDuplex,</td>
</tr>
<tr>
<td></td>
<td>PhysicalPortsTable_PortDescription, PhysicalPortsTable_GroupMember, PhysicalPortsTable_GroupStatus;</td>
</tr>
<tr>
<td></td>
<td>[ ]PhysicalPortsTable ]</td>
</tr>
<tr>
<td></td>
<td>For more information, see Configuring Physical Ethernet Ports on page 130.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Ethernet Groups Table</strong></th>
<th>Defines the transmit (Tx) and receive (Rx) settings for the Ethernet port groups. The format of the ini file table parameter is:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Groups configure network &gt; ether-group [EtherGroupTable]</td>
<td>[EtherGroupTable]</td>
</tr>
<tr>
<td></td>
<td>FORMAT EtherGroupTable_Index = EtherGroupTable_Group, EtherGroupTable_Mode, EtherGroupTable_SpeedDuplex, EtherGroupTable_PortDescription;</td>
</tr>
<tr>
<td></td>
<td>[ ]EtherGroupTable ]</td>
</tr>
</tbody>
</table>

**Parameter**

<table>
<thead>
<tr>
<th>user-info [UserInfoFileURL]</th>
<th>Defines the name of the User Information file and the URL address (IP address or FQDN) of the server on which the file is located. For example: http://server_name/file, https://server_name/file Note: The maximum length of the URL address is 99 characters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>feature-key [FeatureKeyURL]</td>
<td>Defines the name of the License Key file and the URL address of the server on which the file is located.</td>
</tr>
<tr>
<td>template-url [TemplateUrl]</td>
<td>Defines the URL address in the File Template for automatic updates, of the provisioning server on which the files to download are located. For more information, see &quot;File Template for Automatic Provisioning&quot; on page 698.</td>
</tr>
<tr>
<td>template-files-list [AupdFilesList]</td>
<td>Defines the list of file types in the File Template for automatic updates, to download from the provisioning server. For more information, see &quot;File Template for Automatic Provisioning&quot; on page 698.</td>
</tr>
<tr>
<td>web-favicon [WebFaviconFileUrl]</td>
<td>Defines the name of the favicon image file and the URL address of the server on which the file is located. This is used for the Automatic Update feature. For more information, see Customizing the Favicon on page 70.</td>
</tr>
</tbody>
</table>
### 65.2.2 Multiple VoIP Network Interfaces and VLAN Parameters

The IP network interfaces and VLAN parameters are described in the table below.

#### Table 65-10: IP Network Interfaces and VLAN Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IP Interfaces Table</strong></td>
<td></td>
</tr>
<tr>
<td>IP Interfaces configure network &gt; interface network-if [InterfaceTable]</td>
<td>The table configures IP network interfaces. 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 more information, see &quot;Configuring IP Network Interfaces&quot; on page 137.</td>
</tr>
<tr>
<td><strong>VLAN Parameters</strong></td>
<td></td>
</tr>
</tbody>
</table>
| [EnableNTPasOAM]              | Defines the application type for Network Time Protocol (NTP) services.  
  - [1] = OAMP (default)  
  - [0] = Control  
  **Note:** For the parameter to take effect, a device reset is required. |
65.2.3 Routing Parameters

The IP network routing parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send ICMP Unreachable Messages</td>
<td>Enables sending of ICMP Unreachable messages.</td>
</tr>
<tr>
<td>Configure network &gt; network-settings &gt; icmp-disable-unreachable [DisableICMPUnreachable]</td>
<td>▪ [0] Enable = (Default) Device sends these messages.</td>
</tr>
<tr>
<td>▪ [1] Disable = Device does not send these messages.</td>
<td></td>
</tr>
<tr>
<td>Send and Receive ICMP Redirect Messages</td>
<td>Enables sending and receiving of ICMP Redirect messages.</td>
</tr>
<tr>
<td>Configure network &gt; network-settings &gt; icmp-disable-redirec [DisableICMPRedirects]</td>
<td>▪ [0] Enable = (Default) Device sends and accepts these messages.</td>
</tr>
<tr>
<td>▪ [1] Disable = Device rejects these messages and also does not send them.</td>
<td></td>
</tr>
</tbody>
</table>

Static Routes Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static Routes</td>
<td>Defines up to 30 static IP routes for the device.</td>
</tr>
<tr>
<td>Configure network &gt; static [StaticRouteTable]</td>
<td>The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>[ StaticRouteTable ]</td>
<td>FORMAT StaticRouteTable_Index = StaticRouteTable_DeviceName, StaticRouteTable_Destination, StaticRouteTable_PrefixLength, StaticRouteTable_Gateway, StaticRouteTable_Description; [ StaticRouteTable ]</td>
</tr>
<tr>
<td>For a description of the parameter, see &quot;Configuring Static IP Routes&quot; on page 144.</td>
<td></td>
</tr>
</tbody>
</table>

65.2.4 Quality of Service Parameters

The Quality of Service (QoS) parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer-2 Class Of Service (CoS) Parameters (VLAN Tag Priority Field)</td>
<td>The table configures DiffServ-to-VLAN Priority mapping. For each packet sent to the LAN, the VLAN Priority of the packet is set according to the DiffServ value in the IP header of the packet.</td>
</tr>
<tr>
<td>DiffServ Table</td>
<td>The format of this ini file is as follows:</td>
</tr>
<tr>
<td>Configure network &gt; qos vlan-mapping [DiffServToVlanPriority]</td>
<td>[ DiffServToVlanPriority ]</td>
</tr>
<tr>
<td>FORMAT DiffServToVlanPriority_Index = DiffServToVlanPriority_DiffServ, DiffServToVlanPriority_VlanPriority; [ DiffServToVlanPriority ]</td>
<td></td>
</tr>
<tr>
<td>For example:</td>
<td>DiffServToVlanPriority 0 = 46, 6;</td>
</tr>
<tr>
<td>DiffServToVlanPriority 1 = 40, 6;</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DiffServToVlanPriority 2 = 26, 4;</td>
<td>For more information, see Configuring Quality of Service on page 155.</td>
</tr>
<tr>
<td>DiffServToVlanPriority 3 = 10, 2;</td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
</tbody>
</table>

**Layer-3 Class of Service (TOS/DiffServ) Parameters**

**CLI path:** configure network > qos application-mapping

- **Media Premium QoS media-qos [PremiumServiceClassMediaDiffServ]**
  - Global parameter defining the DiffServ value for Premium Media CoS content.
  - You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IPDiffServ). For a detailed description of the parameter and To configure the functionality, see "Configuring IP Profiles" on page 417.
  - **Note:** If the functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.

- **Control Premium QoS control-qos [PremiumServiceClassControlDiffServ]**
  - Global parameter defining the DiffServ value for Premium Control CoS content (Call Control applications).
  - You can also configure the functionality per specific calls, using IP Profiles (IpProfile_SigIPDiffServ). For a detailed description of the parameter and To configure the functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417.
  - **Note:** If the functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.

- **Gold QoS gold-qos [GoldServiceClassDiffServ]**
  - Defines the DiffServ value for the Gold CoS content (Streaming applications).
  - The valid range is 0 to 63. The default is 26.

- **Bronze QoS bronze-qos [BronzeServiceClassDiffServ]**
  - Defines the DiffServ value for the Bronze CoS content (OAMP applications).
  - The valid range is 0 to 63. The default is 10.

### 65.2.5 NAT Parameters

The Network Address Translation (NAT) parameters are described in the table below.

**Table 65-13: NAT Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAT Traversal configuration voip &gt; media</td>
<td>Enables the NAT traversal feature for media when the device communicates</td>
</tr>
<tr>
<td>settings &gt; disable-NAT-traversal</td>
<td>with UAs located behind NAT.</td>
</tr>
<tr>
<td></td>
<td>• [0] Enable NAT Option = NAT traversal is performed only if the UA</td>
</tr>
<tr>
<td></td>
<td>is located behind NAT:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| [NATMode]                  | ✓ UA behind NAT: The device sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA.  
                             ✓ UA not behind NAT: The device sends the packets to the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message.  
                             Note: If the SIP session is established (ACK) and the device (not the UA) sends the first packet, it sends it to the address obtained from the SIP message and only after the device receives the first packet from the UA does it determine whether the UA is behind NAT.  
                             ▪ [1] Disable NAT = (Default) The device considers the UA as not located behind NAT and sends media packets to the UA using the IP address:port specified in the SDP 'c=' line (Connection) of the first received SIP message.  
                             ▪ [2] Force NAT = The device always considers the UA as behind NAT and sends the media packets to the IP address:port obtained from the source address of the first media packet received from the UA. The device only sends packets to the UA after it receives the first packet from the UA (to obtain the IP address).  
                             ▪ [3] NAT By Signaling = The device identifies whether or not the UA is located behind NAT based on the SIP signaling. The device assumes that if signaling is behind NAT that the media is also behind NAT, and vice versa. If located behind NAT, the device sends media as described in option [2] Force NAT; if not behind NAT, the device sends media as described in option [1] Disable NAT. This option is applicable only to SBC calls. If the parameter is configured to this option, Gateway calls use option [0] Enable NAT Option, by default.  
                             For more information on NAT traversal, see 'First Incoming Packet Mechanism' on page 152.                                                                 |

| [NATBindingDefaultTimeout] | The device sends SNMP keep-alive traps periodically - every 9/10 of the time configured by the parameter (in seconds). Therefore, the parameter is applicable only if the SendKeepAliveTrap parameter is set to 1.  
                             The parameter is used to allow SNMP communication with AudioCodes EMS management platform, located in the WAN, when the device is located behind NAT. It is needed to keep the NAT pinhole open for the SNMP messages sent from EMS to the device.  
                             The valid range is 0 to 2,592,000. The default is 30.  
                             **Note:** For the parameter to take effect, a device reset is required.                                                                 |

SIP NAT Detection  
configure voip > sip-definition advanced-settings > sip-nat-detect  
[SIPNatDetection]  
Enables the device to detect whether the incoming INVITE message is sent from an endpoint located behind NAT.  
▪ [0] Disable = Disables the device's NAT Detection mechanism. Incoming SIP messages are processed as received from endpoints that are not located behind NAT and sent according to the SIP standard.  
▪ [1] Enable (default) = Enables the device's NAT Detection mechanism.
65.2.6 DNS Parameters

The Domain name System (DNS) parameters are described in the table below.

Table 65-14: DNS Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal DNS Table</td>
<td>The table defines the internal DNS table for resolving host names into IP addresses.</td>
</tr>
<tr>
<td>configure network &gt; dns</td>
<td>The format of the ini file table parameter is:</td>
</tr>
<tr>
<td>dns-to-ip</td>
<td>[DNS2IP]</td>
</tr>
<tr>
<td>[DNS2IP]</td>
<td>FORMAT Dns2Ip_Index = Dns2Ip_DomainName, Dns2Ip_FirstIpAddress, Dns2Ip_SecondIpAddress, Dns2Ip_ThirdIpAddress;</td>
</tr>
<tr>
<td>For example:</td>
<td>Dns2Ip 0 = DnsName, 1.1.1.1, 2.2.2.2, 3.3.3.3, ;</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Configuring the Internal DNS Table” on page 159.</td>
</tr>
</tbody>
</table>

Internal SRV Table

The table defines the internal SRV table for resolving host names into DNS A-Records. Three different A-Records can be assigned to a host name. Each A-Record contains the host name, priority, weight, and port. The format of the ini file table parameter is:

| Internal SRV Table               | The format of the ini file table parameter is:                               |
| configure network > dns          | [SRV2IP]                                                                     |
| 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 more information, see “Configuring the Internal SRV Table” on page 160.  |

65.2.7 DHCP Parameters

The Dynamic Host Control Protocol (DHCP) parameters are described in the table below.

Table 65-15: DHCP Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable DHCP</td>
<td>Enables DHCP client functionality.</td>
</tr>
<tr>
<td>[DHCPEnable]</td>
<td>[0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>[1] Enable</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td>- For the parameter to take effect, a device reset is required.</td>
<td></td>
</tr>
<tr>
<td>- For a detailed description of DHCP, see &quot;DHCP-based Provisioning&quot; on page 693.</td>
<td></td>
</tr>
<tr>
<td>- The parameter is a &quot;hidden&quot; 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.</td>
<td></td>
</tr>
</tbody>
</table>

[DHCP120OptionMode] Enables the acceptance of DHCP Option 120 in DHCP responses sent by a DHCP server.
- [0] = DHCP Option 120 is not supported and ignored if received in the DHCP response.
- [1] = (Default) DHCP Option 120 is supported and if received, the device adds the SIP server information to the Proxy Set.

[DHCPSpeedFactor] Defines the device's DHCP renewal speed for a leased IP address from a DHCP server.
- [0] = Disable
- [1] = (Default) Normal
When set to 0, the DHCP lease renewal is disabled. Otherwise, the renewal time is divided by this factor. Some DHCP-enabled routers perform better when set to 4.
**Note:** For the parameter to take effect, a device reset is required.

**DHCP Servers Table**

DHCP Servers Table
configure network > dhcp server <index>

[DhcpServer] Defines the device’s embedded DHCP server.
The format of the ini file table parameter is as follows:

```
[ Dhcpserver ]
FORMAT Dhcpserver_Index = Dhcpserver_InterfaceName,
Dhcpserver_StartIPaddress, Dhcpserver_EndIPaddress,
Dhcpserver_SubnetMask, Dhcpserver_LeaseTime,
Dhcpserver_DNSServer1, Dhcpserver_DNSServer2,
Dhcpserver_NetbiosNameServer, Dhcpserver_NetbiosNodeType,
Dhcpserver_NTPServer1, Dhcpserver_NTPServer2,
Dhcpserver_TimeOffset, Dhcpserver_TftpServer,
Dhcpserver_BootFileName, Dhcpserver_ExpandBootfileName,
Dhcpserver_OverrideRouter, Dhcpserver_SipServer,
Dhcpserver_SipServerType;
[ /Dhcpserver ]
```

For more information, see Configuring the Device's DHCP Server.

**DHCP Vendor Class Table**

DHCP Vendor Class table
configure network > dhcp-server vendor-class

[DhcpVendorClass] Defines Vendor Class Identifier (VCI) names (DHCP Option 60) for the device's DHCP server. Only if the DHCPDiscover request message, received from the DHCP client, contains this value does the device provide DHCP services.
The format of the ini file table parameter is as follows:

```
[ Dhcpvendorclass ]
FORMAT DhcpVendorclass_Index = DhcpVendorclass_DhcpserverIndex,
DhcpVendorclass_VendorClassId;
[ /Dhcpvendorclass ]
```
### 65.2.8 NTP and Daylight Saving Time Parameters

The Network Time Protocol (NTP) and daylight saving time parameters are described in the table below.

#### Table 65-16: NTP and Daylight Saving Time Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NTP Parameters</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>Primary NTP Server Address</td>
<td>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).</td>
</tr>
<tr>
<td>secondary-server</td>
<td>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.</td>
</tr>
<tr>
<td>NTP Update Interval</td>
<td>Defines the time interval (in seconds) that the NTP client requests for...</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| update-interval [NTPUpdateInterval]           | a time update. The default interval is 86400 (i.e., 24 hours). The range is 0 to 214783647.  
  **Note:** It is not recommend to set the parameter to beyond one month (i.e., 2592000 seconds). |
| NTP Authentication Key Identifier auth-key-id [NtpAuthKeyId] | Defines the NTP authentication key identifier for authenticating NTP messages. The identifier must match the value configured on the NTP server. The NTP server may have several keys configured for different clients; this number identifies which key is used. The valid value is 1 to 65535. The default is 0 (i.e., no authentication is done). |
| NTP Authentication Secret Key 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. |

**Regional Clock and Daylight Saving Time Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| UTC Offset configure system > clock > utc-offset [NTPServerUTCOffset] | Defines the Universal Time Coordinate (UTC) offset (in seconds) from the local time. The valid range is -43200 to 43200. The default is 0.  
  **Note:** The offset setting is applied only on the hour. For example, if you configure the parameter at 15:42, the device applies the setting only at 16:00. |
| Daylight Saving Time configure system > clock > summer-time > summer-time [DayLightSavingTimeEnable] | Enables daylight saving time (DST).  
  - [0] Disable (default)  
  - [1] Enable |
| Start Time / Day of Month Start configure system > clock > summer-time > start [DayLightSavingTimeStart] | Defines the date and time when DST 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. |
65. Configuration Parameters Reference

### 65.3 Debugging and Diagnostics Parameters

This subsection describes the device's debugging and diagnostic parameters.

#### 65.3.1 General Parameters

The general debugging and diagnostic parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [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).  
  
  **Note:** For the parameter to take effect, a device reset is required. |
| Delay After Reset [sec] configure voip > sip-definition advanced-settings > delay-after-reset | 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 |

#### 65.3.2 SIP Test Call Parameters

The SIP Signaling Test Call parameters are described in the table below.
Table 65-18: SIP Test Call Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Call DTMF String</td>
<td>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 &quot;3212333&quot;. If no string is defined (empty), DTMF is not played.</td>
</tr>
<tr>
<td>Test Call ID</td>
<td>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. <strong>Note:</strong> The parameter is only for testing incoming calls destined to this prefix number.</td>
</tr>
</tbody>
</table>

Test Call Rules Table

<table>
<thead>
<tr>
<th>Test Call Rules</th>
<th>Defines Test Call rules.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMAT Test_Call_Index = Test_Call_EndpointURI, Test_Call_CalledURI, Test_Call_RouteBy, Test_Call_IPGroupName, Test_Call_DestAddress, Test_Call_DestTransportType, Test_Call_SIPInterfaceName, Test_Call_ApplicationType, Test_Call_CallParty, Test_Call_MaxChannels, Test_Call_CallDuration, Test_Call_CallsPerSecond, Test_Call_CallMode, Test_Call_CallDuration, Test_Call_CallParty, Test_Call_CallParameter, Test_Call_CallProfile, Test_Call_Call_BWProfile;</td>
<td></td>
</tr>
<tr>
<td>For more information, see &quot;Configuring Test Call Endpoints&quot; on page 811.</td>
<td></td>
</tr>
</tbody>
</table>

65.3.3 Syslog, CDR and Debug Parameters

The Syslog, CDR and debug parameters are described in the table below.

Table 65-19: Syslog, CDR and Debug Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Syslog</td>
<td>Determines whether the device sends logs and error messages (e.g., CDRs) generated by the device to a Syslog server.</td>
</tr>
<tr>
<td>[EnableSyslog]</td>
<td></td>
</tr>
<tr>
<td>EnableSyslog</td>
<td>[0] Disable (default)</td>
</tr>
<tr>
<td>[1] Enable</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td></td>
</tr>
<tr>
<td>- If you enable Syslog, you must enter an IP address of the Syslog server (using the SyslogServerIP parameter).</td>
<td></td>
</tr>
<tr>
<td>- Syslog messages may increase the network traffic.</td>
<td></td>
</tr>
<tr>
<td>- To configure Syslog SIP message logging levels, use the GwDebugLevel parameter.</td>
<td></td>
</tr>
<tr>
<td>Syslog Server IP</td>
<td>Defines the IP address (in dotted-decimal notation) of the computer</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>configure troubleshoot &gt; syslog &gt; syslog-ip [SyslogServerIP]</td>
<td>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.</td>
</tr>
<tr>
<td>Syslog Server Port configure troubleshoot &gt; syslog &gt; syslog-port [SyslogServerPort]</td>
<td>Defines the UDP port of the Syslog server. The valid range is 0 to 65,535. The default port is 514.</td>
</tr>
</tbody>
</table>
| CDR Server IP Address configure troubleshoot > cdr > 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. **Note:**  
  - 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). |
| CDR Report Level configure troubleshoot > cdr > cdr-report-level [CDRReportLevel] | Enables signaling-related CDRs to be sent to a Syslog server and defines 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.  
  **Note:**  
  - For the SBC application, the parameter enables only signaling-related CDRs. To enable media-related CDRs for SBC calls, use the MediaCDRReportLevel parameter.  
  - The CDR Syslog message complies with RFC 3164 and is identified by: Facility = 17 (local1) and Severity = 6 (Informational).  
  - This mechanism is active only when Syslog is enabled (i.e., the parameter EnableSyslog is set to 1). |
| Media CDR Report Level configure troubleshoot > cdr > media-cdr-rprt-level [MediaCDRReportLevel] | Enables media-related CDRs of SBC calls to be sent to a Syslog server and defines the call stage at which they are sent.  
  - [0] None = (Default) No media-related CDR is sent.  
  - [1] End Media = Sends a CDR only at the end of the call.  
  - [2] Start & End Media = Sends a CDR once the media starts. In some calls it may only be after the call is established, but in other calls the media may start at ringback tone. A CDR is also sent upon termination (end) of the media in the call.  
  - [3] Update & End Media = Sends a CDR when an update occurs in the media of the call. For example, a call starts and a ringback tone occurs, a re-INVITE is sent for a fax call and as a result, a CDR with the MediaReportType field set to "Update" is sent, as the media was changed from voice to T.38. A CDR is also sent |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Start &amp; 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. <strong>Note:</strong> To enable CDR generation as well as enable signaling-related CDRs, use the CDRReportLevel parameter.</td>
<td></td>
</tr>
<tr>
<td>Local Storage Max File Size</td>
<td>Defines the size (in kilobytes) of each stored CDR file. Once the file size is reached, the device creates a new file for subsequent CDRs, and so on. The valid value is 100 to 1024. The default is 1024.</td>
</tr>
<tr>
<td>Local Storage Max Number of Files</td>
<td>Defines the maximum number of stored CDR files. If the maximum number is reached, the device replaces (overwrites) the oldest created file with a subsequent new file, and so on. The valid value is 2 to 4096. The default is 5.</td>
</tr>
<tr>
<td>Local Storage File Creation Interval</td>
<td>Defines how often (in minutes) the device creates a new CDR file. For example, if configured to 60, it creates a new file every hour. This occurs even if the maximum configured file size has not been reached (see the CDRLocalMaxFileSize parameter). However, if the maximum configured file size has been reached and the interval configured by the parameter has not been reached, a new CDR file is created. The valid value is 2 to 1440. The default is 60.</td>
</tr>
</tbody>
</table>
| Debug Level | Enables Syslog debug reporting and logging level.  
- [0] No Debug = (Default) Debug is disabled and Syslog messages are not sent.  
- [1] Basic = Sends debug logs of incoming and outgoing SIP messages.  
- [5] Detailed = Sends debug logs of incoming and outgoing SIP message as well as many other logged processes. |
| EnableNonCallCdr | Enables creation of CDR messages for non-call SIP dialogs (such as SUBSCRIBE, OPTIONS, and REGISTER).  
- [0] = (Default) Disable  
- [1] = Enable |
| Syslog Optimization | 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. |
| Define the maximum size (in bytes) threshold of logged Syslog messages bundled into a single UDP packet, after which they are sent to a Syslog server. The valid value range is 0 to 1220 (where 0 indicates that no bundling occurs). The default is 1220.  
**Note:** The parameter is applicable only if the GWDebugLevel is enabled. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>parameter is enabled.</td>
<td></td>
</tr>
<tr>
<td>Syslog CPU Protection configure troubleshoot &gt; syslog &gt; syslog-cpu-protection [SyslogCpuProtection]</td>
<td>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).</td>
</tr>
<tr>
<td>▪ [0] Disable</td>
<td></td>
</tr>
<tr>
<td>▪ [1] Enable (default)</td>
<td></td>
</tr>
<tr>
<td>Debug Level High Threshold configure voip &gt; sip-definition settings &gt; debug-level-high-threshold [DebugLevelHighThreshold]</td>
<td>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:</td>
</tr>
<tr>
<td>▪ CPU usage equals threshold: Debug level is reduced one level.</td>
<td></td>
</tr>
<tr>
<td>▪ CPU usage is at least 5% greater than threshold: Debug level is reduced another level.</td>
<td></td>
</tr>
<tr>
<td>▪ CPU usage is 5 to 19% less than threshold: Debug level is increased by one level.</td>
<td></td>
</tr>
<tr>
<td>▪ CPU usage is at least 20% less than threshold: Debug level is increased by another level.</td>
<td></td>
</tr>
<tr>
<td>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).</td>
<td></td>
</tr>
<tr>
<td>Note: The device does not increase the debug level to a level that is higher than what you configured for the 'Debug Level' parameter.</td>
<td></td>
</tr>
<tr>
<td>Syslog Facility Number [SyslogFacility]</td>
<td>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.</td>
</tr>
<tr>
<td>▪ [16] = (Default) local use 0 (local0)</td>
<td></td>
</tr>
<tr>
<td>▪ [17] = local use 1 (local1)</td>
<td></td>
</tr>
<tr>
<td>▪ [18] = local use 2 (local2)</td>
<td></td>
</tr>
<tr>
<td>▪ [19] = local use 3 (local3)</td>
<td></td>
</tr>
<tr>
<td>▪ [20] = local use 4 (local4)</td>
<td></td>
</tr>
<tr>
<td>▪ [21] = local use 5 (local5)</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>▪ [22] = local use 6 (local6)</td>
<td></td>
</tr>
<tr>
<td>▪ [23] = local use 7 (local7)</td>
<td></td>
</tr>
</tbody>
</table>

CDR Syslog Sequence Number
configure system > cdr > cdr-seq-num
[CDRSyslogSeqNum]
Enables or disables the inclusion of the sequence number (S=) in CDR Syslog messages.
▪ [0] Disable
▪ [1] Enable (default)

Activity Types to Report via Activity Log Messages
configure troubleshoot > activity-log
[ActivityListToLog]
Defines the operations (activities) performed in the Web interface that are reported to a Syslog server.
▪ [pvc] Parameters Value Change = Changes made on-the-fly to parameters and tables, and Configuration file load. Note that the ini file parameter, EnableParametersMonitoring can also be used to set this option.
▪ [dr] Device Reset = Resetting of the device through the Maintenance Actions page. Note: For this option to take effect, a device reset is required.
▪ [fb] Flash Memory Burning = Saving configuration with burn to flash (in the Maintenance Actions page).
▪ [swu] Device Software Update = Software updates (i.e., loading of cmp file) through the Software Upgrade Wizard.
▪ [ard] Access to Restricted Domains = Access to restricted Web pages:
  ✓ (1) ini parameters (AdminPage)
  ✓ (2) General Security Settings
  ✓ (3) Configuration File
  ✓ (5) License Key
  ✓ (7) Access List
  ✓ (8) Web User Accounts
▪ [naa] Non-Authorized Access = Attempts to log in to the Web interface with a false or empty username or password.
▪ [spc] Sensitive Parameters Value Change = Changes made to "sensitive" parameters:
  ✓ (1) IP Address
  ✓ (2) Subnet Mask
  ✓ (3) Default Gateway IP Address
  ✓ (4) ActivityListToLog
▪ [ll] Login and Logout = Web login and logout attempts.
▪ [cli] = CLI commands entered by the user.
▪ [ae] Action Executed = Logs user actions that are not related to parameter changes. The actions can include, for example, file uploads, file delete, lock-unlock maintenance actions, LDAP clear cache, register-unregister, and start-stop trunk. In the Web, these actions are typically done by clicking a button (e.g., the LOCK button).

Note: For the ini file parameter, enclose values in single quotation marks, for example: ActivityListToLog = 'pvc', 'afl', 'dr', 'fb', 'swu', 'ard', 'naa', 'spc'.

[EnableParametersMonitoring]
Enables the monitoring, through Syslog messages, of parameters that are modified on-the-fly.
▪ [0] = (Default) Disable
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Debug Recording Destination IP</td>
<td>Defines the IP address of the server for capturing debug recording.</td>
</tr>
<tr>
<td>configure troubleshoot &gt; logging settings &gt; dbg-rec-dest-ip [DebugRecordingDestIP]</td>
<td></td>
</tr>
<tr>
<td>Debug Recording Destination Port</td>
<td>Defines the UDP port of the server for capturing debug recording. The default is 925.</td>
</tr>
<tr>
<td>configure troubleshoot &gt; logging settings &gt; dbg-rec-dest-port [DebugRecordingDestPort]</td>
<td></td>
</tr>
<tr>
<td>Enable Core Dump [EnableCoreDump]</td>
<td>Enables the automatic generation of a Core Dump file upon a device crash.</td>
</tr>
<tr>
<td></td>
<td>• [0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Enable</td>
</tr>
<tr>
<td></td>
<td>Note: For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>Core Dump Destination IP [CoreDumpDestIP]</td>
<td>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.</td>
</tr>
<tr>
<td>Logging Filters Table</td>
<td>The table defines log filtering rules for Syslog messages and debug recordings.</td>
</tr>
<tr>
<td>configure troubleshoot &gt; logging logging-filters [LoggingFilters]</td>
<td>The format of the ini file table parameter is:</td>
</tr>
<tr>
<td></td>
<td>FORMAT LoggingFilters_Index = LoggingFilters_FilterType,</td>
</tr>
<tr>
<td></td>
<td>LoggingFilters_Value, LoggingFilters_LogDestination,</td>
</tr>
<tr>
<td></td>
<td>LoggingFilters_CaptureType, LoggingFilters_Mode;</td>
</tr>
<tr>
<td></td>
<td>[ \LoggingFilters ]</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Configuring Log Filter Rules&quot; on page 787.</td>
</tr>
<tr>
<td>SBC CDR Format Table</td>
<td>The table defines CDR customization rules for SBC calls.</td>
</tr>
<tr>
<td>configure troubleshoot &gt; cdr &gt; cdr-format sbc-cdr-format [SBCCDRFormat]</td>
<td>The format of the ini file table parameter is:</td>
</tr>
<tr>
<td></td>
<td>FORMAT SBCCDRFormat_Index = SBCCDRFormat_CDRTYPE,</td>
</tr>
<tr>
<td></td>
<td>SBCCDRFormat_FieldType, SBCCDRFormat_Title,</td>
</tr>
<tr>
<td></td>
<td>SBCCDRFormat_RadiusType, SBCCDRFormat_RadiusID;</td>
</tr>
<tr>
<td></td>
<td>[ \SBCCDRFormat ]</td>
</tr>
<tr>
<td></td>
<td>For more information, see Customizing CDRs for SBC Calls on page 770.</td>
</tr>
</tbody>
</table>
65.3.4 Resource Allocation Indication Parameters

The Resource Allocation Indication (RAI) parameters are described in the table below.

Table 65-20: RAI Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[EnableRAI]</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>- [0] = (Default) Disable</td>
</tr>
<tr>
<td></td>
<td>- [1] = Enable</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>[RAIHighThreshold]</td>
<td>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.</td>
</tr>
<tr>
<td>[RAILowThreshold]</td>
<td>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%.</td>
</tr>
<tr>
<td>[RAILoopTime]</td>
<td>Defines the time interval (in seconds) that the device periodically checks call resource availability.</td>
</tr>
<tr>
<td></td>
<td>The valid range is 1 to 200. The default is 10.</td>
</tr>
</tbody>
</table>

65.4 HA Parameters

The High Availability (HA) parameters are described in the table below.

Table 65-21: HA Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA Device Name</td>
<td>Defines a name for the active device, which is displayed on the Home page to indicate the active device.</td>
</tr>
<tr>
<td>configure network &gt; high-availability &gt; unit-id-name [HAUnitIdName]</td>
<td>The valid value is a string of up to 128 characters. The default value is &quot;Device 1&quot;.</td>
</tr>
<tr>
<td>Redundant HA Device Name</td>
<td>Defines a name for the redundant device, which is displayed on the Home page to indicate the redundant device.</td>
</tr>
<tr>
<td>configure network &gt; high-availability &gt; redundant-unit-id-name</td>
<td>The valid value is a string of up to 128 characters. The default value is &quot;Device 2&quot;.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| HA Remote Address | Defines the Maintenance interface address of the redundant device in the HA system. By default, no value is defined.  
**Note:** For the parameter to take effect, a device reset is required. |
| Preempt Mode | Enables HA switchover based on HA priority.  
- [0] Disable = (Default) A switchover over to the redundant device is done only if a failure occurs in the currently active device.  
- [1] Enable = A switchover over to the redundant device is done if a failure occurs in the currently active device. However, a switchover to the device with the highest priority (configured by the HAPriority parameter) occurs whenever the device recovers from a failure. Therefore, whenever possible, the highest priority device is the active one.  
For more information on the HA switchover mechanism, see Device Switchover upon Failure on page 599.  
**Note:** For the parameter to take effect, a device reset is required. |
| Preempt Priority | Defines the priority of the active device used in the HA Preempt mechanism.  
The valid value is 1 (lowest priority) to 10 (highest priority). The default is 5.  
**Note:**  
- For the parameter to take effect, a device reset is required.  
- The parameter is applicable only if you configure the 'Preempt Mode' parameter to Enable.  
- You must configure each device in the HA system with different parameter values (priorities). |
| Redundant Preempt Priority | Defines the priority of the redundant device used in the HA Preempt mechanism.  
The valid value is 1 (lowest priority) to 10 (highest priority). The default is 5.  
**Note:**  
- For the parameter to take effect, a device reset is required.  
- The parameter is applicable only if you configure the 'Preempt Mode' parameter to Enable.  
- You must configure each device in the HA system with different parameter values (priorities). |
| HA Network Reachability Parameters | Enables the pinging of an active IP network destination in HA mode to test reachability from one of the device's IP network interfaces. If no reply is received from a ping and the previous ping was successful, a switchover occurs to the redundant device.  
- [0] Disable (default)  
- [1] Enable |
<p>| Destination Address | Defines the IP address of the destination that the device pings. The default is 0.0.0.0. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination</td>
<td>Defines the device’s IP network interface from where the ping is sent. The valid value is the name of the IP interface as configured in the 'Interface Name' field of the IP Interfaces table. By default, no IP network is defined.</td>
</tr>
<tr>
<td>Source Interface Name</td>
<td>configure network &gt; high-availability &gt; net-mon-source-interface [HAPingSourceIfName] Defines the device’s IP network interface from where the ping is sent. The valid value is the name of the IP interface as configured in the 'Interface Name' field of the IP Interfaces table. By default, no IP network is defined.</td>
</tr>
<tr>
<td>Ping Timeout</td>
<td>configure network &gt; high-availability &gt; net-mon-ping-timeout [HAPingTimeout] Defines the timeout (in seconds) for which the ping request waits for a reply. The valid value is 1 to 60. The default is 1.</td>
</tr>
<tr>
<td>Ping Retries</td>
<td>configure network &gt; high-availability &gt; net-mon-ping-retries [HAPingRetries] Defines the number of ping requests that the device sends after no response is received from the destination, before the destination is declared unavailable. For example, if you specify 2, the destination is declared as down after three consecutive ping requests fail to evoke a response from the destination. The valid value is 0 to 100. The default 2.</td>
</tr>
</tbody>
</table>

65.5 Security Parameters

This subsection describes the device’s security parameters.

65.5.1 General Security Parameters

The general security parameters are described in the table below.

Table 65-22: General Security Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firewall Table</td>
<td>The table defines the device’s access list (firewall), which defines network traffic filtering rules. The format of the ini file table parameter is: [AccessList] FORMAT AccessList_Index = AccessList_Source_IP, AccessList_Source_Port, AccessList_PrefixLen, AccessList_Source_Port, AccessList_Start_Port, AccessList_End_Port, AccessList_Protocol, AccessList_Use_Specific_Interface, AccessList_Interface_ID, AccessList_Packet_Size, AccessList_Byte_Rate, AccessList_BYTE_Burst, AccessList_ALLOW_Type; [AccessList] For example: AccessList 10 = mgmt.customer.com, , , 32, 0, 80, tcp, 1, OAMP, 0, 0, 0, allow; AccessList 22 = 10.4.0.0, , , 16, 4000, 9000, any, 0, 0, 0, 0, block; In the example above, Rule #10 allows traffic from the host</td>
</tr>
</tbody>
</table>
### Media Latching

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inbound Media Latch Mode</td>
<td>Enables the Media Latching feature.</td>
</tr>
<tr>
<td>configure voip &gt; media settings &gt; inbound-media-latch-mode [InboundMediaLatchMode]</td>
<td></td>
</tr>
<tr>
<td>[InboundMediaLatchMode]</td>
<td></td>
</tr>
</tbody>
</table>

- **[0]** Strict = Device latches onto the first original stream (IP address:port). It does not latch onto any other stream during the session.

- **[1]** Dynamic = (Default) Device latches onto the first stream. If it receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) from a different source(s) and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other stream. If other packets of a different media type are received from the new stream, based on IP address and SSRC for RTCP/RTP and based on IP address only for T.38, the packet is accepted immediately. Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream.

- **[2]** Dynamic-Strict = Device latches onto the first stream. If it receives at least a minimum number of consecutive packets (configured by New<media type>StreamPackets) all from the same source which is different to the first stream and the device has not received packets from the current stream for a user-defined period (TimeoutToRelatch<media type>Msec), it latches onto the next packet received from any other stream. If other packets of different media type are received from the new stream based on IP address and SSRC for RTCP and based on IP address only for T.38, the packet is accepted immediately. Note: If a packet from the original (first latched onto) IP address:port is received at any time, the device latches onto this stream.

- **[3]** Strict-On-First = Typically used for NAT, where the correct IP address:port is initially unknown. The device latches onto the stream received in the first packet. The device does not change this stream unless a packet is later received from the original source.

**Note:** If you configure the parameter to [0] Strict, the device cannot perform NAT traversal. In this setup, configure the NATMode parameter to [1] Disable NAT.

### 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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>incoming stream.</strong></td>
<td>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.</td>
</tr>
<tr>
<td>New SRTP Stream Packets [NewSRTPStreamPackets]</td>
<td>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.</td>
</tr>
<tr>
<td>New SRTCP Stream Packets [NewSRTCPStreamPackets]</td>
<td>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.</td>
</tr>
<tr>
<td>Timeout To Relatch RTP [TimeoutToRelatchRTPMsec]</td>
<td>Defines a period (msec) during which if no packets are received from the current RTP session, the channel can re-latch onto another stream. The valid range is any value from 0. The default is 200.</td>
</tr>
<tr>
<td>Timeout To Relatch SRTP [TimeoutToRelatchSRTPMsec]</td>
<td>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.</td>
</tr>
<tr>
<td>Timeout To Relatch Silence [TimeoutToRelatchSilenceMsec]</td>
<td>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.</td>
</tr>
<tr>
<td>Timeout To Relatch RTCP [TimeoutToRelatchRTCPMsec]</td>
<td>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.</td>
</tr>
<tr>
<td>Fax Relay Rx/Tx Timeout [FaxRelayTimeoutSec]</td>
<td>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.</td>
</tr>
</tbody>
</table>

### 65.5.2 HTTPS Parameters

The Secure Hypertext Transport Protocol (HTTPS) parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| Secured Web Connection (HTTPS) configure system > web > secured-connection [HTTPSOnly] | Determines the protocol used to access the Web interface.  
- [0] HTTP and HTTPS (default).  
- [1] HTTPS Only = Unencrypted HTTP packets are blocked.  
**Note:** For the parameter to take effect, a device reset is required. |
### 65. Configuration Parameters Reference

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure system &gt; web &gt; https-port [HTTPSPort]</td>
<td>Defines the local Secured HTTPS port of the device. The parameter allows secure remote device Web management from the LAN. To enable secure Web management from the LAN, configure the desired port. The valid range is 1 to 65535 (other restrictions may apply within this range). The default port is 443. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
</tbody>
</table>
- [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.  
**Note:**  
- For the parameter to take effect, a device reset is required.  
- For a description on implementing client certificates, see "TLS for Remote Device Management" on page 118. |

#### 65.5.3 SRTP Parameters

The Secure Real-Time Transport Protocol (SRTP) parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
- [0] Disable (default)  
- [1] Enable  
**Note:** For the parameter to take effect, a device reset is required. |
| Master Key Identifier (MKI) Size configure voip > media security > srtp-tx-packet-mki-size [SRTPTxPacketMKISize] | Global parameter that defines the size (in bytes) of the Master Key Identifier (MKI) in SRTP Tx packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_MKISize). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417.  
**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| Symmetric MKI Negotiation configure voip > media security > symmetric-mki [EnableSymmetricMKI] | Global parameter that enables symmetric MKI negotiation. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableSymmetricMKI). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417.  
**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. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offered SRTP Cipher Suites</td>
<td>Defines the offered crypto suites (cipher encryption algorithms) for SRTP.</td>
</tr>
<tr>
<td>configure voip &gt; media security &gt; offer-srtp-cipher [SRTPofferedSuites]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• [0] All = (Default) All available crypto suites.</td>
</tr>
<tr>
<td>Note: The parameter also affects the selection of the crypto in the device's answer. For example, if the device receives an offer with two crypto lines containing HMAC_SHA1_80 and HMAC_SHA_32, it uses the HMAC_SHA_32 key in its SIP 200 OK response if the parameter is set to 2.</td>
<td></td>
</tr>
<tr>
<td>configure voip &gt; sbc settings &gt; sbc-dtls-mtu [SbcDtlsMtu]</td>
<td>Defines the maximum transmission unit (MTU) size for the DTLS handshake. The device does not attempt to send handshake packets that are larger than the configured value. Adjusting the MTU is useful when there are network constraints on the size of packets that can be sent. The valid value range is 228 to 1500. The default is 1500.</td>
</tr>
<tr>
<td>Disable Authentication On Transmitted RTP Packets</td>
<td>Enables authentication on transmitted RTP packets in a secured RTP session.</td>
</tr>
<tr>
<td>configure voip &gt; media security &gt; RTP-authentication-disable-tx [RTPAuthenticationDisableTx]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• [0] Enable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Disable</td>
</tr>
<tr>
<td>Disable Encryption On Transmitted RTP Packets</td>
<td>Enables encryption on transmitted RTP packets in a secured RTP session.</td>
</tr>
<tr>
<td>configure voip &gt; media security &gt; RTP-encryption-disable-tx [RTPEncryptionDisableTx]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• [0] Enable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Disable</td>
</tr>
<tr>
<td>Disable Encryption On Transmitted RTCP Packets</td>
<td>Enables encryption on transmitted RTCP packets in a secured RTP session.</td>
</tr>
<tr>
<td>configure voip &gt; media security &gt; RTCP-encryption-disable-tx [RTCPEncryptionDisableTx]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• [0] Enable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Disable</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition settings &gt; srtp-state-behavior-mode [ResetSRTPStateUponRekey]</td>
<td>Global parameter that enables synchronization of the SRTP state between the device and a server when a new SRTP key is generated upon a SIP session expire. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_ResetSRTPStateUponRekey). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see &quot;Configuring IP Profiles&quot; on page 417.</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
</tbody>
</table>
## 65.5.4 TLS Parameters

The Transport Layer Security (TLS) parameters are described in the table below.

**Table 65-25: TLS Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TLS Contexts Table</strong></td>
<td></td>
</tr>
<tr>
<td>TLS Contexts</td>
<td>Defines SSL/TLS certificates. The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>configure system &gt; tls #</td>
<td>[TLSContexts]</td>
</tr>
<tr>
<td>[TLSContexts]</td>
<td>FORMAT TLSContexts_Index = TLSContexts_Name, TLSContexts_TLSVersion, TLSContexts_DTLSVersion,</td>
</tr>
<tr>
<td></td>
<td>TLSContexts_ServerCipherString, TLSContexts_ClientCipherString, TLSContexts_RequireStrictCert, TLSContexts_OcspEnable, TLSContexts_OcspServerPrimary,</td>
</tr>
<tr>
<td></td>
<td>TLSContexts_OcspServerSecondary, TLSContexts_OcspServerPort, TLSContexts_OcspDefaultResponse, TLSContexts_DHKeySize; [ \TLSContexts ]</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Configuring TLS Certificate Contexts&quot; on page 105.</td>
</tr>
<tr>
<td>TLS Client Re-Handshake Interval</td>
<td>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).</td>
</tr>
<tr>
<td>configure network/security-</td>
<td></td>
</tr>
<tr>
<td>settings/tls-re-hndshk-int</td>
<td>[TLSReHandshakeInterval]</td>
</tr>
<tr>
<td>TLS Mutual Authentication</td>
<td>Defines the device’s mode of operation regarding mutual authentication and certificate verification for TLS connections.</td>
</tr>
<tr>
<td>[SIPSRequireClientCertificate]</td>
<td>• [0] Disable = (Default)</td>
</tr>
<tr>
<td></td>
<td>✓ Device acts as a client: Verification of the server’s certificate depends on the VerifyServerCertificate parameter.</td>
</tr>
<tr>
<td></td>
<td>✓ Device acts as a server: The device does not request the client certificate.</td>
</tr>
<tr>
<td></td>
<td>• [1] Enable =</td>
</tr>
<tr>
<td></td>
<td>✓ Device acts as a client: Verification of the server certificate is required to establish the TLS connection.</td>
</tr>
<tr>
<td></td>
<td>✓ Device acts as a server: The device requires the receipt and verification of the client certificate to establish the TLS connection.</td>
</tr>
<tr>
<td>Peer Host Name Verification Mode</td>
<td>Enables the device to verify the Subject Name of a TLS certificate received from SIP entities for authentication and establishing TLS connections.</td>
</tr>
<tr>
<td>[PeerHostNameVerificationMode]</td>
<td></td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>• [0] Disable (default).</td>
<td></td>
</tr>
<tr>
<td>• [1] Server Only = Verify Subject Name only when acting as a client for the TLS connection.</td>
<td></td>
</tr>
<tr>
<td>• [2] Server &amp; Client = Verify Subject Name when acting as a server or client for the TLS connection.</td>
<td></td>
</tr>
</tbody>
</table>

If the device receives a certificate from a SIP entity (IP Group) and the parameter is configured to **Server Only** or **Server & Client**, it attempts to authenticate the certificate based on the certificate's address.

The device searches for a Proxy Set that contains the same address (IP address or FQDN) as that specified in the certificate's SubjectAltName (Subject Alternative Names). For Proxy Sets with an FQDN, the device checks the FQDN itself and not the DNS-resolved IP addresses. If a Proxy Set is found with a matching address, the device establishes a TLS connection.

If a matching Proxy Set is not found, one of the following occurs:

- If the certificate's SubjectAltName is marked as "critical", the device rejects the call.
- If the SubjectAltName is not marked as "critical", the device checks if the FQDN in the certificate's Common Name (CN) of the SubjectName is the same as that configured for the TLSRemoteSubjectName parameter or for the Proxy Set. If they are the same, the device establishes a TLS connection; otherwise, the device rejects the call.

**Note:**

- If you configure the parameter to **Server & Client**, you also need to configure the SIPSRequireClientCertificate parameter to **Enable**.
- For FQDN, the certificate may use wildcards (*) to replace parts of the domain name.

### TLS Client Verify Server Certificate

**configure network/security-settings/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.

### TLS Remote Subject Name

**configure network/security-settings/tls-rmt-subs-name**

**[TLSRemoteSubjectName]**

Defines the Subject Name that is compared with the name defined in the remote side certificate when establishing TLS connections. If the SubjectAltName of the received certificate is not equal to any of the defined Proxies Host names/IP addresses and is not marked as 'critical', the Common Name (CN) of the Subject field is compared with this value. If not equal, the TLS connection is not established. If the CN uses a domain name, the certificate can also use wildcards (**) to replace parts of the domain name.

The valid range is a string of up to 49 characters.

**Note:** The parameter is applicable only if the parameter PeerHostNameVerificationMode is set to 1 or 2.

### TLS Expiry Check Start

**expiry-check-start**

Defines the number of days before the installed TLS server certificate is to expire at which the device must send a trap.
### 65.5.5 SSH Parameters

Secure Shell (SSH) parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable SSH Server</strong>&lt;br&gt;configure system &gt; cli-settings &gt; ssh [SSHServerEnable]</td>
<td>Enables the device’s embedded SSH server.&lt;br&gt;• [0] Disable (default)&lt;br&gt;• [1] Enable</td>
</tr>
<tr>
<td><strong>Server Port</strong>&lt;br&gt;configure system &gt; cli-settings &gt; ssh-port [SSHServerPort]</td>
<td>Defines the port number for the embedded SSH server.&lt;br&gt;Range is any valid port number. The default port is 22.</td>
</tr>
<tr>
<td><strong>SSH Admin Key</strong>&lt;br&gt;configure system &gt; cli-settings &gt; ssh-admin-key [SSHAdminKey]</td>
<td>Defines the RSA public key for strong authentication for logging in to the SSH interface (if enabled).&lt;br&gt;The value should be a base64-encoded string. The value can be a maximum length of 511 characters.&lt;br&gt;&lt;br&gt;&lt;strong&gt;Note:&lt;/strong&gt; The parameter is overridden by the SSH public key configured for a specific management user in the Local Users table (see Configuring Management User Accounts on page 73).</td>
</tr>
<tr>
<td><strong>Public Key</strong>&lt;br&gt;configure system &gt; cli-settings &gt; ssh-require-public-key [SSHRequirePublicKey]</td>
<td>Enables RSA public keys for SSH.&lt;br&gt;• [0] Disable = (Default) RSA public keys are optional if a value is configured for the parameter SSHAdminKey.&lt;br&gt;• [1] Enable = RSA public keys are mandatory.&lt;br&gt;&lt;br&gt;&lt;strong&gt;Note:&lt;/strong&gt; To define the key size, use the TLSPkeySize parameter.</td>
</tr>
<tr>
<td><strong>Max Payload Size</strong>&lt;br&gt;ssh-max-payload-size [SSHMaxPayloadSize]</td>
<td>Defines the maximum uncompressed payload size (in bytes) for SSH packets.&lt;br&gt;The valid value is 550 to 32768. The default is 32768.</td>
</tr>
<tr>
<td><strong>Max Binary Packet Size</strong>&lt;br&gt;configure system &gt; cli-settings &gt; ssh-max-binary-packet-size [SSHMaxBinaryPacketSize]</td>
<td>Defines the maximum packet size (in bytes) for SSH packets.&lt;br&gt;The valid value is 582 to 35000. The default is 35000.</td>
</tr>
<tr>
<td><strong>Maximum SSH Sessions</strong>&lt;br&gt;configure system &gt; cli-settings &gt; ssh-max-sessions [SSHMaxSessions]</td>
<td>Defines the maximum number of simultaneous SSH sessions.&lt;br&gt;The valid range is 1 to 5. The default is 5.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Enable Last Login Message</td>
<td>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.</td>
</tr>
<tr>
<td>Max Login Attempts</td>
<td>Defines the maximum SSH login attempts allowed for entering an incorrect password by an administrator before the SSH session is rejected. The valid range is 1 to 5. The default is 3. <strong>Note:</strong> The new setting takes effect only for new subsequent SSH connections.</td>
</tr>
</tbody>
</table>

### 65.5.6 IDS Parameters

The Intrusion Detection System (IDS) parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrusion Detection System (IDS) enable-ids [EnableIDS]</td>
<td>Enables the IDS feature.</td>
</tr>
<tr>
<td>ids-clear-period [IDSAAlarmClearPeriod]</td>
<td>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 &quot;quiet&quot; period must be at least twice the Threshold Window value. For example, if IDSAAlarmClearPeriod is set to 20 sec and the Threshold Window is set to 15 sec, the IDSAAlarmClearPeriod 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.</td>
</tr>
</tbody>
</table>

**IDS Policy Table**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDS Policy Table [IDSPolicy]</td>
<td>Defines IDS Policies. The format of the ini file parameter is: [ IDSPolicy ] FORMAT IDSPolicy_Index = IDSPolicy_Name, IDSPolicy_Description; [ \IDSPolicy ] For more information, see &quot;Configuring IDS Policies&quot; on page 172.</td>
</tr>
</tbody>
</table>

**IDS Rule Table**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDS Rule Table [IDSRule]</td>
<td>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,</td>
</tr>
</tbody>
</table>
### 65.5.7 OCSP Parameters

The Online Certificate Status Protocol (OCSP) parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Enable OCSP Server**<br>configure network > ocsp > enable<br>[OCSPEnable] | Enables or disables certificate checking using OCSP.  
- [0] Disable (default)  
- [1] Enable  
For a description of OCSP, see Configuring Certificate Revocation Checking (OCSP). |
| **Primary Server IP**<br>configure network > ocsp > server-ip<br>[OCSPServerIP] | Defines the IP address of the OCSP server.  
The default IP address is 0.0.0.0. |
| **Secondary Server IP**<br>configure network > ocsp > secondary-server-ip<br>[OCSPSecondaryServerIP] | Defines the IP address (in dotted-decimal notation) of the secondary OCSP server (optional).  
The default IP address is 0.0.0.0. |
| **Server Port**<br>configure network > ocsp > server-port<br>[OCSPServerPort] | Defines the OCSP server's TCP port number.  
The default port number is 2560. |
| **Default Response When Server Unreachable**<br>configure network > ocsp > default-response<br>[OCSPDefaultResponse] | Determines whether the device allows or rejects peer certificates when the OCSP server cannot be contacted.  
- [0] Reject (default)  
- [1] Allow |
65.6 Media Transcoding Cluster Parameter

The Media Transcoding Cluster parameters are described in the table below.

Table 65-29: Media Transcoding Cluster Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Manager Functionality</td>
<td>Enables the Media Transcoding Cluster feature.</td>
</tr>
<tr>
<td>configure network &gt; mtc</td>
<td></td>
</tr>
<tr>
<td>settings &gt; enable-mtc-sbc</td>
<td></td>
</tr>
<tr>
<td>[EnableMtcSbc]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• [0] Disable</td>
</tr>
<tr>
<td></td>
<td>• [1] Enable (default)</td>
</tr>
<tr>
<td>Note:</td>
<td>For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>Cluster Graceful Timeout</td>
<td>Defines the graceful period (in seconds) for various maintenance actions performed by the Cluster Manager on the Media Transcoders.</td>
</tr>
<tr>
<td>configure network &gt; mtc</td>
<td></td>
</tr>
<tr>
<td>settings &gt; graceful-timeout</td>
<td></td>
</tr>
<tr>
<td>[MtcGracefulTimeout]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The valid value is 0 to 2,000,000. The default is 1,000.</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Configuring Graceful Timeout&quot; on page 633.</td>
</tr>
<tr>
<td>Bandwidth Limit per Cluster Interface</td>
<td>Defines the maximum bandwidth (in Mbps) allowed on each Cluster interface.</td>
</tr>
<tr>
<td>configure network &gt; mtc</td>
<td></td>
</tr>
<tr>
<td>settings &gt; cluster-network-max-bandwidth</td>
<td></td>
</tr>
<tr>
<td>[MtcClusterNetworkMaxBandwidth]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The valid value is 1 to 10,000. The default is 1,000.</td>
</tr>
<tr>
<td></td>
<td>For more information, see Configuring Maximum Bandwidth for Cluster Interfaces on page 626.</td>
</tr>
<tr>
<td>Cluster Redundancy Mode</td>
<td>Defines the cluster redundancy mode.</td>
</tr>
<tr>
<td>configure network &gt; mtc</td>
<td></td>
</tr>
<tr>
<td>settings &gt; redundancy-mode</td>
<td></td>
</tr>
<tr>
<td>[MtcRedundancyMode]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• [0] HA Mode (Default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Best Effort</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Configuring Cluster Redundancy Mode&quot; on page 625.</td>
</tr>
<tr>
<td>Media Transcoders Table</td>
<td>Defines Media Transcoders.</td>
</tr>
<tr>
<td>configure network &gt; mtc</td>
<td></td>
</tr>
<tr>
<td>entity</td>
<td></td>
</tr>
<tr>
<td>[ MtcEntities ]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FORMAT MtcEntities_Index = MtcEntities_MtceName, MtcEntities_MtceOamIpAddr, MtcEntities_MtceAdminState;</td>
</tr>
<tr>
<td></td>
<td>['\MtcEntities ']</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Step 6: Adding Media Transcoder to Cluster Manager&quot; on page 631.</td>
</tr>
</tbody>
</table>

65.7 Quality of Experience Parameters

The Quality of Experience (QoE) parameters are described in the table below.

Table 65-30: Quality of Experience Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEM Parameters</td>
<td></td>
</tr>
<tr>
<td>Server IP</td>
<td>Defines the IP address of the primary Session Experience Manager (SEM) server to where the quality experience reports are sent.</td>
</tr>
<tr>
<td>configure voip &gt; qoe</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>settings &gt; server-ip [QOESErverIP]</td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>Redundant Server IP configure voip &gt; qoe settings &gt; redundant-server-ip [QOESecondaryServerIp]</td>
<td>Defines the IP address of the secondary SEM server to where the quality experience reports are sent. This is applicable when the SEM &gt; EMS server is in Geographical Redundancy HA mode. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>Interface Name configure voip &gt; qoe settings &gt; interface-name [QOEInterfaceName]</td>
<td>Defines the IP network interface on which the quality experience reports are sent. The default is the OAMP interface. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
</tbody>
</table>
| QoE Connection by TLS configure voip > qoe settings > tls-enable [QOEEnableTLS] | Enables a TLS connection with the SEM server.  
  - [0] Disable (default)  
  - [1] Enable  
**Note:** For the parameter to take effect, a device reset is required. |
| QoE TLS Context Name configure voip > qoe settings > tls-context-name [QoETLSThContextName] | Selects a TLS Context (configured in the TLS Contexts table) for the TLS connection with the SEM server. The valid value is a string representing the name of the TLS Context as configured in the 'Name' field of the TLS Contexts table. The default is the default TLS Context (ID 0). |
| QoE Report Mode report-mode [QoeReportMode] | Defines at what stage of the call the device sends the QoE data of the call to the SEM server.  
  - [0] Report QoE During Call (default)  
  - [1] Report QoE at End of Call  
**Note:** If a QoE traffic overflow between SEM and the device occurs, the device sends the QoE data only at the end of the call, regardless of the settings of the parameter. |

### Quality of Experience Profile Table

The table defines Quality of Experience Profiles. The format of the ini file table parameter is as follows:

```
[QOEProfile]  
FORMAT QOEProfile_Index = QOEProfile_Name, QOEProfile_SensitivityLevel;  
[/QOEProfile]  
```

For more information, see "Configuring Quality of Experience Profiles" on page 307.

### Quality of Experience Color Rules Table

The table defines Quality of Experience Color Rules. The format of the ini file table parameter is as follows:

```
[QOECOLORRules]  
FORMAT QOECOLORRules_Index = QOECOLORRules_QoeProfile, QOECOLORRules_ColorRuleIndex, QOECOLORRules_monitoredParam, QOECOLORRules_direction, QOECOLORRules_profile, QOECOLORRules_MinorThreshold, QOECOLORRules_MinorHysteresis, QOECOLORRules_MajorThreshold, QOECOLORRules_MajorHysteresis;  
[/QOECOLORRules]  
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="QOEColorRules" /></td>
<td>For more information, see &quot;Configuring Quality of Experience Profiles&quot; on page 307.</td>
</tr>
</tbody>
</table>

**Bandwidth Profile Table**

Bandwidth Profile configure voip > qoe bw-profile [BWProfile]  

The table defines Bandwidth Profiles.  
The format of the ini file table parameter is as follows:  

```
[BWProfile]
FORMAT BWProfile_Index = BWProfile_Name,  
BWProfile_EgressAudioBandwidth, BWProfile_IngressAudioBandwidth,  
BWProfile_EgressVideoBandwidth, BWProfile_IngressVideoBandwidth,  
BWProfile_TotalEgressBandwidth, BWProfile_TotalIngressBandwidth,  
BWProfile_WarningThreshold, BWProfile_hysteresis,  
BWProfile_GenerateAlarms;
```

For more information, see "Configuring Bandwidth Profiles" on page 312.  
**Note:** For the parameter to take effect, a device reset is required.

**Quality of Service Rules Table**

Quality of Service Rules configure voip > qoe quality-of-service-rules [QualityOfServiceRules]  

Defines Quality of Service rules.  
The format of the ini file table parameter is as follows:  

```
[QualityOfServiceRules]
FORMAT QualityOfServiceRules_Index =  
QualityOfServiceRules_IPGroupName,  
QualityOfServiceRules_RuleMetric, QualityOfServiceRules_Severity,  
QualityOfServiceRules_RuleAction,  
QualityOfServiceRules_CallsRejectDuration,  
QualityOfServiceRules_AltIPProfileName;
```

For more information, see "Configuring Quality of Service Rules" on page 316.

**Performance Profile Table**

Performance Profile configure system > performance-profile [PerformanceProfile]  

Defines alarm thresholds per metric (ASR, ACD and NER).  
The format of the ini file table parameter is as follows:  

```
[PerformanceProfile]
FORMAT PerformanceProfile_Index = PerformanceProfile_Entity,  
PerformanceProfile_IPGroupName, PerformanceProfile_SRDNName,  
PerformanceProfile_PMTyp, PerformanceProfile_MinorThreshold,  
PerformanceProfile_MajorThreshold, PerformanceProfile_Hysteresis,  
PerformanceProfile_MinimumSample, PerformanceProfile_WindowSize;
```

For more information, see "Configuring Performance Profiles" on page 743.
65.8 Control Network Parameters

65.8.1 IP Group, Proxy, Registration and Authentication Parameters

The proxy server, registration and authentication SIP parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP Groups</td>
<td>This table configures IP Groups. The format of the ini file table parameter is:</td>
</tr>
<tr>
<td></td>
<td>[ IPGroup ]</td>
</tr>
<tr>
<td></td>
<td>FORMAT IPGroup_Index = IPGroup_Type, IPGroup_Name, IPGroup_ProxySetName, IPGroup_SIPGroupName, IPGroup_Username, IPGroup_Password, IPGroup_UUIFormat, IPGroup_QOEProfile, IPGroup_BWProfile, IPGroup_AlwaysUseSourceAddr, IPGroup_MsgManUserDef1, IPGroup_MsgManUserDef2, IPGroup_SIPConnect, IPGroup_SBCPSAPMode, IPGroup_DTLSContext, IPGroup_CreatedByRoutingServer, IPGroup_UsedByRoutingServer, IPGroup_SBCOperationMode, IPGroup_SBCRouteUsingRequestURIPort, IPGroup_SBCKeepOriginalCallID, IPGroup_TopologyLocation, IPGroup_SBCDialPlanName, IPGroup_CallSetupRulesSetId, IPGroup_Tags, IPGroup_SBCUserStickiness;</td>
</tr>
</tbody>
</table>

For more information, see "Configuring IP Groups" on page 345.

Note: For the parameter to take effect, a device reset is required.

Accounts Table

<table>
<thead>
<tr>
<th>Accounts</th>
<th>Defines user accounts for registering and/or authenticating (digest) IP Groups (e.g., an IP-PBX) with a Serving IP Group (e.g., a registrar server). The format of the ini file table parameter is as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Account]</td>
</tr>
<tr>
<td></td>
<td>FORMAT Account_Index = Account_ServedTrunkGroup, Account_ServedIPGroupName, Account_Username, Account_Password, Account_HostName, Account_ContactUser, Account_Register, Account_RegistrarStickiness, Account_RegistrarSearchMode, Account_RegEventPackageSubscription, Account_ApplicationType;</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>[Account]</td>
<td>For more information, see &quot;Configuring Registration Accounts&quot; on page 375.</td>
</tr>
</tbody>
</table>

**Proxy Registration Parameters**

**Proxy Name**
`configure voip > sip-definition proxy-and-registration > 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:** The parameter functions together with the UseProxyIPasHost parameter.

**Use Proxy IP as Host**
`configure voip > sip-definition settings > use-proxy-ip-as-host [UseProxyIPasHost]`
Enables the use of the proxy server's IP address (in dotted-decimal notation) as the host name in SIP From and To headers in REGISTER requests.

- [0] Disable (default)
- [1] Enable

If the parameter is disabled and the device registers to an IP Group (i.e., proxy server), it uses the string configured by the ProxyName parameter as the host name in the REGISTER's Request-URI and uses the string configured by the IP Groups table parameter, SIPGroupName as the host name in the To and From headers. If the IP Group is configured with a Proxy Set that has multiple IP addresses, all the REGISTER messages sent to these proxies are sent with the same host name. **Note:** If the parameter is disabled and the ProxyName parameter is not configured, the proxy's IP address is used as the host name in the REGISTER Request-URI.

**Redundancy Mode**
`configure voip > sip-definition settings > 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).

**Proxy IP List Refresh Time**
`configure voip > sip-definition settings > proxy-ip-lst-rfrsh-time [ProxyIPListRefreshTime]`
Defines the periodic rate (in seconds) at which the device performs DNS queries to resolve FQDNs (host names) into IP addresses. For example, if a Proxy Set is configured with an FQDN and the parameter is configured to 60, the device queries the DNS server every 60 seconds, which refreshes the Proxy Set's list of DNS-resolved IP addresses. The valid value is 0, or 5 to 2,000,000. The default is 60. The value 0 disables periodic DNS queries and DNS resolution is done only once - upon device reset, device power up, or new and modified configuration.

**Always Use Proxy**
`configure voip > sip-definition`
Determines whether the device sends SIP messages and responses through a Proxy server.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| proxy-and-registration > always-use-proxy     | ▪ [0] Disable = (Default) Use standard SIP routing rules.  
▪ [1] Enable = All SIP messages and responses are sent to the Proxy server.  
**Note:** The parameter is applicable only if a Proxy server is used (i.e., the parameter IsProxyUsed is set to 1). |
| [AlwaysSendToProxy]                           |                                                                                                                                                                                                                                                                                                                                            |
| DNS Query Type                                 | Enables the use of DNS Naming Authority Pointer (NAPTR) and Service Record (SRV) queries to resolve Proxy and Registrar servers and to resolve all domain names that appear in the SIP Contact and Record-Route headers.  
▪ [0] A-Record = (Default) No NAPTR or SRV queries are performed.  
▪ [1] SRV = If the Proxy/Registrar IP address parameter, Contact/Record-Route headers, or IP address configured in the routing tables contain a domain name, an SRV query is performed. The device uses the first host name received from the SRV query. The device then performs a DNS A-record query for the host name to locate an IP address.  
▪ [2] NAPTR = An NAPTR query is performed. If it is successful, an SRV query is sent according to the information received in the NAPTR response. If the NAPTR query fails, an SRV query is performed according to the configured transport type.  
**Note:**  
▪ 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 Sets table. |
| Proxy DNS Query Type                          | 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 configured, a NAPTR query is not performed. |
<p>| [ProxyDNSQueryType]                           |                                                                                                                                                                                                                                                                                                                                            |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Gateway Name for OPTIONS</td>
<td>Determines whether the device uses its IP address or string name (&quot;gateway name&quot;) in keep-alive SIP OPTIONS messages (host part of the Request-URI). To configure the &quot;gateway name&quot;, use the SIPGatewayName parameter. The device uses the OPTIONS request as a keep-alive message with its primary and redundant SIP proxy servers (i.e., the EnableProxyKeepAlive parameter is set to 1).</td>
</tr>
<tr>
<td>Use Gateway Name For Options</td>
<td>[UseGatewayNameForOptions]</td>
</tr>
<tr>
<td>[0] No = (Default) Device's IP address is used in keep-alive OPTIONS messages.</td>
<td></td>
</tr>
<tr>
<td>[1] Yes = Device's &quot;gateway name&quot; is used in keep-alive OPTIONS messages.</td>
<td></td>
</tr>
<tr>
<td>[2] Server = Device's IP address is used in the From and To headers in keep-alive OPTIONS messages.</td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td>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'.</td>
</tr>
<tr>
<td>Cnonce</td>
<td>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'.</td>
</tr>
<tr>
<td>Challenge Caching Mode</td>
<td>Enables local caching of SIP message authorization challenges from Proxy servers. The device sends the first request to the Proxy without authorization. The Proxy sends a 401/407 response with a challenge for credentials. The device saves (caches) the response for further uses. The device sends a new request with the appropriate credentials. Subsequent requests to the Proxy are automatically sent with credentials (calculated from the saved challenge). If the Proxy doesn't accept the new request and sends another challenge, the old challenge is replaced with the new one. One of the benefits of the feature is that it may reduce the number of SIP messages transmitted through the network.</td>
</tr>
<tr>
<td>Cnonce</td>
<td>[Cnonce]</td>
</tr>
<tr>
<td>[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.</td>
<td></td>
</tr>
<tr>
<td>[1] INVITE Only = Challenges issued for INVITE requests are cached. This prevents a mixture of REGISTER and INVITE authorizations.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
- This functionality can be configured per Proxy Set in the Proxy Sets table (see "Configuring Proxy Sets" on page 359).
- When enabled, NAPTR/SRV queries are used to discover Proxy servers even if the parameter DNSQueryType is disabled.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
|           | - Challenge caching is used with all proxies and not only with the active one.  
|           | - The challenge can be cached per Account or per user whose credentials are known through the User Info table.  |

**Proxy Address Table**

**Proxy IP Table**  
`configure voip > proxy-ip [ProxyIP]`  
The table defines proxy addresses per Proxy Set.  
The format of the ini file table parameter is as follows:  
```
[ProxyIP]  
FORMAT ProxyIp_Index = ProxyIP_ProxySetId,  
ProxyIp_ProxyIpIndex, ProxyIp_IpAddress,  
ProxyIp_TransportType;  
```

**Proxy Sets Table**

**Proxy Sets**  
`configure voip > proxy-set [ProxySet]`  
Defines the Proxy Sets.  
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_SRDNName,  
ProxySet_ClassificationInput, ProxySet_TLSContextName,  
ProxySet_ProxyRedundancyMode,  
ProxySet_DNSResolveMethod, ProxySet_KeepAliveFailureResp,  
ProxySet_GWIPv4SIPInterfaceName,  
ProxySet_SBCIPv4SIPInterfaceName,  
ProxySet_GWIPv6SIPInterfaceName,  
ProxySet_SBCIPv6SIPInterfaceName,  
ProxySet_MinActiveServersLB,  
ProxySet_SuccessDetectionRetries,  
ProxySet_SuccessDetectionInterval,  
ProxySet_FailureDetectionRetransmissions;  
```

**Registrar Parameters**

**Registration Time**  
`configure voip > sip-definition proxy-and-registration > registration-time [RegistrationTime]`  
Defines the time interval (in seconds) for registering to a Proxy server. The value is used in the SIP Expires header. The parameter also defines the time interval between Keep-Alive messages when the parameter EnableProxyKeepAlive is set to 2 (REGISTER).  
Typically, the device registers every 3,600 sec (i.e., one hour). The device resumes registration according to the parameter RegistrationTimeDivider.  
The valid range is 10 to 2,000,000. The default is 180.

**Re-registration Timing [%]**  
`configure voip > sip-definition settings > re-registration-timing`  
Defines the re-registration timing (in percentage). The timing is a percentage of the re-register timing set by the Registrar server. The valid range is 50 to 100. The default is 50.  
For example: If the parameter is set to 70% and the Registration
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[RegistrationTimeDivider]</td>
<td>Expires time is 3600, the device re-sends its registration request after 3600 x 70% (i.e., 2520 sec). <strong>Note:</strong> The parameter may be overridden if the parameter RegistrationTimeThreshold is greater than 0.</td>
</tr>
<tr>
<td>Registration Retry Time</td>
<td>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. <strong>Note:</strong> Registration retry time can also be configured with the MaxRegistrationBackoffTime parameter.</td>
</tr>
</tbody>
</table>
| Max Registration Backoff Time                | Defines a dynamic time-to-wait interval before the device attempts to register the SIP entity again after a registration failure. The parameter is applicable only to registrations initiated by the device on behalf of SIP entities (for example, User Info, Accounts, Endpoints or the device itself) with a SIP proxy server (registrar). The valid value is 0 to 3000000 (i.e., 3 million seconds). The default is 0 (i.e., disabled). In contrast to the RegistrationRetryTime parameter, which defines a fixed time to wait between registration attempts due to registration failure, this parameter configures the device to increase the time-to-wait interval for each subsequent registration attempt (per RFC 5626, Section 4.5) for a specific registration flow. In other words, the interval changes between registration attempts. The parameter operates together with the RegistrationRetryTime parameter. When the MaxRegistrationBackoffTime parameter is configured, the wait-time before another registration attempt increases after each failed registration (until it reaches the maximum value specified by the parameter). The device uses the following algorithm to calculate the incremental augmented wait-time between each registration attempt:  
  \[
  \text{Wait Time} = \min (\text{max-time}, (\text{base-time} \times (2^{\text{consecutive-failures}})))
  \]
  Where:
  - max-time is the value configured by MaxRegistrationBackoffTime
  - base-time is the value configured by RegistrationRetryTime
  For example, if max-time is 1800 seconds and base-time is 30 seconds, and there were three consecutive registration failures, then the upper-bound wait time is the minimum of (1800, 30*(2^3)), which is (1800, 240) and thus, the minimum of the two values is 240 (seconds). The actual time the device waits before retrying registration is computed by a uniform random time between 50% and 100% of the upper-bound wait time (e.g., for an upper-bound wait-time of 240, the actual wait-time is between 120 and 240 seconds). As can be seen from the algorithm, the upper-bound wait time can never exceed the value of the MaxRegistrationBackoffTime parameter. |
<p>| Registration Time Threshold                   | Defines a threshold (in seconds) for re-registration timing. If the parameter is greater than 0, but lower than the computed re- |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>proxy-and-registration &gt; registration-time-thres [RegistrationTimeThreshold]</td>
<td>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.</td>
</tr>
<tr>
<td>ReRegister On Connection Failure configure voip &gt; sip-definition settings &gt; reg-on-conn-failure [ReRegisterOnConnectionFailure]</td>
<td>Enables the device to perform SIP re-registration upon TCP/TLS connection failure.</td>
</tr>
<tr>
<td></td>
<td>• [0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Enable</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition settings &gt; expl-un-reg [UnregistrationMode]</td>
<td>Enables the device to perform explicit unregisters.</td>
</tr>
<tr>
<td></td>
<td>• [0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Enable = The device sends an asterisk (&quot;<em>&quot;) 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 &quot;0&quot; 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 &quot;</em>&quot; Contact header field value allows a registering UA to remove all bindings associated with an address-of-record (AOR) without knowing their precise values. <strong>Note:</strong> The REGISTER-specific Contact header field value of &quot;*&quot; applies to all registrations, but it can only be used if the Expires header field is present with a value of &quot;0&quot;.</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition settings &gt; add-empty-author-hdr [EmptyAuthorizationHeader]</td>
<td>Enables the inclusion of the SIP Authorization header in initial registration (REGISTER) requests sent by the device.</td>
</tr>
<tr>
<td></td>
<td>• [0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] Enable</td>
</tr>
<tr>
<td></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>• username - set to the value of the private user identity</td>
</tr>
<tr>
<td></td>
<td>• realm - set to the domain name of the home network</td>
</tr>
<tr>
<td></td>
<td>• uri - set to the SIP URI of the domain name of the home network</td>
</tr>
<tr>
<td></td>
<td>• nonce - set to an empty value</td>
</tr>
<tr>
<td></td>
<td>• response - set to an empty value</td>
</tr>
<tr>
<td></td>
<td>For example:</td>
</tr>
<tr>
<td></td>
<td>Authorization: Digest username=<a href="mailto:alice_private@home1.net">alice_private@home1.net</a>, realm=&quot;home1.net&quot;, nonce=&quot;&quot;, response=&quot;e56131d19580cd833064787ecc&quot;</td>
</tr>
</tbody>
</table>
| | **Note:** This registration header is according to the IMS 3GPP
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add initial Route Header</td>
<td>Enables the inclusion of the SIP Route header in initial registration or re-registration (REGISTER) requests sent by the device.</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition proxy-and-registration &gt; add-init-rte-hdr</td>
<td></td>
</tr>
<tr>
<td>[InitialRouteHeader]</td>
<td>- [0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>- [1] Enable</td>
</tr>
<tr>
<td></td>
<td>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: <a href="">sip:10.10.10.10;lr;transport=udp</a> or Route: &lt;sip: pcscfgm.ims.rr.com;lr;transport=udp&gt;</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition settings &gt; ping-pong-keep-alive</td>
<td>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.</td>
</tr>
<tr>
<td>[UsePingPongKeepAlive]</td>
<td>- [0] Disable (default)</td>
</tr>
<tr>
<td></td>
<td>- [1] Enable</td>
</tr>
<tr>
<td></td>
<td>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 &quot;ping&quot; keep-alive and a corresponding server-to-client &quot;pong&quot; 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 &quot;pong&quot; within an appropriate amount of time, it considers the flow failed.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition settings &gt; ping-pong-keep-alive-time</td>
<td>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.</td>
</tr>
<tr>
<td>[PingPongKeepAliveTime]</td>
<td>The default range is 5 to 2,000,000. The default is 120.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Max Generated Register Rate</td>
<td>Defines the maximum number of user register requests</td>
</tr>
</tbody>
</table>
### 65. Configuration Parameters Reference

#### 65.8.2 Network Application Parameters

The SIP network application parameters are described in the table below.

**Table 65-32: SIP Network Application Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SRDs Table</strong></td>
<td></td>
</tr>
<tr>
<td>SRDs configure voip &gt; srd [SRD]</td>
<td>Defines Signaling Routing Domains (SRD). The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td></td>
<td>[ SRD ]</td>
</tr>
<tr>
<td></td>
<td>FORMAT SRD_Index = SRD_Name, SRD_IntraSRDMediaAnchoring, SRD_BlockUnRegUsers, SRD_MaxNumOfRegUsers, SRD_EnableUnAuthenticatedRegistrations, SRD_SharingPolicy, SRD_UsedByRoutingServer, SRD_SBCOperationMode, SRD_SBCRoutingPolicyName;</td>
</tr>
<tr>
<td></td>
<td>[ `SRD ]</td>
</tr>
<tr>
<td></td>
<td>For more information, see &quot;Configuring SRDs&quot; on page 327.</td>
</tr>
<tr>
<td><strong>SIP Interfaces Table</strong></td>
<td></td>
</tr>
<tr>
<td>SIP Interfaces configure voip &gt; sip-interface [SIPInterface]</td>
<td>Defines SIP Interfaces. The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td></td>
<td>[ SIPInterface ]</td>
</tr>
<tr>
<td></td>
<td>FORMAT SIPInterface_Index = SIPInterface_InterfaceName, SIPInterface_NetworkInterface, SIPInterface_ApplicationType,</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>SIPInterface_UDPPort, SIPInterface_TCPPort, SIPInterface_TLSPort, SIPInterface_AdditionalUDPPorts, SIPInterface_SRDNName, SIPInterface_MessagePolicyName, SIPInterface_TLSContext, SIPInterface_TLSMutualAuthentication, SIPInterface_TCPKeepaliveEnable, SIPInterface_ClassificationFailureResponseType, SIPInterface_PreClassificationManSet, SIPInterface_EncapsulatingProtocol, SIPInterface_MediaRealm, SIPInterface_SBCDirectMedia, SIPInterface_BlockUnRegUsers, SIPInterface_MaxNumOfRegUsers, SIPInterface_EnableUnAuthenticatedRegistrations, SIPInterface_UsedByRoutingServer, SIPInterface_TopologyLocation, SIPInterface_PreParsingManSetName; [ SIPInterface ]</td>
<td>For more information, see &quot;Configuring SIP Interfaces&quot; on page 337.</td>
</tr>
</tbody>
</table>

configure voip > sip-definition settings > tcp-keepalive-time [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.  
**Note:**  
- Simple ACKs such as keepalives are not considered data packets.  
- TCP keepalive is enabled per SIP Interface in the SIP Interfaces table.  

configure voip > sip-definition settings > tcp-keepalive-interval [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 Interfaces table.  

configure voip > sip-definition settings > tcp-keepalive-retry [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 Interfaces table.  

**NAT Translation Table**  
NAT Translation Table  
configure network > nat-translation [NATTranslation]  
Defines NAT rules for translating source IP addresses per VoIP interface (SIP control and RTP media traffic) into NAT IP addresses.  
The format of the ini file table parameter is as follows:  

```
[ NATTranslation ]
FORMAT NATTranslation_Index =  
NATTranslation_SrcIPInterfaceName,  
NATTranslation_TargetIPAddress,  
NATTranslation_SourceStartPort,  
NATTranslation_SourceEndPort,  
NATTranslation_TargetStartPort,  
NATTranslation_TargetEndPort;
[/NATTranslation ]
```
For more information, see "Configuring NAT Translation per IP Interface" on page 149.  

**Media Realms table**  
Media Realms  
configure voip > realm  
Defines Media Realms.  
The format of the ini file table parameter is as follows:
65.9 General SIP Parameters

The general SIP parameters are described in the table below.

Table 65-33: General SIP Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[MaxSDPSessionVersionId]</td>
<td>Defines the maximum number of characters allowed in the SDP body's &quot;o=&quot; (originator and session identifier) field for the session ID and session version values. An example of an &quot;o=&quot; line with session ID and session version is shown.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>version values (in bold) is shown below:</td>
<td></td>
</tr>
<tr>
<td>o=jdoe 2890844526 2890842807 IN IP4 10.47.16.5</td>
<td>The valid value range is 1,000 to 214,748,3647 (default).</td>
</tr>
<tr>
<td>[UseRandomUser]</td>
<td>Enables the device to assign a random string value to the user part of the SIP Contact header for new user Account registrations with the device. The assigned user part is unique to each Account. An example of a randomly assigned user part is shown (in bold) below: Contact: <a href="">sip:HRaNEmZnfX6xZl4@pc33.atlanta.com</a> When enabled, each Account registers with its unique user part, and all INVITE messages for this new Account are sent with this unique user part. These assigned user parts are also used for registration refreshes and for un-registering these Accounts. The assigned user parts are used until the parameter is disabled. When enabled again, new random user parts are assigned to the Accounts.   • [0] = (Default) Disable • [1] = Enable To configure Accounts, see Configuring Registration Accounts on page 375.</td>
</tr>
<tr>
<td>[UnregisterOnStartup]</td>
<td>Enables the device to unregister all user Accounts that were registered with the device, upon a device reset. During device start-up, each Account sends a REGISTER message (containing &quot;Contact: &quot;*) to unregister all contact URIs belonging to its Address-of-Record (AOR). A second after they are unregistered, the device re-registers the Account. • [0] = (Default) Disable • [1] = Enable To configure Accounts, see Configuring Registration Accounts on page 375.</td>
</tr>
<tr>
<td>Send reject on overload</td>
<td>Disables the sending of SIP 503 (Service Unavailable) responses upon receipt of new SIP dialog-initiating requests when the device's CPU is overloaded and thus, unable to accept and process new SIP messages. • [0] Disable = No SIP 503 response is sent when CPU overloaded. • [1] = Enable (default) = SIP 503 response is sent when CPU overloaded. Note: Even if the parameter is disabled (i.e., 503 is not sent), the device still discards the new SIP dialog-initiating requests when the CPU is overloaded.</td>
</tr>
<tr>
<td>EnableNonInvite408Reply</td>
<td>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.</td>
</tr>
<tr>
<td>Remote Management by SIP Notify</td>
<td>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)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| remote-reset [EnableSIPRemoteReset] | - [1] Enable. The action depends on the Event header value:  
  - 'check-sync;reboot=false': triggers the regular Automatic Update feature (if Automatic Update has been enabled on the device)  
  - 'check-sync;reboot=true': triggers a device reset  
  **Note:** The Event header value is proprietary to AudioCodes. |
| 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 100. The default is 100. |
| [SIPForceRport] | Determines whether the device sends SIP responses to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the SIP Via header.  
  - [0] = (Default) Disabled. The device sends the SIP response to the UDP port defined in the Via header. If the Via header contains the 'rport' parameter, the response is sent to the UDP port from where the SIP request is received.  
  - [1] = Enabled. SIP responses are sent to the UDP port from where SIP requests are received even if the 'rport' parameter is not present in the Via header. |
| Reject Cancel after Connect configure voip > sip-definition settings > reject-cancel-after-connect [RejectCancelAfterConnect] | Enables or disables the device to accept or reject SIP CANCEL requests received after the receipt of a 200 OK in response to an INVITE (i.e., call established). According to the SIP standard, a CANCEL can be sent only during the INVITE transaction (before 200 OK), and once a 200 OK response is received the call can be rejected only by a BYE request.  
  - [0] Disable = (Default) Accepts a CANCEL request received during the INVITE transaction by sending a 200 OK response and terminates the call session.  
  - [1] Enable = Rejects a CANCEL request received during the INVITE transaction by sending a SIP 481 (Call/Transaction Does Not Exist) response and maintains the call session. |
| Verify Received RequestURI configure voip > sip-definition settings > verify-rcvd-requri [VerifyReceivedRequestUri] | Enables the device to reject SIP requests (such as ACK, BYE, or re-INVITE) whose user part in the Request-URI is different from the user part received in the Contact header of the last sent SIP request.  
  - [0] Disable = (Default) Even if the user is different, the device accepts the SIP request.  
  - [1] Enable = If the user is different, the device rejects the SIP request (BYE is responded with 481; re-INVITE is responded with 404; ACK is ignored). |
| Max Number of Active Calls configure voip > sip-definition settings > 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). |
| QoS statistics in SIP Release Call configure voip > sip-definition | 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) |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
The X-RTP-Stat header provides the following statistics:  
- Number of received and sent voice packets  
- Number of received and sent voice octets  
- Received packet loss, jitter (in ms), and latency (in ms)  
The X-RTP-Stat header contains the following fields:  
- PS=\langle\text{voice packets sent}\rangle  
- OS=\langle\text{voice octets sent}\rangle  
- PR=\langle\text{voice packets received}\rangle  
- OR=\langle\text{voice octets received}\rangle  
- PL=\langle\text{receive packet loss}\rangle  
- JI=\langle\text{jitter in ms}\rangle  
- LA=\langle\text{latency in ms}\rangle  

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.7.20A.000.038
Reason: Q.850 ;cause=16 ;text="local"
Content-Length: 0
```

Enable Early Media  
[EnableEarlyMedia]  
Global parameter enabling the Early Media feature for sending media (e.g., ringing) before the call is established.  
You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEarlyMedia). For a detailed description of the parameter and for configuring the functionality, see "Configuring IP Profiles" on page 417.  
**Note:** If the functionality is configured for a specific profile, the settings of the global parameter is ignored for calls associated with the profile.

Session Expires Disconnect Time  
[SessionExpiresDisconnectTime]  
Defines a session expiry timeout.  
The new session expiry timeout is calculated by subtracting the configured value from the original timeout as specified in the Session-Expires header. However, the new timeout must be greater than or equal to one-third (1/3) of the Session-Expires value. If the refresher does not send a refresh request within the new timeout, the device disconnects the session (i.e., sends a SIP BYE).  
For example, if you configure the parameter to 32 seconds and the Session-Expires value is 180 seconds, the session timeout occurs 148
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>seconds (i.e., 180 minus 32) after the last session refresh. If the Session-Expires header value is 90 seconds, the timeout occurs 60 seconds after the last refresh. This is because 90 minus 32 is 58 seconds, which is less than one third of the Session-Expires value (i.e., 60/3 is 30, and 90 minus 30 is 60). The valid range is 0 to 32 (in seconds). The default is 32.</td>
<td></td>
</tr>
<tr>
<td>[RemoveToTagInFailure Response]</td>
<td>Determines whether the device removes the ‘to’ header tag from final SIP failure responses to INVITE transactions.</td>
</tr>
<tr>
<td></td>
<td>• [0] = (Default) Do not remove tag.</td>
</tr>
<tr>
<td>[EnableRTCPAttribute]</td>
<td>Enables the use of the 'rtcp' attribute in the outgoing SDP.</td>
</tr>
<tr>
<td></td>
<td>• [0] = Disable (default)</td>
</tr>
<tr>
<td></td>
<td>• [1] = Enable</td>
</tr>
<tr>
<td>[OPTIONSUserPart]</td>
<td>Defines the user part value of the Request-URI for outgoing SIP OPTIONS requests. If no value is configured, the configuration parameter 'Username' value is used. A special value is ‘empty’, indicating that no user part in the Request-URI (host part only) is used. The valid range is a 30-character string. By default, this value is not defined.</td>
</tr>
<tr>
<td>Fax Signaling Method</td>
<td>Global parameter defining the SIP signaling method for establishing and transmitting a fax session when the device detects a fax. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IsFaxUsed). For a detailed description of the parameter, see &quot;Configuring IP Profiles&quot; on page 417.</td>
</tr>
<tr>
<td>fax-sig-method</td>
<td></td>
</tr>
<tr>
<td>[IsFaxUsed]</td>
<td>Global parameter defining the SIP signaling method for establishing and transmitting a fax session when the device detects a fax. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_IsFaxUsed). For a detailed description of the parameter, see &quot;Configuring IP Profiles&quot; on page 417.</td>
</tr>
<tr>
<td>[FaxVBDBehavior]</td>
<td>Determines the device’s fax transport behavior when G.711 VBD coder is negotiated at call start.</td>
</tr>
<tr>
<td></td>
<td>• [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).</td>
</tr>
<tr>
<td></td>
<td>• [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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| [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: 
  \[a=rtpmap:120 \text{NoAudio}/8000\]  
  \[\text{Note: For incoming SDP offers, NoAudio is always supported.} \]  
| SIP Transport Type | configure voip > sip-definition settings > 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)  
  [Note:  
  - It's recommended to use TLS for communication with a SIP Proxy and not for direct device-to-device communication.  
  - For incoming calls (i.e., incoming), the device accepts all these protocols.} |
| Display Default SIP Port | configure voip > sip-definition settings > 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:  
  \[\text{From: } <\text{sip:}:+4000@10.8.4.105:5060;user=phone>;tag=f25419a96;epid=009FAB8F3E}\]  
  - [0] Disable (default)  
  - [1] Enable |
| Enable SIPS | configure voip > sip-definition settings > enable-sips [EnableSIPS] | Enables secured SIP (SIPS URI) connections over multiple hops.  
  - [0] Disable (default)  
  - [1] Enable  
  When the SIPTransportType parameter is set to 2 (i.e., TLS) and the parameter EnableSIPS is disabled, TLS is used for the next network hop only. When the parameter SIPTransportType is set to 2 or 1 (i.e., TCP or TLS) and EnableSIPS is enabled, TLS is used through the entire connection (over multiple hops).  
  [Note: If the parameter is enabled and the parameter SIPTransportType is set to 0 (i.e., UDP), the connection fails.] |
| TCP/TLS Connection Reuse | tcp-conn-reuse [EnableTCPConnectionReuse] | Enables the reuse of an established TCP or TLS connection between the device and a SIP user agent (UA) for subsequent SIP requests sent to the UA. Any new out-of-dialog requests (e.g., INVITE or REGISTER) use the same secured connection. One of the benefits of enabling the parameter is that it may improve performance by eliminating the need for additional TCP/TLS handshakes with the UA, allowing sessions to be established rapidly.  
  - [0] Disable = The device uses a new TCP or TLS connection with the UA.  
  - [1] Enable = (Default) The device uses the same TCP or TLS  

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>connection for all SIP requests with the UA.</td>
<td>Note: For SIP responses, the device always uses the same TCP/TLS connection, regardless of the parameter settings.</td>
</tr>
<tr>
<td>Fake TCP alias</td>
<td>Enables the re-use of the same TCP/TLS connection for sessions with the same user, even if the &quot;alias&quot; parameter is not present in the SIP Via header of the first INVITE.</td>
</tr>
<tr>
<td>[FakeTCPalias]</td>
<td>- [0] Disable = (Default) TCP/TLS connection reuse is done only if the &quot;alias&quot; parameter is present in the Via header of the first INVITE.</td>
</tr>
<tr>
<td></td>
<td>- [1] Enable</td>
</tr>
<tr>
<td>Note:</td>
<td>To enable TCP/TLS connection re-use, set the EnableTCPConnectionReuse parameter to 1.</td>
</tr>
<tr>
<td>Reliable Connection Persistent Mode</td>
<td>Enables setting of all TCP/TLS connections as persistent and therefore, not released.</td>
</tr>
<tr>
<td>[ReliableConnectionPersistentMode]</td>
<td>- [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.</td>
</tr>
<tr>
<td></td>
<td>- [1] = Enable - TCP connections to all destinations are persistent and not released unless the device reaches 70% of its maximum TCP resources.</td>
</tr>
<tr>
<td>Note:</td>
<td>While trying to send a SIP message connection, reuse policy determines whether live connections to the specific destination are re-used.</td>
</tr>
<tr>
<td>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.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Note: If the destination is a Proxy server, the TCP/TLS connection is persistent regardless of the settings of the parameter.</td>
</tr>
<tr>
<td>TCP Timeout</td>
<td>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.</td>
</tr>
<tr>
<td>[SIPTCPTimeout]</td>
<td>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.</td>
</tr>
<tr>
<td>SIP Destination Port</td>
<td>Defines the SIP destination port for sending initial SIP requests.</td>
</tr>
<tr>
<td>[SIPDestinationPort]</td>
<td>The valid range is 1 to 65534. The default port is 5060.</td>
</tr>
<tr>
<td>Note:</td>
<td>SIP responses are sent to the port specified in the Via header.</td>
</tr>
<tr>
<td>Use user=phone in SIP URL</td>
<td>Determines whether the 'user=phone' string is added to the SIP URI and SIP To header.</td>
</tr>
<tr>
<td>[IsUserPhone]</td>
<td>- [0] No = 'user=phone' string is not added.</td>
</tr>
<tr>
<td></td>
<td>- [1] Yes = (Default) 'user=phone' string is part of the SIP URI and SIP To header.</td>
</tr>
<tr>
<td>Use user=phone in From Header</td>
<td>Determines whether the 'user=phone' string is added to the From and Contact SIP headers.</td>
</tr>
<tr>
<td>[IsUserPhone]</td>
<td>- [0] No = (Default) Doesn't add 'user=phone' string.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>definition settings &gt; phone-in-from-hdr [IsUserPhoneInFrom]</td>
<td>• [1] Yes = 'user=phone' string is part of the From and Contact headers.</td>
</tr>
</tbody>
</table>

Use Tel URI for Asserted Identity configure voip > sip-definition settings > uri-for-assert-id [UseTelURIForAssertedIdentity] Determines the format of the URI in the P-Asserted-Identity and P-Preferred-Identity headers.  
• [0] Disable = (Default) 'sip:'  
• [1] Enable = 'tel:'

Enable GRUU configure voip > sbc settings > enable-gruu [EnableGRUU] Determines whether the Globally Routable User Agent URIs (GRUU) mechanism is used, according to RFC 5627. This is used for obtaining a GRUU from a registrar and for communicating a GRUU to a peer within a dialog.  
• [0] Disable (default)  
• [1] Enable  

A GRUU is a SIP URI that routes to an instance-specific UA and can be reachable from anywhere. There are a number of contexts in which it is desirable to have an identifier that addresses a single UA (using GRUU) rather than the group of UA's indicated by an Address of Record (AOR). For example, in call transfer where user A is talking to user B, and user A wants to transfer the call to user C. User A sends a REFER to user C:  

```
REFER sip:C@domain.com SIP/2.0
From: sip:A@domain.com;tag=99asd
To: sip:C@domain.com
Refer-To: (URI that identifies B's UA)
```

The Refer-To header needs to contain a URI that user C can use to place a call to user B. This call needs to route to the specific UA instance that user B is using to talk to user A. User B should provide user A with a URI that has to be usable by anyone. It needs to be a GRUU.  

• Obtaining a GRUU: The mechanism for obtaining a GRUU is through registrations. A UA can obtain a GRUU by generating a REGISTER request containing a Supported header field with the value “gruu”. The UA includes a “+sip.instance” Contact header parameter of each contact for which the GRUU is desired. This Contact parameter contains a globally unique ID that identifies the UA instance. The global unique ID is created from one of the following:  
  ✓ If the REGISTER is per the device’s client (endpoint), it is the MAC address concatenated with the phone number of the client.  
  ✓ If the REGISTER is per device, it is the MAC address only.  
  ✓ When using TP, “User Info” can be used for registering per endpoint. Thus, each endpoint can get a unique id – its phone number. The globally unique ID in TP is the MAC address concatenated with the phone number of the endpoint.  

If the remote server doesn’t support GRUU, it ignores the parameters of the GRUU. Otherwise, if the remote side also supports GRUU, the REGISTER responses contain the “gruu” parameter in each Contact header. The parameter contains a SIP or SIPS URI that represents a GRUU corresponding to the UA instance that registered the contact. The server provides the same GRUU for the same AOR and instance-id when sending REGISTER again after registration expiration. RFC 5627 specifies that the remote target is a GRUU target if its’ Contact URL has
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[IsCiscoSCEMode]</td>
<td>Determines whether a Cisco gateway exists at the remote side.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>[0]</strong> = (Default) No Cisco gateway exists at the remote side.</td>
</tr>
<tr>
<td></td>
<td>▪ <strong>[1]</strong> = A Cisco gateway exists at the remote side.</td>
</tr>
<tr>
<td></td>
<td>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 Silence Suppression for the used coder is configured to 2 (enable without adaptation). In this case, Silence Suppression is used on the channel but not declared in the SDP.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The IsCiscoSCEMode parameter is applicable only when the selected coder is G.729.</td>
</tr>
<tr>
<td>[UserAgentDisplayInfo]</td>
<td>Defines the string that is used in the SIP User-Agent and Server response headers. When configured, the string <code>&lt;UserAgentDisplayInfo value&gt;/software version</code> is used, for example: <code>User-Agent: myproduct/v.7.20A.000.038</code></td>
</tr>
<tr>
<td></td>
<td>If not configured, the default string, <code>&lt;AudioCodes productName&gt;/software version</code> is used, for example: <code>User-Agent: Audiocodes-Sip-Gateway-Mediant Software SBC/v.7.20A.000.038</code></td>
</tr>
<tr>
<td></td>
<td>The maximum string length is 50 characters.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> 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).</td>
</tr>
<tr>
<td>[SIPSDPSessionOwner]</td>
<td>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 &quot;AudiocodesGW&quot;.</td>
</tr>
<tr>
<td></td>
<td>For example: <code>o=AudiocodesGW 1145023829 1145023705 IN IP4 10.33.4.126</code></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The parameter is applicable only when the device creates a new SIP message (and SDP) such as when the device plays a ringback tone. The parameter is not applicable to SIP messages that the device receives from one end and sends to another (i.e., does not modify the SDP's 'o' field).</td>
</tr>
</tbody>
</table>
| [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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subject</td>
<td>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.</td>
</tr>
<tr>
<td>Multiple Packetization Time Format</td>
<td>Determines whether the 'mptime' attribute is included in the outgoing SDP.</td>
</tr>
<tr>
<td>Enable Ptime</td>
<td>Determines whether the 'ptime' attribute is included in the SDP.</td>
</tr>
<tr>
<td>3xx Behavior</td>
<td>Determines the device’s behavior regarding call identifiers when a 3xx response is received for an outgoing INVITE request. The device can use the same call identifiers (Call-ID, To, and From tags) or change them in the new initiated INVITE.</td>
</tr>
<tr>
<td>Retry-After Time</td>
<td>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.</td>
</tr>
<tr>
<td>Fake Retry After</td>
<td>Determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by the parameter.</td>
</tr>
</tbody>
</table>

### Subject

Configure `voip > sip-definition settings > usrd-def-subject` [SIPSubject]

The `SIPSubject` parameter defines the Subject header value in outgoing INVITE messages. If not specified, the Subject header isn't included (default). The maximum length is up to 50 characters.

### Multiple Packetization Time Format

Configure `voip > sip-definition settings > mutptime-format` [MultiPtimeFormat]

The `MultiPtimeFormat` parameter determines whether the 'mptime' attribute is included in the outgoing SDP. The 'mptime' attribute enables the device to define a separate packetization period for each negotiated coder in the SDP. The 'mptime' attribute is only included if the parameter is enabled even if the remote side includes it in the SDP offer. Upon receipt, each coder receives its 'ptime' value in the following precedence: from 'mptime' attribute, from 'ptime' attribute, and then from default value.

### Enable Ptime

Configure `voip > sip-definition settings > enable-ptime` [EnablePtime]

The `EnablePtime` parameter determines whether the 'ptime' attribute is included in the SDP.

### 3xx Behavior

Configure `voip > sip-definition settings > 3xx-behavior` [3xxBehavior]

The `3xxBehavior` parameter determines the device’s behavior regarding call identifiers when a 3xx response is received for an outgoing INVITE request. The device can use the same call identifiers (Call-ID, To, and From tags) or change them in the new initiated INVITE.

### Retry-After Time

Configure `voip > sip-definition settings > retryaftr-time` [RetryAfterTime]

The `RetryAfterTime` parameter 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.

### Fake Retry After

Configure `voip > sip-definition settings > fake-retry-after` [FakeRetryAfter]

The `FakeRetryAfter` parameter determines whether the device, upon receipt of a SIP 503 response without a Retry-After header, behaves as if the 503 response included a Retry-After header and with the period (in seconds) specified by the parameter.

- **0**: Disable (default)
- Any positive value (in seconds) for defining the period

When enabled, this feature allows the device to operate with Proxy servers that do not include the Retry-After SIP header in SIP 503 (Service Unavailable) responses to indicate an unavailable service. The Retry-After header is used with the 503 (Service Unavailable) response to indicate how long the service is expected to be unavailable.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 &quot;Retry-After&quot; period.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable P-Associated-URI Header</th>
<th>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.</th>
</tr>
</thead>
<tbody>
<tr>
<td>p-associated-uri-hdr</td>
<td>• [0] Disable (default)</td>
</tr>
<tr>
<td>[EnablePAssociatedURIHeader]</td>
<td>• [1] Enable</td>
</tr>
<tr>
<td>Note: P-Associated-URIs in registration responses is handled only if the device is registered per endpoint (using the User Information file).</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source Number Preference</th>
<th>Determines from which SIP header the source (calling) number is obtained in incoming INVITE messages.</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure voip &gt; sip-definition settings &gt; src-nb-preference</td>
<td>• If not configured or if any string other than &quot;From&quot; or &quot;Pai2&quot; is configured, the calling number is obtained from a specific header using the following logic:</td>
</tr>
<tr>
<td></td>
<td>b. If the above header is not present, then the first P-Asserted-Identity header is used.</td>
</tr>
<tr>
<td></td>
<td>c. If the above header is not present, then the Remote-Party-ID header is used.</td>
</tr>
<tr>
<td></td>
<td>d. If the above header is not present, then the From header is used.</td>
</tr>
<tr>
<td></td>
<td>• &quot;From&quot; = The calling number is obtained from the From header.</td>
</tr>
<tr>
<td></td>
<td>• &quot;Pai2&quot; = The calling number is obtained using the following logic:</td>
</tr>
<tr>
<td></td>
<td>a. If a P-Preferred-Identity header is present, the number is obtained from it.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>c. If only one P-Asserted-Identity header is present, the calling number is obtained from it.</td>
</tr>
<tr>
<td>Note:</td>
<td>• The &quot;From&quot; and &quot;Pai2&quot; values are not case-sensitive.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enable Reason Header</th>
<th>Enables the usage of the SIP Reason header.</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure voip &gt; sip-definition settings &gt; reason-header</td>
<td>• [0] Disable</td>
</tr>
<tr>
<td>[EnableReasonHeader]</td>
<td>• [1] Enable (default)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gateway Name</th>
<th>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).</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure voip &gt; sip-definition settings &gt; gw-name</td>
<td>• [SIPGatewayName]</td>
</tr>
<tr>
<td>[SIPGatewayName]</td>
<td>Note:</td>
</tr>
<tr>
<td></td>
<td>• Ensure that the parameter value is the one with which the Proxy has</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| configure voip > sip-definition settings > zero-sdp-behavior [ZeroSDPHandling] | Determines the device’s response to an incoming SDP that includes an IP address of 0.0.0.0 in the SDP's Connection Information field (i.e., "c=IN IP4 0.0.0.0").
- **[0]** = (Default) Sets the IP address of the outgoing SDP’s c= field to 0.0.0.0.
- **[1]** = Sets the IP address of the outgoing SDP c= field to the IP address of the device. If the incoming SDP doesn’t contain the "a=inactive" line, the returned SDP contains the "a=recvonly" line. |
| Enable Delayed Offer configure voip > sip-definition settings > 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. |
| configure voip > sip-definition settings > crypto-life-time-in-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_CMAC_128_HMAC_SHA1_80 inline:hhQe10yZRcRcpkH5xYY9R1de370g91G1mpvNp[2^31]", it removes the lifetime parameter "2^31".
- **[0]** = Disable (default) |
- **[1]** = Enable |
| Enable Contact Restriction contact-restriction [EnableContactRestriction] | Determines whether the device sets the Contact header of outgoing INVITE requests to ‘anonymous’ for restricted calls.
- **[0]** = Disable (default) |
- **[1]** = Enable |
| configure voip > sip-definition settings > use-aor-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. |
| Enable User-Information Usage configure voip > sip-definition settings > user-inf-usage [EnableUserInfoUsage] | Enables the usage of the User Information, which is loaded to the device in the User Information Auxiliary file. For more information on User Information, see "User Information File" on page 663.
- **[0]** = Disable (default) |
- **[1]** = Enable |
| Note: For the parameter to take effect, a device reset is required. |
| configure voip > sip-definition settings > handle- | 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. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reason-header</td>
<td><strong>[HandleReasonHeader]</strong>&lt;br&gt;• [1] = (Default) Use the Reason header value for Release Reason mapping.</td>
</tr>
<tr>
<td>[EnableSilenceSuppInSDP]</td>
<td>Determines the device’s behavior upon receipt of SIP Re-INVITE messages that include the SDP's 'silencesupp:off' attribute.&lt;br&gt;• [0] = (Default) Disregard the 'silencesupp' attribute.&lt;br&gt;• [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.&lt;br&gt;Note: The parameter is applicable only if the G.711 coder is used.</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition &gt; number</td>
<td>Defines the maximum number of concurrent, outgoing SIP REGISTER dialogs. The parameter is used to control the registration rate.</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition settings &gt; rport-support</td>
<td>Enables the usage of the 'rport' parameter in the Via header.&lt;br&gt;• [0] = Disabled (default)&lt;br&gt;• [1] = Enabled&lt;br&gt;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.&lt;br&gt;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.&lt;br&gt;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.&lt;br&gt;If the Via header includes 'rport' with a port value (e.g., rport=1001), the destination port of the response is the port indicated in the 'rport' parameter.</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition settings &gt; x-channel-header</td>
<td>Enables the device to add the SIP X-Channel header to outgoing SIP messages. The header provides information on the physical channel on which the call is received or sent.&lt;br&gt;• [0] Disable = (Default) X-Channel header is not used.&lt;br&gt;• [1] Enable = X-Channel header is generated by the device and sent in SIP INVITE requests and 180, 183, and 200 OK responses. The header includes the channel and the device's IP address, using the following syntax:&lt;br&gt;x-channel: ds/ds1-/&lt;channel&gt;;IP=&lt;device's IP address&gt;&lt;br&gt;For example, the below shows a call on channel 4 of the device with IP address 192.168.13.1:&lt;br&gt;x-channel: ds/ds1-4;IP=192.168.13.1</td>
</tr>
<tr>
<td>[EnableRekeyAfter181]</td>
<td>Enables the device to send a re-INVITE with a new (different) SRTP key (in the SDP) if a SIP 181 response is received (&quot;call is being forwarded&quot;). The re-INVITE is sent immediately upon receipt of the 200 OK (when the call is answered).&lt;br&gt;• [0] = Disable (default)&lt;br&gt;• [1] = Enable&lt;br&gt;Note: The parameter is applicable only if SRTP is used.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>of-active-dialogs</td>
<td><strong>NumberOfActiveDialogs</strong></td>
</tr>
</tbody>
</table>
|                                               | The valid range is 1 to 20. The default is 20. **Note:**  
|                                               |  
|                                               | • Once a 200 OK is received in response to a REGISTER message, the REGISTER message is not considered in this maximum count limit.  
|                                               | • The parameter applies only to outgoing REGISTER messages (i.e., incoming is unlimited).                                                                                                                   |
|                                               |                                                                                                                                                                                                             |
| Network Node ID                               | **NetworkNodeId**                                                                                                                                                                                              |
|                                               | Defines the Network Node Identifier of the device for Avaya UCID. The valid value range is 1 to 0x7FFF. The default is 0. **Note:**  
|                                               |  
|                                               | • To use this feature, you must set the parameter to any value other than 0.  
|                                               | • To enable the generation by the device of the Avaya UCID value and adding it to the outgoing INVITE sent to the IP Group (Avaya entity), use the IP Groups table’s parameter 'UUI Format'. |
|                                               |                                                                                                                                                                                                             |
| Enable Microsoft Extension                    | **EnableMicrosoftExt**                                                                                                                                                                                          |
|                                               | Enables the modification of the called and calling number for numbers received with Microsoft’s proprietary "ext=xxx" parameter in the SIP INVITE URI user part. Microsoft Office Communications Server sometimes uses this proprietary parameter to indicate the extension number of the called or calling party.  
|                                               | • [0] Disable (default)  
|                                               | • [1] Enable  
|                                               | For example, if a calling party makes a call to telephone number 622125519100 Ext. 104, the device receives the SIP INVITE (from Microsoft’s application) with the URI user part as INVITE sip:622125519100;ext=104@10.1.1.10 (or INVITE tel:622125519100;ext=104). If the parameter EnableMicrosoftExt is enabled, the device modifies the called number by adding an "e" as the prefix, removing the "ext=" parameter, and adding the extension number as the suffix (e.g., e622125519100104). Once modified, the device can then manipulate the number further, using the Number Manipulation tables to leave only the last 3 digits (for example) for sending to a PBX. |
|                                               |                                                                                                                                                                                                             |
| configure voip > sip-definition settings > sip- | **UseSIPURIForDiversionHeader**                                                                                                                                                                               |
| uri-for-diversion-header                      | Defines the URI format in the SIP Diversion header.  
|                                               | • [0] = 'tel:' (default)  
|                                               | • [1] = 'sip:'  
|                                               |                                                                                                                                                                                                             |
| configure voip > sip-definition settings > 100- | **TimeoutBetween100And18x**                                                                                                                                                                                    |
| to-18x-timeout                                | 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).                                                                                                                      |
|                                               |                                                                                                                                                                                                             |
| configure voip > sip-definition settings > ignore- | **IgnoreRemoteSDPMKI**                                                                                                                                                                                          |
| remote-sdp-mki                                | 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  
<p>| | |
|                                               |                                                                                                                                                                                                             |
| configure voip &gt; sip-                         | Defines the echo canceller format in the outgoing SDP. The 'ecan'                                                                                                                                              |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| definition settings > sdp-ecan-frmt [SDPEcanFormat] | 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:** The parameter is applicable only when the IsFaxUsed parameter is set to 2, and for re-INVITE messages generated by the device as result of modem or fax tone detection. |
| First Call Ringback Tone ID | Defines the index of the first ringback tone in the CPT file. This option enables an Application server to request the device to play a distinctive ringback tone to the calling party according to the destination of the call. The tone is played according to the Alert-Info header received in the 180 Ringing SIP response (the value of the Alert-Info header is added to the value of the parameter). The valid range is -1 to 1,000. The default is -1 (i.e., play standard ringback tone).  
  **Note:**  
  - It is assumed that all ringback tones are defined in sequence in the CPT file.  
  - In case of an MLPP call, the device uses the value of the parameter plus 1 as the index of the ringback tone in the CPT file (e.g., if this value is set to 2, then the index is 2, i.e., 1 + 1). |
| Presence Publish IP Group ID | Assigns the IP Group (by ID) configured for the Skype for Business Server (presence server). This is where the device sends SIP PUBLISH messages to notify of changes in presence status of Skype for Business users when making and receiving calls using third-party endpoint devices. For more information on integration with Microsoft presence, see Microsoft Skype for Business Presence of Third-Party Endpoints on page 299. |
| Enable MsPresence message | Enables the device to notify (using SIP PUBLISH messages) Skype for Business Server (presence server) of changes in presence status of Skype for Business users when making and receiving calls using third-party endpoint devices.  
  - `[0]` Disable (default)  
  - `[1]` Enable  
  For more information on integration with Microsoft presence, see Microsoft Skype for Business Presence of Third-Party Endpoints on page 299. |
| Media IP Version Preference | 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_MedialIPVersionPreference). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see Configuring IP Profiles on page 417. |
| WebSocket Keep-Alive Period | Defines how often (in seconds) the device sends ping messages (keep alive) to check whether the WebSocket session with the Web client is still connected. |
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>websocket-keepalive</td>
<td>The valid value is 5 to 2000000. The default is 0 (i.e., ping messages are not sent). For more information on WebSocket, see SIP over WebSocket on page 568. <strong>Note:</strong> The device always replies to WebSocket ping control messages with pong messages.</td>
</tr>
</tbody>
</table>

#### Retransmission Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| SIP T1 Retransmission Timer | 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. |

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP T2 Retransmission Timer</td>
<td>Defines the maximum interval (in msec) between retransmissions of SIP messages (except for INVITE requests). The default is 4000. <strong>Note:</strong> The time interval between subsequent retransmissions of the same SIP message starts with SipT1Rtx and is multiplied by two until SipT2Rtx.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIP Maximum RTX</td>
<td>Defines the maximum number of UDP transmissions of SIP messages (first transmission plus retransmissions). The range is 1 to 30. The default is 7.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of RTX Before Hot-Swap</td>
<td>Defines the number of retransmitted INVITE/REGISTER messages before the call is routed (hot swap) to another Proxy/Registrar. The valid range is 1 to 30. The default is 3. For example, if configured to 3 and no response is received from an IP destination, the device attempts another three times to send the call to the IP destination. If still unsuccessful, it attempts to redirect the call to another IP destination. <strong>Note:</strong> The parameter is also used for alternative routing (see Alternative Routing Based on IP Connectivity).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| SIP Message Manipulations Table | 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,
```
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageManipulations_ActionSubject, MessageManipulations_ActionType, MessageManipulations_ActionValue, MessageManipulations_RowRole; [MessageManipulations]</td>
<td>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 more information, see Configuring SIP Message Manipulation on page 393.</td>
</tr>
</tbody>
</table>

### Message Policies Table

**configure voip > message message-policy [MessagePolicy]**

**configure voip > sip-definition settings > message-policy-reject-response-type [MessagePolicyRejectResponseType]**

Defines 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_Name, 
MessagePolicy_MaxMessageLength, 
MessagePolicy_MaxHeaderLength, MessagePolicy_MaxBodyLength, 
MessagePolicy_MaxNumHeaders, MessagePolicy_MaxNumBodies, 
MessagePolicy_SendRejection, MessagePolicy_MethodList, 
MessagePolicy_MethodListType, MessagePolicy_BodyList, 
MessagePolicy_BodyListType, 
MessagePolicy_UseMaliciousSignatureDB;
[/MessagePolicy]
```

For more information, see Configuring SIP Message Policy Rules.

### Pre-Parsing Manipulation Sets Table

**configure voip > message pre-parsing-manip-sets [PreParsingManipulationSets]**

Defines Pre-parsing Manipulation Sets.

The format of the ini file table parameter is as follows:

```
[PreParsingManipulationSets ]
FORMAT PreParsingManipulationSets_Index = PreParsingManipulationSets_Name; 
[/PreParsingManipulationSets ]
```

For more information, see Configuring Pre-parsing Manipulation Rules on page 403.

### Pre-Parsing Manipulation Rules Table

**configure voip > message pre-**

Defines Pre-parsing Manipulation rules.

The format of the ini file table parameter is as follows:

```
[PreParsingManipulationRules ]
FORMAT PreParsingManipulationRules_Index =
```

### 65.10 Coders and Profile Parameters

The profile parameters are described in the table below.

#### Table 65-34: Profile Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Coder Groups Table</strong></td>
<td></td>
</tr>
<tr>
<td>Coder Groups</td>
<td>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.</td>
</tr>
<tr>
<td>configure voip &gt; coders-</td>
<td></td>
</tr>
<tr>
<td>and-profiles audio-coders-</td>
<td>The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>groups</td>
<td></td>
</tr>
<tr>
<td>[AudioCodersGroups]</td>
<td>[AudioCoders]</td>
</tr>
<tr>
<td>FORMAT AudioCodersGroups_Index = AudioCodersGroups_Name;</td>
<td></td>
</tr>
<tr>
<td>FORMAT AudioCoders_Name</td>
<td></td>
</tr>
<tr>
<td>[AudioCoders]</td>
<td>FORMAT AudioCoders_Name = AudioCoders_AudioCodersGroupId, AudioCoders_AudioCodersIndex, AudioCoders_Name,</td>
</tr>
<tr>
<td>FORMAT AudioCoders_Name</td>
<td>AudioCoders_rate, AudioCoders_PayloadType, AudioCoders_Sce, AudioCoders_CoderSpecific;</td>
</tr>
<tr>
<td>Note: For a list of supported coders and for configuring Coder Groups, see &quot;Configuring Coder Groups&quot; on page 407.</td>
<td></td>
</tr>
<tr>
<td><strong>IP Profiles Table</strong></td>
<td></td>
</tr>
<tr>
<td>configure voip &gt; coders-</td>
<td></td>
</tr>
<tr>
<td>and-profiles ip-profile</td>
<td></td>
</tr>
<tr>
<td>[IPProfile]</td>
<td>Defines the IP Profiles table. The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>FORMAT IpProfile_Index = IpProfile_ProfileName,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpPreference, IpProfile_CodersGroupName,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpFaxUsed, IpProfile_JitterBufMinDelay,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpAssertOptFactor, IpProfile_IPDiffServ,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpAssertIPDiffServ, IpProfile_RTPRedundancyDepth,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpCNGmode, IpProfile_VxTransportType, IpProfile_NSEMode,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpDTMFUsed, IpProfile_PlayRBTone2IP,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpEnableEarlyMedia, IpProfile_ProgressIndicator2IP,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpEnableEcoCanceller, IpProfile_CopyDest2RedirectNumber,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpMediaSecurityBehaviour, IpProfile_CallLimit,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpDisconnectOnBrokenConnection, IpProfile_FirstTxDtmfOption,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpSecondTxDtmfOption, IpProfile_RxDtmfOption,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpEnableHold, IpProfile_InputGain, IpProfile_VoiceVolume,</td>
<td></td>
</tr>
<tr>
<td>FORMAT IpAddIEInSetup, IpProfile_SBCExtensionCodersGroupName,</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IpProfile_MediaIPVersionPreference, IpProfile_TranscodingMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCAllowedMediaTypes,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCAllowedAudioCodersGroupName,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCAllowedVideoCodersGroupName,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCAllowedCodersMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCMediaSecurityBehaviour,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRFC2833Behavior, IpProfile_SBCAlternativeDTMFMethod,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCSendMultipleDTMFMethods,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCAssertIdentity, IpProfile_AMDSensitivityParameterSuit,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_AMDSensitivityLevel,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_AMDMaxGreetingTime,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_AMDMaxPostSilenceGreetingTime,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCDiversionMode, IpProfile_SBCHistoryInfoMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_EnableQSIGTunneling,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCFaxCodersGroupName,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCFaxBehavior,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCFaxOfferMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCFaxAnswerMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SbcPrackMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCSessionExpiresMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteUpdateSupport,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteReinviteSupport,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteDelayedOfferSupport,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteReferBehavior,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemote3xxBehavior,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteMultiple18xSupport,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteEarlyMediaResponseType,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteEarlyMediaSupport,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_EnableSymmetricMKI,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_MKISize,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCEnforceMKISize,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteEarlyMediaRTP,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteSupportsRFC3960,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteCanPlayRingback,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_EnableEarly183,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCEarlyAnswerTimeout,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBC2833DTMFPayloadType,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCUserRegistrationTime,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_ResetSRTPStateUponRekey,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_AmdMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCReliableHoldToneSource,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_GenerateSRTPKeys,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteHoldFormat,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteReplacesBehavior,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCSDPPtimeAnswer,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCPreferredPTime,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCUseSilenceSupp,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRTCPRedundancyBehavior,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCPlayRBTToTransferree,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRTCPMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCJitterCompensation,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteRenegotiateOnFaxDetection,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_JitterBufMaxDelay,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCUserBehindUdpNATRegistrationTime,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCUserBehindTcpNATRegistrationTime,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCSDPHandleRTCPPort,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoveCryptoLifetimeInSDP,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCIceMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRTPMux,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCMediaSecurityMethod,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCHandleXDetect,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRTCPFeedback,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteRepresentationMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCKeepVIAHeaders,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCKeepRoutingHeaders,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCKeepUserAgentHeader,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteMultipleEarlyDialogs,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCRemoteMultipleAnswersMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCDirectMediaTag,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCAdaptRFC2833BToVoiceCoderBW,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_CreatedByRoutingServer,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCFaxReroutingMode,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCMaxCallDuration,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCGenerateRTCP,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCISUPBodyHandling,</td>
<td></td>
</tr>
<tr>
<td>IpProfile_SBCISSUPVariant,</td>
<td></td>
</tr>
</tbody>
</table>
### 65.11 Channel Parameters

This subsection describes the device's channel parameters.

#### 65.11.1 Voice Parameters

The voice parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Gain</td>
<td>Global parameter defining the pulse-code modulation (PCM) input (received) gain control level (in decibels). You can also configure the functionality per specific calls, using IP Profiles (IpProfile_InputGain). For a detailed description of the parameter and for configuring the functionality, see &quot;Configuring IP Profiles&quot; on page 417. Note: If the functionality is configured for a specific profile, the settings of the global parameter is ignored for calls associated with the profile.</td>
</tr>
<tr>
<td>Voice Volume</td>
<td>Global parameter defining the voice gain control (in decibels). This defines the level of the transmitted signal. You can also configure the functionality per specific calls, using IP Profiles (IpProfile_VoiceVolume). For a detailed description of the parameter and for configuring the functionality, see &quot;Configuring IP Profiles&quot; on page 417. Note: If the functionality is configured for a specific profile, the settings of the global parameter is ignored for calls associated with the profile.</td>
</tr>
</tbody>
</table>
| configure voip > media voice > codecs > G726-voice-payload-format [VoicePayloadFormat] | Determines the bit ordering of the G.726 voice payload format.  
- [0] = (Default) Little Endian  
- [1] = Big Endian  
**Note:** To ensure high voice quality when using G.726, both communicating ends should use the same endianness format. Therefore, when the device communicates with a third-party entity that uses the G.726 voice coder and voice quality is poor, change the settings of the parameter (between Big Endian and Little Endian). |
<p>| MF Transport Type | Currently, not supported. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Echo Canceler</td>
<td>Global parameter enabling echo cancellation (i.e., echo from voice calls is removed). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_EnableEchoCanceller). For a detailed description of the parameter and for configuring the functionality, see &quot;Configuring IP Profiles&quot; on page 417. <strong>Note:</strong> If the functionality is configured for a specific profile, the settings of this global parameter is ignored for calls associated with the profile.</td>
</tr>
<tr>
<td>Network Echo Suppressor Enable</td>
<td>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). <strong>Options:</strong></td>
</tr>
<tr>
<td>- [0] Disable (default)</td>
<td></td>
</tr>
<tr>
<td>- [1] Enable</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
<td></td>
</tr>
<tr>
<td>Echo Canceller Type</td>
<td>Defines the echo canceller type. <strong>Options:</strong></td>
</tr>
<tr>
<td>- [0] Line echo canceller = (Default) Echo canceller for Tel side.</td>
<td></td>
</tr>
<tr>
<td>Attenuation Intensity</td>
<td>Defines the acoustic echo suppressor signals identified as echo attenuation intensity. The valid range is 0 to 3. The default is 0.</td>
</tr>
<tr>
<td>Max ERL Threshold - DB</td>
<td>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.</td>
</tr>
<tr>
<td>Min Reference Delay x10 msec</td>
<td>Defines the acoustic echo suppressor minimum reference delay (in 10-ms units). The valid range is 0 to 40. The default is 0.</td>
</tr>
<tr>
<td>Max Reference Delay x10 msec</td>
<td>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).</td>
</tr>
<tr>
<td>configure voip &gt; media voice &gt; echo-canceller-hybrid-loss [ECHybridLoss]</td>
<td>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.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>configure voip &gt; media voice &gt; echo-canceller-NLP-mode [ECNLPMode]</td>
<td>Enables Non-Linear Processing (NLP) mode for echo cancellation.</td>
</tr>
<tr>
<td>configure voip &gt; media voice &gt; echo-canceller-aggressive-NLP [EchoCancellerAggressiveNLP]</td>
<td>Enables the Aggressive NLP at the first 0.5 second of the call.</td>
</tr>
<tr>
<td>configure voip &gt; media RTP-RTCP &gt; number-of-SID-coefficients [RTPSIDCoeffNum]</td>
<td>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].</td>
</tr>
</tbody>
</table>

**Answer Detector (AD) Parameters**

| Enable Answer Detector [EnableAnswerDetector] | Currently, not supported. |
| Answer Detector Activity Delay [AnswerDetectorActivityDelay] | Defines the time (in 100-msec resolution) between activating the Answer Detector and the time that the detector actually starts to operate. The valid range is 0 to 1023. The default is 0. |
| Answer Detector Silence Time [AnswerDetector SilenceTime] | Currently, not supported. |
| Answer Detector Redirection [AnswerDetectorRedirection] | Currently, not supported. |
| Answer Detector Sensitivity [AnswerDetectorSensitivity] | Defines the Answer Detector sensitivity. The range is 0 (most sensitive) to 2 (least sensitive). The default is 0. |

**65.11.2 Coder Parameters**

The coder parameters are described in the table below.

**Table 65-36: Coder Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Silk Tx Inband FEC [silk-tx-inband-fec]</td>
<td>Enables forward error correction (FEC) for the SILK coder.</td>
</tr>
<tr>
<td>[0] Disable (default)</td>
<td></td>
</tr>
</tbody>
</table>
65.11.3 DTMF Parameters

The dual-tone multi-frequency (DTMF) parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTMF Transport Type</td>
<td>Determines the DTMF transport type.</td>
</tr>
<tr>
<td>configure voip &gt; media voice &gt; DTMF-transport-type [DTMFTransportType]</td>
<td>- [0] Mute DTMF = DTMF digits are removed from the voice stream and are not relayed to remote side.</td>
</tr>
<tr>
<td></td>
<td>- [3] RFC 2833 Relay DTMF = (Default) DTMF digits are removed from the voice stream and are relayed to remote side according to RFC 2833.</td>
</tr>
<tr>
<td></td>
<td>- [7] RFC 2833 Relay Decoder Mute = DTMF digits are sent according to RFC 2833 and muted when received.</td>
</tr>
<tr>
<td></td>
<td>Note: The parameter is automatically updated if the parameters FirstTxDTMFOption or RxDTMFOption are configured.</td>
</tr>
<tr>
<td>DTMF Volume (-31 to 0 dB)</td>
<td>Global parameter defining the DTMF gain control value (in decibels).</td>
</tr>
</tbody>
</table>
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTMF-volume</td>
<td>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. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>inter-digit-interval</td>
<td>Defines the time (in msec) between generated DTMF digits (if FirstTxDTMFOption = 1, 2 or 3). The valid range is 0 to 32767. The default is 100.</td>
</tr>
<tr>
<td>[DTMFDigitLength]</td>
<td>Defines the time (in msec) for generating DTMF tones (if FirstTxDTMFOption = 1, 2 or 3). It also configures the duration that is sent in INFO (Cisco) messages. The valid range is 0 to 32767. The default is 100.</td>
</tr>
<tr>
<td>RxDTMFHangOverTime</td>
<td>Defines the Voice Silence time (in msec) after playing DTMF or MF digits that arrive as Relay. Valid range is 0 to 2,000 msec. The default is 1,000 msec.</td>
</tr>
<tr>
<td>TxDTMFHangOverTime</td>
<td>Defines the Voice Silence time (in msec) after detecting the end of DTMF or MF digits when the DTMF Transport Type is either Relay or Mute. Valid range is 0 to 2,000 msec. The default is 1,000 msec.</td>
</tr>
<tr>
<td>NTEMaxDuration</td>
<td>Defines the maximum time for sending Named Telephony Events / NTEs (RFC 4733/2833 DTMF relay), regardless of the DTMF signal duration on the other side. The range is -1 to 200,000,000 msec. The default is -1 (i.e., NTE stops only upon detection of an End event).</td>
</tr>
</tbody>
</table>

### 65.11.4 RTP, RTCP and T.38 Parameters

The RTP, RTCP and T.38 parameters are described in the table below.

**Table 65-38: RTP/RTCP and T.38 Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broken Connection Mode</td>
<td>Global parameter that defines the device’s handling of 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 the parameter and for configuring this functionality in the IP Profiles table, see ‘Configuring IP Profiles’ on page 417. <strong>Note:</strong> 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>
</tbody>
</table>

---

User's Manual 912 Document #: LTRT-41877
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broken Connection Timeout</td>
<td>Defines the timeout interval (in 100-msec units) after which a call is disconnected if an RTP packet is not received during an established call (i.e., RTP flow suddenly stops during the call). The valid range is from 3 (i.e., 300 msec) to approx. 2684354 (i.e., 74.5 hours). The default is 100 (i.e., 10000 msec or 10 seconds).</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: The parameter is applicable only if the DisconnectOnBrokenConnection parameter is configured to 1.</td>
</tr>
<tr>
<td>configure voip &gt; sip-definition</td>
<td></td>
</tr>
<tr>
<td>settings &gt; broken-connection-</td>
<td></td>
</tr>
<tr>
<td>event-timeout</td>
<td></td>
</tr>
<tr>
<td>[BrokenConnectionEventTimeout]</td>
<td></td>
</tr>
<tr>
<td>configure voip &gt; sbc settings</td>
<td>Defines the timeout interval (in msec) after which a call is disconnected if RTP packets are not received. The timer begins from call setup and if no packets have been received when the timer expires, the device disconnects the call. The valid range is 0-50000. The default is 0 (i.e., disconnects the call immediately).</td>
</tr>
<tr>
<td>no-rtp-detection-timeout</td>
<td></td>
</tr>
<tr>
<td>[NoRTPDetectionTimeout]</td>
<td></td>
</tr>
<tr>
<td>Dynamic Jitter Buffer Minimum</td>
<td>Global parameter defining the minimum delay (in msec) of the device’s dynamic Jitter Buffer. You can also configure the functionality per specific calls, using IP Profiles (IpProfile_JitterBufMinDelay). For a detailed description of the parameter and for configuring the functionality, see Configuring IP Profiles on page 417.</td>
</tr>
<tr>
<td>Delay</td>
<td></td>
</tr>
<tr>
<td>configure voip &gt; media rtp-rtcp</td>
<td></td>
</tr>
<tr>
<td>&gt; jitter-buffer-minimum-delay</td>
<td></td>
</tr>
<tr>
<td>[DJBufMinDelay]</td>
<td></td>
</tr>
<tr>
<td>Dynamic Jitter Buffer Optimization Factor</td>
<td>Global parameter defining the Dynamic Jitter Buffer frame error/delay optimization factor. You can also configure the functionality per specific calls, using IP Profiles (IpProfile_JitterBufOptFactor). For a detailed description of the parameter and for configuring the functionality, see Configuring IP Profiles on page 417.</td>
</tr>
<tr>
<td>configure voip &gt; media rtp-rtcp</td>
<td></td>
</tr>
<tr>
<td>&gt; jitter-buffer-optimization-factor</td>
<td></td>
</tr>
<tr>
<td>[DJBufOptFactor]</td>
<td></td>
</tr>
<tr>
<td>RTP Redundancy Depth</td>
<td>Global parameter that enables the device to generate RFC 2198 redundant packets. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_RTPRedundancyDepth). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see “Configuring IP Profiles” on page 417.</td>
</tr>
<tr>
<td>configure voip &gt; media rtp-rtcp</td>
<td></td>
</tr>
<tr>
<td>&gt; RTP-redundancy-depth</td>
<td></td>
</tr>
<tr>
<td>[RTPRedundancyDepth]</td>
<td></td>
</tr>
<tr>
<td>RFC 2198 Payload Type</td>
<td>Defines the RTP redundancy packet payload type (according to RFC 2198).</td>
</tr>
<tr>
<td>configure voip &gt; media rtp-rtcp</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>RTP-redundancy-payload-type (RFC2198PayloadType)</td>
<td>The valid value is 96 to 127. The default is 104. <strong>Note:</strong> The parameter is applicable only if the RTPRedundancyDepth parameter is set to 1.</td>
</tr>
<tr>
<td>Packing Factor (RTPPackingFactor)</td>
<td>N/A. Controlled internally by the device according to the selected coder.</td>
</tr>
<tr>
<td>RFC 2833 TX Payload Type (RFC2833TxPayloadType)</td>
<td>Defines the Tx RFC 2833 DTMF relay dynamic payload type for outbound calls. The valid range is 96 to 127. The default is 96. <strong>Note:</strong> When RFC 2833 payload type negotiation is used (i.e., the parameter FirstTxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.</td>
</tr>
<tr>
<td>RFC 2833 RX Payload Type (RFC2833RxPayloadType)</td>
<td>Defines the Rx RFC 2833 DTMF relay dynamic payload type for inbound calls. The valid range is 96 to 127. The default is 96. <strong>Note:</strong> When RFC 2833 payload type negotiation is used (i.e., the parameter FirstTxDTMFOption is set to 4), this payload type is used for the received DTMF packets. If negotiation isn't used, this payload type is used for receive and for transmit.</td>
</tr>
<tr>
<td>[EnableDetectRemoteMACChange]</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>• [0] = Nothing is changed.</td>
</tr>
<tr>
<td></td>
<td>• [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.</td>
</tr>
<tr>
<td></td>
<td>• [2] = (Default) The device uses the received GARP packets to change the MAC address of the transmitted RTP packets.</td>
</tr>
<tr>
<td></td>
<td>• [3] = Options 1 and 2 are used. <strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>• For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td></td>
<td>• If the device is located in a network subnet which is connected to other gateways using a router that uses Virtual Router Redundancy Protocol (VRRP) for redundancy, then set the parameter to 0 or 2.</td>
</tr>
<tr>
<td>RTP Base UDP Port (BaseUDPport)</td>
<td>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 &quot;Configuring RTP Base UDP Port&quot; on page 195. The range of possible UDP ports is 6,000 to 65,535. The default base UDP port is 6000. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>configure voip &gt; media rtp-rtcp &gt; base-udp-port [BaseUDPport]</td>
<td></td>
</tr>
<tr>
<td>configure voip &gt; media rtp-rtcp &gt; udp-port-spacing</td>
<td>Defines the port spacing (&quot;jumps&quot;) of local UDP ports allocated by the device to media channels (legs) within the configured</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>[UdpPortSpacing]</td>
<td>port range.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td>▪ A device reset is required for the parameter to take effect.</td>
</tr>
<tr>
<td></td>
<td>▪ For more information on configuring the UDP port range, see Configuring RTP Base UDP Port on page 195.</td>
</tr>
</tbody>
</table>

**No-Op Packets Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>no-operation-enable</td>
<td>Enables the transmission of RTP or T.38 No-Op packets.</td>
</tr>
<tr>
<td>[NoOpEnable]</td>
<td>▪ [0] = Disable (default)</td>
</tr>
<tr>
<td></td>
<td>▪ [1] = Enable</td>
</tr>
<tr>
<td></td>
<td>This mechanism ensures that the NAT binding remains open during RTP or T.38 silence periods.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NoOpInterval]</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The valid range is 20 to 65,000 msec. The default is 10,000.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> To enable No-Op packet transmission, use the NoOpEnable parameter.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>no-operation-interval</td>
<td>Defines the payload type of No-Op packets.</td>
</tr>
<tr>
<td>[RTPNoOpPayloadType]</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> When defining the parameter, ensure that it doesn't cause collision with other payload types.</td>
</tr>
</tbody>
</table>

**RTP Control Protocol Extended Reports (RTCP XR) Parameters**

For more information on RTCP XR, see "Configuring RTCP XR" on page 757.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable RTCP XR</td>
<td>Enables voice quality monitoring and RTCP XR, according to RFC 3611.</td>
</tr>
<tr>
<td>configure voip &gt; media rtp-rtcp &gt; voice-quality-monitoring-enable</td>
<td>▪ [0] Disable (default)</td>
</tr>
<tr>
<td>[VQMonEnable]</td>
<td>▪ [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.</td>
</tr>
<tr>
<td></td>
<td>▪ [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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Gap Size</td>
<td>Defines the voice quality monitoring - minimum gap size (number of frames).</td>
</tr>
<tr>
<td>[VQMonGMin]</td>
<td>The default is 16.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Burst Threshold [VQMonBurstHR]</td>
<td>Defines the voice quality monitoring - excessive burst alert threshold. The default is -1 (i.e., no alerts are issued).</td>
</tr>
<tr>
<td>Delay Threshold [VQMonDelayTHR]</td>
<td>Defines the voice quality monitoring - excessive delay alert threshold. The default is -1 (i.e., no alerts are issued).</td>
</tr>
<tr>
<td>R-Value Delay Threshold [VQMonEOCRValTHR]</td>
<td>Defines the voice quality monitoring - end of call low quality alert threshold. The default is -1 (i.e., no alerts are issued).</td>
</tr>
<tr>
<td>RTCP XR Packet Interval [RTCPInterval]</td>
<td>Defines the time interval (in msec) between adjacent RTCP XR reports. This interval starts from call establishment. Thus, the device can send RTCP XR reports during the call, in addition to at the end of the call. If the duration of the call is shorter than this interval, RTCP XR is sent only at the end of the call. The valid value range is 0 to 65,535. The default is 5,000.</td>
</tr>
</tbody>
</table>
| Disable RTCP XR Interval Randomization [DisableRTCPRandomize] | Determines whether RTCP report intervals are randomized or whether each report interval accords exactly to the parameter RTCPInterval.  
  - [0] Disable = (Default) Randomize  
  - [1] Enable = No Randomize |
| SBC RTCP XR Report Mode [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 |
| Publication IP Group ID [PublicationIPGroupID] | Defines the IP Group to where the device sends RTCP XR reports. By default, no value is defined. |

### 65.12 SBC Parameters

The SBC and CRP parameters are described in the table below.

**Table 65-39: SBC and CRP Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRP-specific Parameters</td>
<td></td>
</tr>
</tbody>
</table>
| CRP Application [EnableCRPApplication] | Enables the CRP application.  
  - [0] Disable (default)  
  - [1] Enable  
  **Note:** For the parameter to take effect, a device reset is required. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| CRP Survivability Mode | Defines the CRP mode.  
- [0] Standard Mode (default)  
- [1] Always Emergency Mode  
- [2] Auto-answer REGISTER |
| configure voip > sbc settings > crp-survivability-mode | Enables fallback routing from the proxy server to the Gateway (PSTN).  
- [0] = Disable (default)  
- [1] = Enable |
| SBC-specific Parameters | Enables the Session Border Control (SBC) application.  
- [0] Disable  
- [1] Enable (default)  
**Note:**  
- For the parameter to take effect, a device reset is required.  
- The parameter is enabled by default only if the License Key contains at least one of the SBC-related capacity features (e.g., “SBC-Signaling”); otherwise, the parameter is disabled. |
| SBC Maximum Call Duration | Determines whether incoming calls that cannot be classified (i.e., classification process fails) to a Source IP Group are rejected or processed.  
- [0] Reject = (Default) Call is rejected if classification fails.  
- [1] Allow = If classification fails, the incoming packet is assigned to a source IP Group (and subsequently processed) as follows:  
  - The source SRD is determined according to the SIP Interface to where the SIP-initiating dialog request is sent. The source IP Group is set to the default IP Group associated with this SRD.  
  - If the source SRD is ID 0, then source IP Group ID 0 is chosen. In case of any other SRD, then the first IP Group associated with this SRD is chosen as the source IP Group or the call. If no IP Group is associated with this SRD, the call is rejected. |
| SBC and CRP Parameters | Defines the maximum duration (in minutes) per SBC call (global). If the duration is reached, the device terminates the call.  
The valid range is 0 to 35,791, where 0 is unlimited duration. The default is 0.  
**Note:** You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCMaxCallDuration). For a detailed description of the parameter and for configuring this |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBC No Answer Timeout</td>
<td>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.</td>
</tr>
</tbody>
</table>
| configure voip > sbc settings > num-of-subscribes | 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.  
**Note:**  
- For the parameter to take effect, a device reset is required.  
- The maximum number of SUBSCRIBE sessions can be increased by reducing the maximum number of SBC channels in the License Key. For every reduced SBC session, the device gains two SUBSCRIBE sessions. |
| configure voip > sbc settings > sbc-dialog-subsc-route-mode | Enables the device to route in-dialog, refresh SIP SUBSCRIBE requests to the "working" (has connectivity) proxy.  
- [0] = (Default) Disable – the device sends in-dialog, refresh SUBSCRIBES according to the address in the Contact header of the 200 OK response received from the proxy to which the initial SUBSCRIBE was sent (as per the SIP standard).  
- [1] = Enable – the device routes in-dialog, refresh SUBSCRIBES to the "working" proxy (regardless of the Contact header). The "working" proxy (address) is determined by the device’s keep-alive mechanism for the Proxy Set that was used to route the initial SUBSCRIBE.  
**Note:** For this feature to be functional, ensure the following:  
- Keep-alive mechanism is enabled for the Proxy Set ('Proxy Keep-Alive' parameter is set to any value other than **Disable**).  
- Load-balancing between proxies is disabled |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| configure voip > sbc settings > sbc-max-fwd-limit | Defines the Max-Forwards SIP header value. The Max-Forwards header is used to limit the number of servers (such as proxies) that can forward the SIP request. The Max-Forwards value indicates the remaining number of times this request message is allowed to be forwarded. This count is decremented by each server that forwards the request. The parameter affects the Max-Forwards header in the received message as follows:  
  - If the received header's original value is 0, the message is not passed on and is rejected.  
  - If the received header's original value is less than the parameter's value, the header's value is decremented before being sent on.  
  - If the received header's original value is greater than the parameter's value, the header's value is replaced by the user-defined parameter's value.  
  The valid value range is 1-70. The default is 10. |
| SBC MaxForwardsLimit | ('Proxy Load Balancing Method' parameter is set to Disable). |
| configure voip > sbc settings > sbc-sess-exp-time | 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. |
| [SBCSessionExpires] | Minimum Session-Expires | 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. |
| configure voip > sbc settings > min-session-expires | | |
| [SBCMinSE] | configure voip > sbc settings > sbc-session-refresh-policy | 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, the parameter is useful when a UA does not | [SBCSessionRefreshingPolicy] |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>• [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'.</td>
</tr>
<tr>
<td></td>
<td>• [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'.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: 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.</td>
</tr>
<tr>
<td>User Registration Grace Time</td>
<td>Defines additional time (in seconds) to add to the registration expiry time users that are registered in the device's Users Registration database. The valid value is 0 to 2,000,000. The default is 0. For more information, see Registration Refreshes on page 461.</td>
</tr>
<tr>
<td>SBC DB Routing Search Mode</td>
<td>Defines the method for searching a registered user in the device's User Registration database when a SIP INVITE message is received for routing to a user. If the registered user is found (i.e., destination URI in INVITE), the device routes the call to the user's corresponding contact address specified in the database.</td>
</tr>
<tr>
<td></td>
<td>• [0] All permutations = (Default) Device searches for the user in the database using the entire Request-URI (user@host). If not found, it searches for the user part of the Request-URI. For example, it first searches for &quot;<a href="mailto:4709@joe.company.com">4709@joe.company.com</a>&quot; and if not found, it searches for &quot;4709&quot;.</td>
</tr>
<tr>
<td></td>
<td>• [1] Dest URI dependant = Device searches for the user in the database using the entire Request-URI (user@host) only. For example, it searches for &quot;<a href="mailto:4709@joe.company.com">4709@joe.company.com</a>&quot;.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: If the Request-URI contains the &quot;tel:&quot; URI or &quot;user=phone&quot; parameter, the device searches only for the user part.</td>
</tr>
<tr>
<td>Handle P-Asserted-Identity</td>
<td>Global parameter that defines the handling of the SIP P-Asserted-Identity header. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCAssertIdentity). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see &quot;Configuring IP Profiles&quot; on page 417.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Keep original user in Register</strong>&lt;br&gt;configure voip &gt; sbc settings &gt; keep-contact-user-in-reg&lt;br&gt;[SBCKeepContactUserinRegister]</td>
<td>Determines whether the device replaces the Contact user with a unique Contact user in the outgoing message in response to a REGISTER request.</td>
</tr>
<tr>
<td><strong>SBC Remote Refer Behavior</strong>&lt;br&gt;configure voip &gt; sbc settings &gt; sbc-refer-bhvr&lt;br&gt;[SBCReferBehavior]</td>
<td>Global parameter that defines the handling of SIP REFER requests. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCRemoteReferBehavior). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see &quot;Configuring IP Profiles&quot; on page 417.</td>
</tr>
<tr>
<td><strong>configure voip &gt; sbc settings &gt; sbc-xfer-prefix&lt;br&gt;[SBCXferPrefix]</strong></td>
<td>When the SBCReferBehavior is set to 1, the device, while interworking the SIP REFER message, adds the prefix &quot;T~&amp;R-&quot; 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. <strong>Note:</strong> This feature is also applicable to 3xx redirect responses. The device adds the prefix &quot;T~&amp;R-&quot; to the URI user part in the Contact header if the SBC3xxBehavior parameter is set to 1.</td>
</tr>
<tr>
<td><strong>configure voip &gt; sbc settings &gt; sbc-3xx-bhvt&lt;br&gt;[SBC3xxBehavior]</strong></td>
<td>Global parameter that defines the handling of SIP 3xx redirect responses. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCRemote3xxBehavior). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see &quot;Configuring IP Profiles&quot; on page 417. <strong>Note:</strong> 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>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| configure voip > sbc settings > enforce-media-order [SBCEnforceMediaOrder] | Enables the device to include all previously negotiated media lines within the current session ('m=' line) in the SDP offer-answer exchange (RFC 3264).  
   - [0] Disable (default)  
   - [1] Enable  
   For example, assume a call (audio) has been established between two endpoints and one endpoint wants to subsequently send an image in the same call session. If the parameter is enabled, the endpoint includes the previously negotiated media type (i.e., audio) with the new negotiated media type (i.e., image) in its SDP offer:  
   ```
   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 the parameter is disabled, the only ‘m=’ line included in the SDP is the newly negotiated media (i.e., image). |
| SBC Diversion URI Type configure voip > sbc settings > sbc-diversion-uri-type [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 Profiles table determine the call redirection (diversion) SIP header to use - History-Info or Diversion. |
| SBC Server Auth Mode configure voip > sbc settings > 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 RADIUS server.  
   - [1] = Authentication is done by the RFC 5090 compliant RADIUS server.  
   **Note:** Currently, option [1] is not supported. |
| Lifetime of the nonce in seconds configure voip > sbc settings > lifetime-of-nonce [AuthNonceDuration] | Defines the lifetime (in seconds) that the current nonce is valid for server-based authentication. The device challenges a message that attempts to use a server nonce beyond this period. The parameter is used to provide replay protection (i.e., ensures that old communication streams are not used in replay attacks). The valid value range is 30 to 600. The default is 300. |
### Parameter | Description
--- | ---
**Authentication Challenge Method**
configure voip > sbc settings > auth-chlng-mthd [AuthChallengeMethod] | Defines the type of server-based authentication challenge.
- **0** = (Default) Send SIP 401 "Unauthorized" with a WWW-Authenticate header as the authentication challenge response.
- **1** = Send SIP 407 "Proxy Authentication Required" with a Proxy-Authenticate header as the authentication challenge response.

**Authentication Quality of Protection**
configure voip > sbc settings > 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** = 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** = 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** = (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** = No 'qop' parameter is offered in the SIP 401 challenge message.

**SBC User Registration Time**
configure voip > sbc settings > sbc-usr-rgstr-time [SBCUserRegistrationTime] | Global parameter that defines the duration (in seconds) of the periodic registrations that occur between the user and the device (the device responds with this value to the user). You can also configure this functionality per specific calls, using IP Profiles (IpProfile_SBCUserRegistrationTime). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417.

**SBC Proxy Registration Time**
 | Defines the duration (in seconds) for which the user is
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure voip &gt; sbc settings &gt; sbc-prxy-rgstr-time [SBCProxyRegistrationTime]</td>
<td>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.</td>
</tr>
</tbody>
</table>
| configure voip > sbc settings > sbc-rand-expire [SBCRandomizeExpires] | Defines a value (in seconds) that is used to calculate a new value for the expiry time in the Expires header of SIP 200 OK responses for user registration and subscription requests from users. The expiry time value appears in the Expires header in REGISTER and SUBSCRIBE SIP messages. When the device receives such a request from a user, it forwards it to the proxy or registrar server. Upon a successful registration or subscription, the server sends a SIP 200 OK response. If the expiry time was unchanged by the server, the device applies this feature and changes the expiry time in the SIP 200 OK response before forwarding it to the user; otherwise, the device does not change the expiry time. This feature is useful in scenarios where multiple users may refresh their registration or subscription simultaneously, thereby causing the device to handle many such sessions at a given time. This may result in an overload of the device (reaching maximum session capacity), thereby preventing the establishment of new calls or preventing the handling of some user registration or subscription requests. When this feature is enabled, the device assigns a random expiry time to each user registration or subscription and thus, ensuring future user registration and subscription requests are more distributed over time (i.e., do not all occur simultaneously). The device takes any random number between 0 and the value configured by the parameter, and then subtracts this random number from the original expiry time value. For example, assume that the original expiry time is 120 and the parameter is set to 10. If the device randomly chooses the number 5 (i.e., between 0 and 10), the resultant expiry time will be 115 (120 minus 5). The valid value is 0 to 20. The default is 10. If set to 0, the device does not change the expiry time. **Note:**  
  - The lowest expiry time that the device sends in the 200 OK, regardless of the resultant calculation, is 10 seconds. For example, if the original expiry time is 12 seconds and the parameter is set to 5, theoretically, the new expiry time can be less than 10 (e.g., 12 – 4 = 8). However, the expiry time will be set to 10. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SBC Survivability Registration Time</strong>&lt;br&gt; configure voip &gt; sbc settings &gt; sbc-survrgstr-time&lt;br&gt; [SBCSurvivabilityRegistrationTime]</td>
<td>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.</td>
</tr>
<tr>
<td><strong>configure voip &gt; sbc settings &gt; sas-notice</strong>&lt;br&gt; [SBCEnableSurvivabilityNotice]</td>
<td>Enables the device to notify Aastra IP phones that the device is currently operating in Survivability mode.</td>
</tr>
</tbody>
</table>
| | • [0] = Disable  
| | • [1] = Enable  
| | For more information, see "Enabling Survivability Display on Aastra IP Phones" on page 579. |
| **SBC Dialog-Info Interworking**<br> configure voip > sbc settings > sbc-dialog-info-interwork<br> [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 487. |
| **configure voip > sbc settings > sbc-keep-call-id**<br> [SBCKeepOriginalCallId] | Global parameter that enables the device to use the same call identification (SIP Call-ID header value) received in incoming messages for the call identification in outgoing messages. The call identification value is contained in the SIP Call-ID header.  
| | You can also configure the functionality per specific calls, using IP Profiles. For a detailed description of the parameter and for configuring the functionality in the IP Profiles table, see Configuring IP Profiles on page 417. |
| **SBC GRUU Mode**<br> configure voip > sbc settings > sbc-gruu-mode<br> [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
## Bye Authentication

configure voip > sbc settings > 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.

## SBC Enable Subscribe Trying

configure voip > sbc settings > 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

## BroadWorks Survivability Feature

configure voip > sbc settings > sbc-broadworks-survivability

### [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** Disable = (Default) Normal processing of REGISTER messages.
- **1** Enable = 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 "Enabling Auto-Provisioning of Subscriber-Specific Information of BroadWorks Server for Survivability" on page 574.

## SBC Direct Media

configure voip > sip-interface > sbc-direct-

Enables the Direct Media feature (i.e., no Media Anchoring) for all SBC calls, whereby SIP signaling is handled by the device without handling the public GRUU to users. (Currently not supported.)

The parameter allows the device to act as a GRUU server for its SIP UA clients, providing them with public GRUU’s, according to RFC 5627. The public GRUU provided to the client is denoted in the SIP Contact header parameters, "pub-gruu". Public GRUU remains the same over registration expirations. On the other SBC leg communicating with the Proxy/Registrar, the device acts as a GRUU client. The device creates a GRUU value for each of its registered clients, which is mapped to the GRUU value received from the Proxy server. In other words, the created GRUU value is only used between the device and its clients (endpoints).

Public-GRUU:

```
sip:userA@domain.com;gr=unique-id
```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>media</td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>- [0] Disable = (Default) All calls traverse the device (i.e., no direct media).</td>
</tr>
<tr>
<td></td>
<td>- [1] Enable = Direct media flow between endpoints for all SBC calls.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The setting of direct media in the SIP Interfaces table overrides this global parameter. In other words, even if the parameter is disabled for direct media (i.e., Media Anchoring is enabled), if direct media is enabled for a SIP Interface (in the SIP Interfaces table), calls between endpoints belonging to the SIP Interface employ direct media.</td>
</tr>
<tr>
<td></td>
<td>- For more information on No Media Anchoring, see &quot;Direct Media&quot; on page 464.</td>
</tr>
</tbody>
</table>

Transcoding Mode
configure voip > sbc settings > 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 the parameter and for configuring this functionality in the IP Profiles table, see Configuring IP Profiles on page 417.

**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile.

Preferences Mode
configure voip > sbc settings > sbc-preferences
[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 Audio Coders Groups 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 Audio Coders Groups table.                                             |
|                    | **Note:** The parameter is applicable only if a Coders Group for Extension coders is assigned to the IP Profile (IPProfile_SBCExtensionCodersGroupName).                                           |

SBC RTCP Mode
configure voip > sbc settings > sbc-rtcp-mode
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
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| [SBCRTCPMode] | description of the parameter and for configuring this functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417.  
**Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |

**SBC Send Invite To All Contacts**

configure voip > sbc settings > 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 573.  

**SBC Shared Line Registration Mode**

configure voip > sbc settings > sbc-shared-line-reg-mode  
[SBCSharedLineRegMode]  

Enables the termination on the device of SIP REGISTER messages from secondary lines that belong to the Shared Line feature.  
- **[0]** Disable = (Default) Device forwards the REGISTER messages as is (i.e., not terminated on the device).  
- **[1]** Enable = REGISTER messages of secondary lines are terminated on the device.  
**Note:** The device always forwards REGISTER messages of the primary line.  

**SBC Forking Handling Mode**

configure voip > sbc settings > 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 discarded.  

**Configure voip > sbc settings > sbc-media-sync**  
[EnableSBCMediaSync]  

Enables synchronization of media between two SIP user agents when a call is established between them. Media synchronization means that the media is properly negotiated (SDP offer/answer) between the user agents. In some scenarios, the call is established despite the media not being synchronized. This may occur, for example, in call transfer (SIP REFER) where the media between the transfer target and transferee are not synchronized. The device performs media synchronization by sending a re-INVITE immediately after the call is established in order for the user agents to negotiate the media (SDP offer/answer).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[SBCRemoveSIPSFromNonSecuredTransport]</td>
<td>Defines the SIP headers for which the device replaces “sips:” with “sip:” in the outgoing SIP-initiating dialog request (e.g., INVITE) when the destination transport type is unsecured (e.g., UDP). (The “sips:” URI scheme indicates secured transport, for example, TLS.)</td>
</tr>
<tr>
<td>[SBCFaxDetectionTimeout]</td>
<td>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 Profiles table (see the IpProfile_SBCRemoteRenegotiateOnFaxDetection parameter in Configuring IP Profiles on page 417).</td>
</tr>
<tr>
<td>Admission Control Table</td>
<td>Defines Call Admission Control (CAC) rules. The format of the ini file table parameter is as follows: [SBCAdmissionControl] FORMAT SBCAdmissionControl_Index = SBCAdmissionControl_AdmissionControlName, SBCAdmissionControl_LimitType, SBCAdmissionControl_IPGroupName, SBCAdmissionControl_SRDNName, SBCAdmissionControl_SIPInterfaceName, SBCAdmissionControl_RequestType, SBCAdmissionControl_RequestDirection, SBCAdmissionControl_Limit, SBCAdmissionControl_LimitPerUser, SBCAdmissionControl_Rate,</td>
</tr>
</tbody>
</table>

- [0] Disable = (Default) Media synchronization is performed only if the RTP mode (e.g., a=sendrecv, a=sendonly, a=recvonly, and a=inactive) between the user agents are different and synchronization is required.
- [1] Enable = Media synchronization is performed if the media, including RTP mode or any other media such as coders, is different and has not been negotiated between the user agents.
- [2] Never = Media synchronization is never performed.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SBCAdmissionControl_MaxBurst, SBCAdmissionControl_Reservation; [SBCAdmissionControl]</td>
<td>For more information, see &quot;Configuring Admission Control&quot; on page 489.</td>
</tr>
</tbody>
</table>

**Allowed Audio Coders Table**

Allowed Audio Coders  
configure voip > coders-and-profiles allowed-audio-coders <group index > coder index>  
[AllowedAudioCoders]  
Defines audio coders for the Allowed Audio Coders Group.  
The format of the ini file table parameter is as follows:  
[ AllowedAudioCoders ]  
FORMAT AllowedAudioCoders_Index = AllowedAudioCoders_AllowedAudioCodersGroupName, AllowedAudioCoders_AllowedAudioCodersIndex, AllowedAudioCoders_CoderID, AllowedAudioCoders_UserDefineCoder;  
[ \AllowedAudioCoders ]  
For more information, see "Configuring Allowed Audio Coder Groups" on page 412.

**Allowed Audio Coders Groups Table**

Allowed Audio Coders Groups  
configure voip > coders-and-profiles allowed-audio-coders-groups  
[AllowedAudioCodersGroups]  
Defines the index and name of the Allowed Audio Coders Group.  
The format of the ini file table parameter is as follows:  
[ AllowedAudioCodersGroups ]  
FORMAT AllowedAudioCodersGroups_Index = AllowedAudioCodersGroups_Name;  
[ \AllowedAudioCodersGroups ]  
For more information, see "Configuring Allowed Audio Coder Groups" on page 412.

**Allowed Video Coders Groups Table**

Allowed Video Coders Groups  
configure voip > coders-and-profiles allowed-video-coders-groups  
[AllowedVideoCodersGroups]  
Defines the index and name of the Allowed Video Coders Group.  
The format of the ini file table parameter is as follows:  
[ AllowedVideoCodersGroups ]  
FORMAT AllowedVideoCodersGroups_Index = AllowedVideoCodersGroups_Name;  
[ \AllowedVideoCodersGroups ]  
For more information, see "Configuring Allowed Video Coder Groups" on page 415.

**Allowed Video Coders Table**

Allowed Video Coders  
coders-and-profiles allowed-video-coders <group index > coder index>  
[AllowedVideoCoders]  
Defines video coders for the Allowed Video Coders Group.  
The format of the ini file table parameter is as follows:  
[ AllowedVideoCoders ]  
FORMAT AllowedVideoCoders_Index = AllowedVideoCoders_AllowedVideoCodersGroupName, AllowedVideoCoders_AllowedVideoCodersIndex, AllowedVideoCoders_UserDefineCoder;  
[ \AllowedVideoCoders ]
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Classification Table</strong></td>
<td>Defines call Classification rules. The format of the ini file table parameter is as follows: [Classification] FORMAT Classification_Index = Classification_ClassificationIndex, Classification_ClassificationName, Classification_MessageConditionName, Classification_SRDNName, Classification_SrcSIPInterfaceName, Classification_SrcAddress, Classification_SrcPort, Classification_SrcTransportType, Classification_SrcUsernamePrefix, Classification_SrcHost, Classification_DestUsernamePrefix, Classification_DestHost, Classification_ActionType, Classification_SrcIPGroupName, Classification_DestRoutingPolicy, Classification_IpProfileName; []Classification ] For more information, see &quot;Configuring Classification Rules&quot; on page 493.</td>
</tr>
<tr>
<td><strong>Condition Table</strong></td>
<td>Defines SIP Message Condition rules. [ConditionTable ] FORMAT ConditionTable_Index = ConditionTable_Condition, ConditionTable_Description; []ConditionTable ] For more information, see &quot;Configuring Message Condition Rules&quot; on page 501.</td>
</tr>
<tr>
<td><strong>SBC IP-to-IP Routing Table</strong></td>
<td>Defines SBC IP-to-IP routing rules. The format of the ini file table parameter is as follows: [IP2IPRouting ] FORMAT IP2IPRouting_Index = IP2IPRouting_RouteName, IP2IPRouting_RoutingPolicyName, IP2IPRouting_SrcIPGroupName, IP2IPRouting_SrcUsernamePrefix, IP2IPRouting_SrcHost, IP2IPRouting_DestUsernamePrefix, IP2IPRouting_DestHost, IP2IPRouting_RequestType, IP2IPRouting_MessageConditionName, IP2IPRouting_ReRouteIPGroupName, IP2IPRouting_Trigger, IP2IPRouting_CallSetupRulesSetId, IP2IPRouting_DestType, IP2IPRouting_DestIPGroupName,</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP2IPRouting_DestSIPInterfaceName, IP2IPRouting_DestAddress, IP2IPRouting_DestPort, IP2IPRouting_DestTransportType, IP2IPRouting_AltRouteOptions, IP2IPRouting_GroupPolicy, IP2IPRouting_CostGroup, IP2IPRouting_DestTags, IP2IPRouting_SrcTags, IP2IPRouting_IPGroupSetName, IP2IPRouting_RoutingTagName, IP2IPRouting_InternalAction;</td>
</tr>
<tr>
<td>For more information, see &quot;Configuring SBC IP-to-IP Routing Rules&quot; on page 501.</td>
</tr>
</tbody>
</table>

#### IP Group Set Table

<table>
<thead>
<tr>
<th>IP Group Set</th>
<th>configure voip &gt; sbc routing ip-group-set [IPGroupSet]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defines IP Group Sets for call load-balancing. The format of the ini file table parameter is as follows: [IPGroupSet] FORMAT IPGroupSet_Index = IPGroupSet_Name, IPGroupSet_Policy, IPGroupSet_Tags; [IPGroupSet] For more information, see Configuring IP Group Sets on page 525.</td>
</tr>
</tbody>
</table>

#### IP Group Set Member Table

<table>
<thead>
<tr>
<th>IP Group Set Member</th>
<th>configure voip &gt; sbc routing ip-group-set-member [IPGroupSetMember]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defines IP Groups for IP Group Sets for call load-balancing. The format of the ini file table parameter is as follows: [IPGroupSetMember] FORMAT IPGroupSetMember_Index = IPGroupSetMember_IPGroupSetId, IPGroupSetMember_IPGroupSetMemberIndex, IPGroupSetMember_IPGroupName, IPGroupSetMember_Weight; [IPGroupSetMember] For more information, see Configuring IP Group Sets on page 525.</td>
</tr>
</tbody>
</table>

#### Alternative Routing Reasons Table

<table>
<thead>
<tr>
<th>Alternative Routing Reasons</th>
<th>configure voip &gt; sbc routing sbc-alternative-routing-reasons [SBCAlternativeRoutingReasons]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defines SBC alternative routing reason rules. The format of the ini file table parameter is as follows: [SBCAlternativeRoutingReasons] FORMAT SBCAlternativeRoutingReasons_Index = SBCAlternativeRoutingReasons_ReleaseCause; [SBCAlternativeRoutingReasons] For more information, see &quot;Configuring SIP Response Codes for Alternative Routing Reasons&quot; on page 519.</td>
</tr>
</tbody>
</table>

#### Inbound Manipulations Table

<table>
<thead>
<tr>
<th>Inbound Manipulations</th>
<th>configure voip &gt; sbc manipulation ip-inbound-manipulation [IPInboundManipulation]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Defines Inbound Manipulation rules. The format of the ini file table parameter is as follows: [IPInboundManipulation] FORMAT IPInboundManipulation_Index =</td>
</tr>
</tbody>
</table>
### Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPInboundManipulation_ManipulationName</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_IsAdditionalManipulation,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_ManipulatedURI,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_ManipulationPurpose,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_SrcIPGroupName,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_SrcUsernamePrefix,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_SrcHost,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_DestUsernamePrefix,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_DestHost,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_RequestType,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_RemoveFromLeft,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_RemoveFromRight,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_LeaveFromRight,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_Prefix2Add,</td>
<td></td>
</tr>
<tr>
<td>IPInboundManipulation_Suffix2Add;</td>
<td></td>
</tr>
</tbody>
</table>

### Outbound Manipulations Table

<table>
<thead>
<tr>
<th>Outbound Manipulations</th>
<th>Defines outbound manipulation rules.</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure voip &gt; sbc manipulation ip-</td>
<td>The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>outbound-manipulation</td>
<td>[IPOutboundManipulation] FORMAT IPOutboundManipulation_Index =</td>
</tr>
<tr>
<td>[IPOutboundManipulation]</td>
<td>IPOutboundManipulation_ManipulationName,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_RoutingPolicyName,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_IsAdditionalManipulation,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_SrcIPGroupName,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_DestIPGroupName,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_SrcUsernamePrefix,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_SrcHost,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_DestUsernamePrefix,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_DestHost,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_CallingNamePrefix,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_MessageConditionName,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_RequestType,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_ReRouteIPGroupName,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_Trigger,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_ManipulatedURI,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_RemoveFromLeft,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_RemoveFromRight,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_LeaveFromRight,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_Prefix2Add,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_Suffix2Add,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_PrivacyRestrictionMode,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_DestTags,</td>
</tr>
<tr>
<td></td>
<td>IPOutboundManipulation_SrcTags;</td>
</tr>
<tr>
<td></td>
<td>[IPOutboundManipulation]</td>
</tr>
</tbody>
</table>

### Routing Policies Table

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>configure voip &gt; sbc routing sbc-routing-</td>
<td></td>
</tr>
</tbody>
</table>
### Parameter | Description
--- | ---
| policy | The format of the ini file table parameter is as follows:  
\[SBCRoutingPolicy\]  
FORMAT SBCRoutingPolicy_Index = SBCRoutingPolicy_Name, SBCRoutingPolicy_LCREnable, SBCRoutingPolicy_LCRAverageCallLength, SBCRoutingPolicy_LCRDefaultCost, SBCRoutingPolicy_LdapServerGroupName;  
\[SBCRoutingPolicy\]  
For more information, see "Configuring SBC Routing Policy Rules" on page 521.

#### Dial Plan Table

| Dial Plan | configure voip > sbc dial-plan | Defines the name of the Dial Plan.  
The format of the ini file table parameter is as follows:  
\[DialPlan\]  
FORMAT DialPlan_Index = DialPlan_Name;  
\[DialPlan\]  
For more information, see "Configuring Dial Plans" on page 541.

| Dial Plan Rule | configure voip > sbc dial-plan-rule | Defines the dial plan rules per Dial Plan.  
For more information, see "Configuring Dial Plans" on page 541.  
**Note:**  
- The table is hidden in the ini file.  
- To configure Dial Plan rules from a file, see "Importing and Exporting Dial Plans" on page 546.

#### Malicious Signature Table

| Malicious Signature | configure voip > sbc malicious-signature-database | Defines the malicious signature patterns  
The format of the ini file table parameter is as follows:  
\[MaliciousSignatureDB\]  
FORMAT MaliciousSignatureDB_Index = MaliciousSignatureDB_Name, MaliciousSignatureDB_Pattern;  
\[MaliciousSignatureDB\]  
For more information, see "Configuring Malicious Signatures" on page 559.

### 65.12.1 Supplementary Services

The SBC and CRP supplementary services parameters are described in the table below.

**Table 65-40: SBC and CRP Supplementary Services Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emergency Call Preemption Parameters</td>
<td></td>
</tr>
</tbody>
</table>
### 65.13 IP Media Parameters

The IP media parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| IPMedia Detectors | Enables the device's DSP detectors for detection features such as AMD.  
- [0] Disable (default)  
- [1] Enable  
**Note:**  
- For the parameter to take effect, a device reset is required.  
- The DSP Detectors feature is available only if the device is installed with a License Key that includes this feature. For installing a License Key, see "License Key" on page 669. |
| Number of Media Channels | Defines the maximum number of DSP channels allocated for various functionalities such as transcoding, .  
The default is 0.  
**Note:**  
- For the parameter to take effect, a device reset is required.  
- The SBC application does not require DSP channels. The SBC application uses DSP channels only if media transcoding is needed, where two DSP channels are used per transcoding session. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Automatic Gain Control (AGC) Parameters</strong></td>
<td></td>
</tr>
</tbody>
</table>
| Enable AGC configure voip > media ipmedia > 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  
  For a description of AGC, see Automatic Gain Control (AGC) on page 203. |
| AGC Slope configure voip > media ipmedia > agc-gain-slope [AGCGainSlope] | Determines the AGC convergence rate:  
  - [0] 0 = 0.25 dB/sec  
  - [1] 1 = 0.50 dB/sec  
  - [2] 2 = 0.75 dB/sec  
  - [3] 3 = 1.00 dB/sec (default)  
  - [4] 4 = 1.25 dB/sec  
  - [5] 5 = 1.50 dB/sec  
  - [6] 6 = 1.75 dB/sec  
  - [7] 7 = 2.00 dB/sec  
  - [8] 8 = 2.50 dB/sec  
  - [9] 9 = 3.00 dB/sec  
  - [10] 10 = 3.50 dB/sec  
  - [11] 11 = 4.00 dB/sec  
  - [12] 12 = 4.50 dB/sec  
  - [13] 13 = 5.00 dB/sec  
  - [14] 14 = 5.50 dB/sec  
  - [15] 15 = 6.00 dB/sec  
  - [16] 16 = 7.00 dB/sec  
  - [17] 17 = 8.00 dB/sec  
  - [18] 18 = 9.00 dB/sec  
  - [19] 19 = 10.00 dB/sec  
  - [20] 20 = 11.00 dB/sec  
  - [21] 21 = 12.00 dB/sec  
  - [22] 22 = 13.00 dB/sec  
  - [23] 23 = 14.00 dB/sec  
  - [24] 24 = 15.00 dB/sec  
  - [25] 25 = 20.00 dB/sec  
  - [26] 26 = 25.00 dB/sec  
  - [27] 27 = 30.00 dB/sec  
  - [28] 28 = 35.00 dB/sec  
  - [29] 29 = 40.00 dB/sec  
  - [30] 30 = 50.00 dB/sec  
  - [31] 31 = 70.00 dB/sec |
| AGC Redirection configure voip > media ipmedia > 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. |
<p>| AGC Target Energy configure voip &gt; media ipmedia &gt; | Defines the signal energy value (dBm) that the AGC attempts to attain. |</p>
<table>
<thead>
<tr>
<th><strong>Parameter</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>agc-target-energy [AGCTargetEnergy]</td>
<td>The valid range is 0 to -63 dBm. The default is -19 dBm.</td>
</tr>
<tr>
<td>AGC Minimum Gain configure voip &gt; media ipmedia &gt; agc-min-gain [AGCMinGain]</td>
<td>Defines the minimum gain (in dB) by the AGC when activated. The range is 0 to -31. The default is -20. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>AGC Maximum Gain configure voip &gt; media ipmedia &gt; agc-max-gain [AGCMaxGain]</td>
<td>Defines the maximum gain (in dB) by the AGC when activated. The range is 0 to 18. The default is 15. <strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
</tbody>
</table>
| Disable Fast Adaptation configure voip > media ipmedia > agc-disable-fast-adaptation [AGCDisableFastAdaptation] | Enables the AGC Fast Adaptation mode.  
| | - [0] = Disable (default)  
| | - [1] = Enable  
| **Note:** For the parameter to take effect, a device reset is required. |

### Answering Machine Detector (AMD) Parameters

**For more information on AMD, see "Answering Machine Detection (AMD)" on page 200.**

| **Answer Machine Detector Sensitivity Parameter Suite** configure voip > media ipmedia > amd-sensitivity-parameter-suit [AMDSensitivityParameterSuit] | Global parameter that defines the AMD Parameter Suite to use. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDSensitivityParameterSuit). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| **Answer Machine Detector Sensitivity Level** configure voip > media ipmedia > amd-sensitivity-level [AMDSensitivityLevel] | Global parameter that defines the AMD detection sensitivity level of the selected AMD Parameter Suite. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDSensitivityLevel). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417. **Note:** If this functionality is configured for a specific IP Profile, the settings of this global parameter is ignored for calls associated with the IP Profile. |
| **AMD Sensitivity File** [AMDSensitivityFileName] | Defines the name of the AMD Sensitivity file that contains the AMD Parameter Suites.  
| **Note:**  
| - This file must be in binary format (.dat). You can use the DConvert utility to convert the original file format from XML to .dat.  
| - You can load this file using the Web interface (see "Loading Auxiliary Files" on page 655). |
| [AMDSensitivityFileUrl] | Defines the URL path to the AMD Sensitivity file for downloading from a remote server. |
| [AMDMinimumVoiceLength] | Defines the AMD minimum voice activity detection duration (in 5-ms units). Voice activity duration below this threshold is
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[AMDMaxGreetingTime]</td>
<td>Global parameter that defines the maximum duration that the device can take to detect a greeting message. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDMaxGreetingTime). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see &quot;Configuring IP Profiles&quot; on page 417. <strong>Note:</strong> 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>[AMDMaxPostGreetingSilenceTime]</td>
<td>Global parameter that defines the maximum duration of silence from after the greeting time is over (defined by AMDMaxGreetingTime) until the device's AMD decision. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AMDMaxPostSilenceGreetingTime). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see &quot;Configuring IP Profiles&quot; on page 417. <strong>Note:</strong> 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>[AMDTimeout]</td>
<td>Defines the timeout (in msec) between receiving Connect messages from the Tel side and sending AMD results. The valid range is 1 to 30,000. The default is 2,000 (i.e., 2 seconds).</td>
</tr>
<tr>
<td>AMD Beep Detection Mode</td>
<td>Determines the AMD beep detection mode. This mode detects the beeps played at the end of an answering machine message, by using the X-Detect header extension. The device sends a SIP INFO message containing the field values Type=AMD and SubType=Beep. This feature allows users of certain third-party, Application server to leave a voice message after an answering machine plays the &quot;beep&quot;.</td>
</tr>
<tr>
<td>- [0] Disabled (default)</td>
<td></td>
</tr>
<tr>
<td>- [1] Start After AMD</td>
<td></td>
</tr>
<tr>
<td>- [2] Start Immediately</td>
<td></td>
</tr>
<tr>
<td>Answer Machine Detector Beep</td>
<td>Defines the AMD beep detection timeout (i.e., the duration that the beep detector functions from when detection is initiated). This is used for detecting beeps at the end of an answering machine message. The valid value is in units of 100 milliseconds, from 0 to 1638. The default is 200 (i.e., 20 seconds).</td>
</tr>
<tr>
<td>Detection Timeout</td>
<td></td>
</tr>
<tr>
<td>Answer Machine Detector Beep</td>
<td>Defines the AMD beep detection sensitivity for detecting beeps at the end of an answering machine message. The valid value is 0 to 3, where 0 (default) is the least sensitive.</td>
</tr>
<tr>
<td>Detection Sensitivity</td>
<td></td>
</tr>
<tr>
<td>AMD Mode</td>
<td>Global parameter that enables the device to disconnect the IP-</td>
</tr>
</tbody>
</table>

User's Manual 938 Document #: LTRT-41877
### 65.14 Services

#### 65.14.1 SIP-based Media Recording Parameters

The SIP-based media recording parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| **SIP Recording Application**                  | Enables the SIP-based Media Recording feature:  
- **[0]** Disable (default)  
- **[1]** Enable  
**Note:** For the parameter to take effect, a device reset is required.  |
| **Recording Server (SRS) Destination Username** | Defines the SIP user part for the recording server. This user part is added in the SIP To header of the INVITE message that the device sends to the recording server.  
The valid value is a string of up to 50 characters. By default, no user part is defined. |
| **SIP Recording Rules**                         | Defines SIP Recording Routing rules (for siprec). The format of the ini file table parameter is as follows:  
[ SIPRecRouting ]  
FORMAT SIPRecRouting_Index =  
SIPRecRouting_RecordedIPGroupName,  
SIPRecRouting_RecordedSourcePrefix,  
SIPRecRouting_RecordedDestinationPrefix,  
SIPRecRouting_PeerIPGroupName,  
SIPRecRouting_PeerTrunkGroupID, SIPRecRouting_Caller,  
SIPRecRouting_SRSIPGroupName,  
SIPRecRouting_SRSRedundantIPGroupName;  
[ \SIPRecRouting ]  
For more information, see "Configuring SIP Recording Rules" on page 224. |

---

**configure voip > sip-definition settings > amd-mode [AMDmode]**  
to-Tel call upon detection of an answering machine on the Tel side. You can also configure this functionality per specific calls, using IP Profiles (IpProfile_AmdMode). For a detailed description of the parameter and for configuring this functionality in the IP Profiles table, see "Configuring IP Profiles" on page 417.  
**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.
65.14.2 RADIUS and LDAP Parameters

65.14.2.1 General Parameters

The general RADIUS and LDAP parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use Local Users Database</td>
<td>Defines when the device uses the Local Users table or an LDAP/RADIUS server for authenticating the login credentials (username-password) of users when logging into the device's management interface (e.g., Web or CLI).</td>
</tr>
<tr>
<td>[MgmtUseLocalUsersDatabase]</td>
<td>When No Auth Server Defined = (Default) The device authenticates the users using the Local Users table in the following scenarios: ✓ If no LDAP/RADIUS server is configured. ✓ If an LDAP/RADIUS server is configured, but connectivity with the server is down. If there is connectivity with the server, the device uses the server to authenticate the user. [0] Always = The device first attempts to authenticate the user using the Local Users table. If no user is found (based on the username-password combination), it attempts to authenticate the user using the LDAP/RADIUS server.</td>
</tr>
<tr>
<td>Behavior upon Authentication Server Timeout</td>
<td>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 Local Users table (local database). Note: The parameter is applicable to LDAP- and RADIUS-based management-user login authentication.</td>
</tr>
<tr>
<td>[MgmtBehaviorOnTimeout]</td>
<td>Default Access Level</td>
</tr>
<tr>
<td>[DefaultAccessLevel]</td>
<td></td>
</tr>
</tbody>
</table>

65.14.2.2 RADIUS Parameters

The RADIUS parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General RADIUS Parameters</td>
<td>Enables the RADIUS application. [0] Disable (Default)</td>
</tr>
<tr>
<td>Enable RADIUS Access Control</td>
<td></td>
</tr>
<tr>
<td>[RadiusAccessControl]</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| settings > enable [EnableRADIUS]              | • [1] Enable  
**Note:** For the parameter to take effect, a device reset is required.                                                             |
| [RadiusTrafficType]                           | Defines the device's network interface for communicating (RADIUS traffic) with the RADIUS server(s).                                        |
|                                               | • [0] OAMP (default)  
• [1] Control  
**Note:** If set to Control, only one Control interface must be configured in the IP Interfaces table; otherwise, RADIUS communication will fail. |
| RADIUS VSA Vendor ID configure system >       | Defines the vendor ID that the device accepts when parsing a RADIUS response packet.                                                        |
| radius settings > vsa-vendor-id [RadiusVSAVendorID] | The valid range is 0 to 0xFFFFFFFF. The default is 5003.                                                                                   |
| [MaxRADIUSSessions]                           | Defines the number of concurrent calls that can communicate with the RADIUS server (optional).                                               |
|                                               | The valid range is 0 to 240. The default is 240.                                                                                           |
| RADIUS Packets Retransmission [RADIUSRetransmission] | Defines the number of RADIUS retransmission retries when no response is received from the RADIUS server. See also the RadiusTo parameter.   |
|                                               | The valid range is 1 to 10. The default is 1.                                                                                              |
| RADIUS Response Time Out [RadiusTO]          | Defines the time interval (in seconds) that the device waits for a response before it performs a RADIUS retransmission. See also the RadiusRetransmission parameter. |
|                                               | The valid range is 1 to 30. The default is 2.                                                                                              |
| **RADIUS Accounting Parameters**             |                                                                                                                                              |
| RADIUS Accounting Type configure voip >       | Defines at what stage of the call that RADIUS accounting messages are sent to the RADIUS accounting server.                                 |
| sip-definition settings > radius-accounting   | • [0] At Call Release = (Default) Sent at call release only.                                                                                |
| AAA Indications configure system > cdr >      | Enables the Authentication, Authorization and Accounting (AAA) indications.                                                              |
| aaa-indications [AAAIndications]              | • [0] None = (Default) No indications.                                                                                                    |
|                                               | • [3] Accounting Only = Only accounting indications are used.                                                                            |
| **RADIUS User Authentication Parameters**     |                                                                                                                                              |
| Use RADIUS for Web/Telnet Login              | Enables RADIUS queries for Web and Telnet login authentication. When enabled, logging into the device's Web and Telnet embedded servers is done through a RADIUS server. The device communicates with a user-defined RADIUS server and verifies the given username and password against a remote database in a secure manner. |
| configure system > radius settings >         | • [0] Disable (default)  
• [1] Enable  
**Note:** For RADIUS login authentication to function, you must also configure the EnableRADIUS parameter to 1 (Enable). |
<p>| enable-mgmt-login [WebRADIUSLogin]           |                                                                                                                                              |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RADIUS authentication</td>
<td>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.</td>
</tr>
<tr>
<td>Password Local Cache Mode</td>
<td>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).</td>
</tr>
<tr>
<td>Absolute Expiry Timer</td>
<td>When you access a Web page, the timeout doesn't reset, instead it continues decreasing.</td>
</tr>
<tr>
<td>Reset Timer Upon Access</td>
<td>(Default) Upon each access to a Web page, the timeout always resets (reverts to the initial value configured by RadiusLocalCacheTimeout).</td>
</tr>
<tr>
<td>Password Local Cache Timeout</td>
<td>Defines the time (in seconds) the locally stored username and password (verified by the RADIUS server) are valid. When this time expires, the username and password become invalid and a must be re-verified with the RADIUS server. The valid range is 1 to 0xFFFFFFFF. The default is 300 (5 minutes).</td>
</tr>
<tr>
<td>RADIUS VSA Access Level Attribute</td>
<td>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.</td>
</tr>
<tr>
<td>RADIUS Servers Table</td>
<td>Defines RADIUS servers. The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>[RadiusServers]</td>
<td></td>
</tr>
<tr>
<td>FORMAT RadiusServers_Index = RadiusServers_ServerGroup, RadiusServers_IPAddress, RadiusServers_AuthenticationPort, RadiusServers_AccountingPort, RadiusServers_SharedSecret;</td>
<td></td>
</tr>
<tr>
<td>For a detailed description of this table, see &quot;Configuring RADIUS Servers&quot; on page 228.</td>
<td></td>
</tr>
</tbody>
</table>

### LDAP Parameters

The Lightweight Directory Access Protocol (LDAP) parameters are described in the table below.

#### Table 65-45: LDAP Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDAP Service</td>
<td>Enables the LDAP feature.</td>
</tr>
<tr>
<td>[0] Disable (default)</td>
<td></td>
</tr>
<tr>
<td>[1] Enable</td>
<td></td>
</tr>
</tbody>
</table>

---

**User's Manual 942**

**Document #: LTRT-41877**
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[LDAPServiceEnable]</td>
<td><strong>Note:</strong> For the parameter to take effect, a device reset is required.</td>
</tr>
<tr>
<td>LDAP Authentication Filter configure system &gt; ldap settings &gt; auth-filter [LDAPAuthFilter]</td>
<td>Defines the LDAP search filter attribute for searching the login username in the directory's subtree for LDAP-based user authentication and authorization. You can use the dollar ($) sign to represent the username. For example, if the parameter is set to &quot;(sAMAccountName=$)&quot; and the user logs in with the username &quot;SueM&quot;, the LDAP query is run for sAMAccountName=SueM.</td>
</tr>
</tbody>
</table>
| **Use LDAP for Web > Telnet Login** configure system > ldap settings > enable-mgmt-login [MgmtLDAPLogin] | Enables LDAP-based management-user login authentication and authorization.  
• [0] Disable (default)  
• [1] Enable  
**Note:** For the parameter to take effect, a device reset is required. |
<p>| [LDAPDebugMode] | Determines whether to enable the LDAP task debug messages. This is used for providing debug information regarding LDAP tasks. The valid value range is 0 to 3. The default is 0. |
| LDAP Numeric Attribute ldap-numeric-attr [LDAPNumericAttributes] | Defines up to five LDAP Attributes (separated by commas) for which the device employs LDAP query searches in the AD for numbers that may have characters between the digits. For more information, see &quot;Enabling LDAP Searches for Numbers with Characters&quot; on page 257. |
| MS LDAP OCS Number attribute name configure voip &gt; sip-definition settings &gt; ldap-ocs-nm-attr [MSLDAPAOCSSNumAttributeName] | Defines the name of the attribute that represents the user's Skype for Business number in the Microsoft AD database. The valid value is a string of up to 49 characters. The default is &quot;msRTCSIP-Line&quot;. |
| MS LDAP PBX Number attribute name configure voip &gt; sip-definition settings &gt; 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 &quot;telephoneNumber&quot;. |
| MS LDAP MOBILE Number attribute name configure voip &gt; sip-definition settings &gt; 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 &quot;mobile&quot;. |
| **configure voip &gt; sip-definition settings &gt; ldap-private-nm-attr [MSLDAPPPrivateNumAttributeName] | Defines the name of the attribute that represents the user's private number in the AD. If this value equals the value of MSLDAPPPrimaryKey or MSLDAPSecondaryKey parameter, then the device queries the AD for the destination number in this private attribute name; otherwise, the parameter is not used as a search key. The default is &quot;msRTCSIP-PrivateLine&quot;. |</p>
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MS LDAP DISPLAY Name Attribute Name</strong></td>
<td>Configure voip &gt; sip-definition settings &gt; ldap-display-nm-attr [MSLDAPDisplayAttributeName] 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 &quot;displayName&quot;.</td>
</tr>
<tr>
<td><strong>configure voip &gt; sip-definition settings &gt; ldap-primary-key [MSLDAPPrimaryKey]</strong></td>
<td>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 &quot;PBX&quot; attribute name (configured by the MSLDAPPBXNumAttributeName parameter). The default is not configured.</td>
</tr>
<tr>
<td><strong>configure voip &gt; sip-definition settings &gt; ldap-secondary-key [MSLDAPSecondaryKey]</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>LDAP Cache Service</strong></td>
<td>Configure system &gt; ldap settings &gt; ldap-cache-enable [LDAPCacheEnable] Enables the LDAP cache service.</td>
</tr>
<tr>
<td></td>
<td><code>- [0] Disable (default)</code></td>
</tr>
<tr>
<td></td>
<td><code>- [1] Enable</code></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong></td>
</tr>
<tr>
<td></td>
<td><code>- For the parameter to take effect, a device reset is required.</code></td>
</tr>
<tr>
<td></td>
<td>`- For more information on LDAP caching, see &quot;Configuring the Device's LDAP Cache&quot; on page 247.&quot;</td>
</tr>
</tbody>
</table>

**LDAP Servers Table**

<table>
<thead>
<tr>
<th>LDAP Servers</th>
<th>Configure system &gt; ldap ldap-configuration [LdapConfiguration] Defines LDAP servers. The format of the ini file table parameter is as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>[ LdapConfiguration ]</code></td>
</tr>
<tr>
<td></td>
<td><code>[ \LdapConfiguration ]</code></td>
</tr>
<tr>
<td></td>
<td>`For more information, see &quot;Configuring LDAP Servers&quot; on page 240.&quot;</td>
</tr>
</tbody>
</table>

**LDAP Server Search Base DN Table**

<table>
<thead>
<tr>
<th>LDAP Server Search Base DN Table</th>
<th>Configure system &gt; 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. The format of the ini file table parameter is as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>[ LdapServersSearchDNs ]</code></td>
</tr>
<tr>
<td></td>
<td><code>FORMAT LdapServersSearchDNs_Index = LdapServersSearchDNs_Base_Path, LdapServersSearchDNs_LdapConfigurationIndex,</code></td>
</tr>
</tbody>
</table>
### Parameter Reference

#### LdapServersSearchDNs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LdapServersSearchDNs</td>
<td>LdapServersSearchDNs[\LdapServersSearchDNs];</td>
</tr>
<tr>
<td>SearchDNs_InternalIndex</td>
<td>For more information, see &quot;Configuring LDAP DN (Base Paths) per LDAP Server&quot; on page 243.</td>
</tr>
</tbody>
</table>

#### Management LDAP Groups Table

**configure system > ldap mgmt-ldap-groups [MgmtLDAPGroups]**

<table>
<thead>
<tr>
<th>Management LDAP Groups Table</th>
<th>Defines the users group attribute in the AD and corresponding management access level.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANAGEMENT_LDAP_GROUPS</td>
<td>The format of the ini file table parameter is as follows:</td>
</tr>
<tr>
<td>FORMAT MgmtLDAPGroups_Index =</td>
<td>MgmtLDAPGroups_LdapConfigurationIndex, MgmtLDAPGroups_GroupIndex, MgmtLDAPGroups_Level, MgmtLDAPGroups_Group;</td>
</tr>
<tr>
<td>MgmtLDAPGroups_Group</td>
<td>For more information, see &quot;Configuring Access Level per Management Groups Attributes&quot; on page 245.</td>
</tr>
</tbody>
</table>

#### LDAP Server Groups Table

**configure system > ldap ldap-server-groups [LDAPServerGroups]**

<table>
<thead>
<tr>
<th>LDAP Server Groups Table</th>
<th>Defines LDAP Server Groups.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FORMAT LdapServerGroups_Index =</td>
<td>LdapServerGroups_Name, LdapServerGroups_ServerType, LdapServerGroups_SearchMethod,</td>
</tr>
<tr>
<td>LdapServerGroups_CacheEntryTimeout</td>
<td>LdapServerGroups_CacheEntryRemovalTimeout, LdapServerGroups_CacheDnsMethod;</td>
</tr>
<tr>
<td>LdapServerGroups_SearchDnsMethod</td>
<td>For more information, see &quot;Configuring LDAP Server Groups&quot; on page 238.</td>
</tr>
</tbody>
</table>

---

**65.14.3 Least Cost Routing Parameters**

The Least Cost Routing (LCR) parameters are described in the table below.

#### Table 65-46: LCR Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost Groups Table</td>
<td>Defines the Cost Groups for LCR, where each Cost Group is configured with a name, fixed call connection charge, and a call rate (charge per minute). [CostGroupTable]</td>
</tr>
<tr>
<td>[CostGroupTable]</td>
<td>For example: CostGroupTable 2 = &quot;Local Calls&quot;, 2, 1; For more information, see &quot;Configuring Cost Groups&quot; on page 265.</td>
</tr>
</tbody>
</table>
### 65.14.4 Call Setup Rules Parameters

The Call Setup Rules parameters are described in the table below.

#### Table 65-47: Call Setup Rules Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call Setup Rules</td>
<td>Defines Call Setup Rules that the device runs at call setup for LDAP-based routing and other advanced routing logic requirements including manipulation.</td>
</tr>
<tr>
<td>configure voip &gt; message</td>
<td></td>
</tr>
<tr>
<td>call-setup-rules</td>
<td></td>
</tr>
<tr>
<td>[CallSetupRules]</td>
<td></td>
</tr>
</tbody>
</table>

FORMAT CallSetupRules_Index = CallSetupRules_RulesSetID, CallSetupRules_QueryType, CallSetupRules_QueryTarget, CallSetupRules_AttributesToQuery, CallSetupRules_AttributesToGet, CallSetupRules_RowRole, CallSetupRules_Condition, CallSetupRules_ActionSubject, CallSetupRules_ActionType, CallSetupRules_ActionValue;

[CallSetupRules]

For more information, see "Configuring Call Setup Rules" on page 405.

### 65.14.5 HTTP-based Services

The HTTP-based service parameters are described in the table below.

#### Table 65-48: HTTP-based Service Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quality Status</td>
<td>Enables QoS-based routing by the routing server. The device collects QoS metrics (media and signaling) and sends them to the routing server.</td>
</tr>
<tr>
<td>configure system &gt; http-</td>
<td></td>
</tr>
<tr>
<td>services &gt; routing-qos-</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td></td>
</tr>
<tr>
<td>[RoutingServerQualityStatus]</td>
<td></td>
</tr>
</tbody>
</table>

- [0] Disable (default)
- [1] Enable

For more information, see Configuring QoS-Based Routing by Routing Server on page 278.

| Quality Status Rate           | Defines the rate (in sec) at which the device sends QoS reports to the routing server. |
| configure system > http-      | The valid range is 15-3600. The default is 60. |
| services > routing-qos-       | For more information, see Configuring QoS-Based Routing by Routing Server on page 278. |
| status-rate                   |             |
| [RoutingServerQualityStatusRate] |             |

For more information, see Configuring QoS-Based Routing by Routing Server on page 278.
### 65. Configuration Parameters Reference

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topology Status</td>
<td>Enables the reporting of the device's topology status (using the REST TopologyStatus API command) to HTTP remote hosts.</td>
</tr>
</tbody>
</table>
| ```configure system > http-services > routing- server-group-status [RoutingServerGroupStatus]``` | - [0] Disable (default)  
  - [1] Enable  
  For more information, see "Configuring HTTP Services" on page 268.                                              |

### Remote Web Services Table

| Remote Web Services | Defines remote Web services.  
| configure system > http-services > http-remote-services [HTTPRemoteServices] | The format of the ini file table parameter is as follows:  
| For more information, see "Configuring Remote Web Services" on page 268. |

### HTTP Remote Hosts Table

| HTTP Remote Hosts | Defines remote HTTP hosts per remote Web service.  
| configure system > http-services > http-remote-hosts [HTTPRemoteHosts] | The format of the ini file table parameter is as follows:  
| [HTTPRemoteHosts] | FORMAT HTTPRemoteHosts_Index = HTTPRemoteHosts_HTTPRemoteServiceIndex, HTTPRemoteHosts_RemoteHostIndex, HTTPRemoteHosts_Name, HTTPRemoteHosts_Address, HTTPRemoteHosts_Port, HTTPRemoteHosts.Interface, HTTPRemoteHosts_HTTPTransportType, HTTPRemoteHosts_HostStatus; [HTTPRemoteHosts]  
| For more information, see "Configuring Remote HTTP Hosts" on page 273. |

### 65.14.6 HTTP Proxy Parameters

The HTTP Proxy service parameters are described in the table below.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| HTTP Proxy Application | Enables the HTTP Proxy application.  
  - [0] Disable (default)  
  - [1] Enable  
  **Note**: For the parameter to take effect, a device reset is required. |
| HTTP Proxy Debug Level | Enables debugging of HTTP services.  
  - [0] No Debug = (Default) Disable.  
  - [1] Basic = Enables debug and sends basic information in Syslog.  

### HTTP Interfaces Table

| HTTP Interfaces | Defines local listening interfaces for receiving HTTP/S requests from Web clients for HTTP/S-based services.  
  The format of the ini file table parameter is as follows:  
  ```ini  
  [HTTPInterface]  
  FORMAT HTTPInterface_Index = HTTPInterface_InterfaceName, HTTPInterface_NetworkInterface, HTTPInterface_Protocol, HTTPInterface_Port, HTTPInterface_TLSContext, HTTPInterface_VerifyCert;  
  [/HTTPInterface]  
  ```  
  For more information, see "Configuring HTTP Interfaces" on page 281. |

### HTTP Proxy Services Table

| HTTP Proxy Services | Defines HTTP Proxy based services.  
  The format of the ini file table parameter is as follows:  
  ```ini  
  [HTTPProxyService]  
  FORMAT HTTPProxyService_Index = HTTPProxyService_ServiceName, HTTPProxyService_ListeningInterface, HTTPProxyService_URLPrefix, HTTPProxyService_KeepAliveMode;  
  [/HTTPProxyService]  
  ```  
  For more information, see "Configuring HTTP Proxy Services" on page 283. |

### HTTP Proxy Hosts Table

| HTTP Proxy Hosts | Defines HTTP Proxy hosts. The table is a "child" of the HTTP Proxy Services table (HTTPProxyService). An HTTP Proxy Host represents the HTTP-based managed equipment (e.g., IP Phone).  
  The format of the ini file table parameter is as follows:  
  ```ini  
  [HTTPProxyHost]  
  FORMAT HTTPProxyHost_Index = HTTPProxyHost_HTTPProxyServiceId, HTTPProxyHost_HTTPProxyHostId, HTTPProxyHost_NetworkInterface, HTTPProxyHost_IpAddress, HTTPProxyHost_Protocol, HTTPProxyHost_Port, HTTPProxyHost_TLSContext, HTTPProxyHost_VerifyCert;  
  [/HTTPProxyHost]  
  ```  
  For more information, see "Configuring HTTP Proxy Hosts" on page 284. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>284.</td>
<td></td>
</tr>
</tbody>
</table>

**EMS Services Table**

<table>
<thead>
<tr>
<th>EMS Services configure network &gt; http-proxy ems-serv [EMSService]</th>
<th>Defines an HTTP-based EMS Service so that the device can act as an HTTP Proxy that enables AudioCodes EMS to manage AudioCodes equipment (such as IP Phones) over HTTP when the equipment is located behind NAT (e.g., in the LAN) and EMS is located in a public domain (e.g., in the WAN). The format of the ini file table parameter is as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ EMSService ]</td>
<td>FORMAT EMSService_Index = EMSService_ServiceName, EMSService_PrimaryServer, EMSService_SecondaryServer, EMSService_DeviceLoginInterface, EMSService_EMSInterface; [ ]EMSService ] For more information, see &quot;Configuring an HTTP-based EMS Service&quot; on page 286.</td>
</tr>
</tbody>
</table>
This page is intentionally left blank.
## 66 Channel Capacity

The following table lists maximum concurrent SIP signaling sessions, concurrent media sessions, and registered users.

### Table 66-1: Maximum Signaling, Media Sessions and Registered Users

<table>
<thead>
<tr>
<th>Product</th>
<th>Max. Signaling Sessions</th>
<th>Max. Media Sessions</th>
<th>Codec Transcoding</th>
<th>Max. Registered Users</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>RTP-RTP</td>
<td>SRTP-RTP</td>
<td></td>
</tr>
<tr>
<td>Mediant SE SBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DL320e G8 4-cores 3.1 GHz 16 GB RAM</td>
<td>15,000</td>
<td>10,000</td>
<td>6,500</td>
<td>N/A</td>
</tr>
<tr>
<td>DL360p G8 20-cores 2.8 GHz 64 GB RAM - or - DL360 G9 8-cores 2.6 GHz 32 GB RAM</td>
<td>24,000</td>
<td>16,000</td>
<td>14,000</td>
<td>N/A</td>
</tr>
<tr>
<td>Mediant VE SBC</td>
<td>1 vCPU, 2 GB RAM</td>
<td>250</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>VMware</td>
<td>1/2/4 vCPU, 8 GB RAM</td>
<td>3,000</td>
<td>3,000</td>
<td>2,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2 vCPU: 2-vCPU Mediant VE SBC on page 954</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 vCPU: 4-vCPU Mediant VE SBC on page 955</td>
</tr>
<tr>
<td></td>
<td>4/8 vCPU 16 GB RAM</td>
<td>9,000</td>
<td>6,000</td>
<td>5,000</td>
</tr>
<tr>
<td></td>
<td>1 vCPU 2 GB RAM</td>
<td>250</td>
<td>250</td>
<td>250</td>
</tr>
<tr>
<td>Mediant VE SBC</td>
<td>1/2/4 vCPU 4 GB RAM</td>
<td>1,800</td>
<td>1,800</td>
<td>1,400</td>
</tr>
<tr>
<td>OpenStack KVM</td>
<td></td>
<td></td>
<td></td>
<td>2 vCPU: 2-vCPU Mediant VE SBC on page 954</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4 vCPU: 4-vCPU Mediant VE SBC on page 955</td>
</tr>
<tr>
<td></td>
<td>4/8 vCPU 16 GB RAM</td>
<td>4,000</td>
<td>2,700</td>
<td>2,700</td>
</tr>
<tr>
<td></td>
<td>8 vCPU 16 GB RAM SR-IOV Intel NICs</td>
<td>24,000</td>
<td>24,000</td>
<td>10,000</td>
</tr>
<tr>
<td></td>
<td>8 vCPU 32 GB RAM SR-IOV Intel NICs</td>
<td>24,000</td>
<td>24,000</td>
<td>10,000</td>
</tr>
<tr>
<td>Product</td>
<td>Max. Signaling Sessions</td>
<td>Max. Media Sessions</td>
<td>Codec Transcoding</td>
<td>Max. Registered Users</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------</td>
<td>---------------------</td>
<td>-------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RTP-RTP</td>
<td>SRTP-RTP</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hyper-V</td>
<td>1 vCPU 2 GB RAM</td>
<td>250</td>
<td>250</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>1/2/4 vCPU 4 GB RAM</td>
<td>900</td>
<td>600</td>
<td>600</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AWS / EC2</td>
<td>c4.2xlarge</td>
<td>2,000</td>
<td>2,000</td>
<td>2,000</td>
</tr>
<tr>
<td>Mediant VE SBC with Media Transcoders</td>
<td>8 vCPU 64 GB RAM SR-IOV Intel NICs</td>
<td>24,000</td>
<td>24,000</td>
<td>12,000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**
- The figures listed in the table are accurate at the time of publication of this document. However, these figures may change due to a later software update. For the latest figures, please contact your AudioCodes sales representative.
- Installation and use of voice coders is subject to obtaining the appropriate license and royalty payments.
- **Registered Users** is the maximum number of users that can be registered with the device. This applies to the supported application (SBC or CRP).
- The capacity figures for Mediant VE are for running on the recommended platforms only, when there are no other virtual machines (VM) running on these platforms.
- 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.
  - A gateway session (i.e. TDM-RTP or TDM-SRTP) is also considered as a media session for the calculation of media sessions. In other words, the maximum Media Sessions specified in the table refer to the sum of Gateway and SBC sessions.
  - In case of direct media (i.e., Anti-tromboning / Non-Media Anchoring), where only SIP signaling traverses the SBC and media flows directly between the SIP entities, only a signaling session resource is used. Thus, for products with a greater signaling session capacity than media, even when media session resources have been exhausted, additional signaling sessions can still be handled for direct-media calls.
√ For call sessions requiring transcoding, one transcoding session resource is also used. For example, for a non-direct media call in which one leg uses G.711 and the other leg G.729, one signaling resource, one media session resource, and one transcoding session resource is used.

• Mediant VE SBC with Media Transcoders: The following limitations exist:
  √ * Each transcoding session is weighted as two RTP-RTP sessions without transcoding. Therefore, the number of sessions without transcoding plus the doubled number of sessions with transcoding must be less than the maximum RTP-RTP figure specified in the table. As result, if all sessions are with transcoding, the maximum number of sessions is half the maximum RTP-RTP sessions without transcoding specified in the table.
  √ ** The maximum SRTP-RTP sessions is also effected by the above limitations. For example, if all sessions are using transcoding, the maximum number of SRTP-RTP sessions is also limited by half of the maximum RTP-RTP sessions without transcoding.

• Mediant VE SBC with vMT-type Media Transcoder: The host running the vMT virtual machine requires the following configuration:
  √ At least 2.8 GHz CPU with Intel® AVX support
  √ SR-IOV enabled NICs
  √ KVM environment
  √ 8 hyper-threaded vCPUs should be allocated to the vMT virtual machine (4 physical cores)
  √ 4-GB RAM should be allocated to the vMT virtual machine

• Mediant VE SBC and vMT-type Media Transcoder: Codec-transcoding functionality is supported only on Intel CPUs with AVX enhancement. In addition, AVX support must be reflected on the vCPU of the SBC virtual machine.

• Mediant VE SBC with Media Transcoder Cluster is currently supported only on the OpenStack-KVM hypervisor.
66.1 Mediant VE SBC

The maximum number of supported SBC sessions is listed in "Channel Capacity" on page 951. These SBC sessions also support SRTP and RTCP XR. When DSP capabilities are required, the number of sessions that can use DSP capabilities is reduced, as shown in the tables in this section.

66.1.1 Mediant VE SBC for OpenStack and VMware Hypervisors

The following tables list maximum channel capacity for Mediant VE SBC 2.8 GHz running on OpenStack-KVM or VMware hypervisors.

66.1.1.1 2-vCPU Mediant VE SBC

The following table lists maximum channel capacity for the 2-vCPU (1 vCPU reserved for DSP) Mediant VE SBC.

Table 66-2: Channel Capacity for 2-vCPU Mediant VE SBC on OpenStack-KVM /VMware

<table>
<thead>
<tr>
<th>Session Coders</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Coder Profile</td>
<td>To Coder</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 1</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 1</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-NB / G.722</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-NB / G.722</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-WB (G.722.2)</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-WB (G.722.2)</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-NB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-NB</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-WB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-WB</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-NB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-NB</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-WB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-WB</td>
</tr>
</tbody>
</table>
Note:

- **Profile 1**: G.711 at 20ms only, without T.38 support.
- **Profile 2**: G.711, G.726, G.729, G.723.1, T.38.
- **Basic**: excludes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- **Extended**: includes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- Acoustic Echo Suppressor may reduce capacity. For more information, contact your AudioCodes sales representative.

The table below lists the maximum number of concurrent sessions per detection feature. The figures are based on the following assumptions:

- Timeout for fax detection is 10 seconds (default)
- Call duration is 100 seconds
- Fax detection is required on both legs of the SBC call
- Answer Detection (AD), Answering Machine Detection (AMD), Beep Detection, and Call Progress Tone Detection (CP) is only on one leg of the SBC call (should this not be the case, figures will be reduced)
- Sessions are only for forwarding sessions (i.e., no transcoding)

### Table 66-3: Channel Capacity per Detection Feature for 2-vCPU Mediant VE SBC on OpenStack-KVM/VMware

<table>
<thead>
<tr>
<th>Special Detection Features</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax Detection</td>
<td>2,400</td>
</tr>
<tr>
<td>AD/AMD/Beep Detection</td>
<td>2,400</td>
</tr>
<tr>
<td>CP Detection</td>
<td>2,400</td>
</tr>
<tr>
<td>Jitter Buffer</td>
<td>200</td>
</tr>
</tbody>
</table>

### 66.1.1.2 4-vCPU Mediant VE SBC

The following table lists maximum channel capacity for the 4-vCPU (3 vCPUs reserved for DSP) Mediant VE SBC.

### Table 66-4: Channel Capacity for 4-vCPU Mediant VE SBC on OpenStack-KVM/VMware

<table>
<thead>
<tr>
<th>From Coder Profile</th>
<th>To Coder</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Extended</td>
<td>Basic</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 1</td>
<td>750</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 1</td>
<td>375</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2</td>
<td>250</td>
</tr>
<tr>
<td>Profile 1 + AMR-NB / G.722</td>
<td>375</td>
<td>450</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>250</td>
</tr>
</tbody>
</table>
## Session Coders and Number of Sessions

<table>
<thead>
<tr>
<th>From Coder</th>
<th>To Coder</th>
<th>Extended</th>
<th>Basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-WB</td>
<td>125</td>
<td>150</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-WB</td>
<td>100</td>
<td>125</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-NB</td>
<td>350</td>
<td>375</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-NB</td>
<td>225</td>
<td>250</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-WB</td>
<td>225</td>
<td>225</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-WB</td>
<td>175</td>
<td>175</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-NB</td>
<td>225</td>
<td>250</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-NB</td>
<td>175</td>
<td>175</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-WB</td>
<td>175</td>
<td>175</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-WB</td>
<td>125</td>
<td>150</td>
</tr>
</tbody>
</table>

### Note:

- **Profile 1**: G.711 at 20ms only, without T.38 support.
- **Profile 2**: G.711, G.726, G.729, G.723.1, T.38.
- **Basic**: excludes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- **Extended**: includes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection
- Acoustic Echo Suppressor may reduce capacity. For more information, contact your AudioCodes sales representative.

The table below lists the maximum number of concurrent sessions per detection feature. The figures are based on the following assumptions:

- Timeout for fax detection is 10 seconds (default)
- Call duration is 100 seconds
- Fax detection is required on both legs of the SBC call
- Answer Detection (AD), Answering Machine Detection (AMD), Beep Detection, and Call Progress Tone Detection (CP) is only on one leg of the SBC call (should this not be the case, figures will be reduced)
- Sessions are only for forwarding sessions (i.e., no transcoding)

### Table 66-5: Channel Capacity per Detection Feature for 4-vCPU Mediant VE SBC on OpenStack

<table>
<thead>
<tr>
<th>Special Detection Features</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax Detection</td>
<td>7,200</td>
</tr>
<tr>
<td>AD/AMD/Beep Detection</td>
<td>7,200</td>
</tr>
<tr>
<td>CP Detection</td>
<td>7,200</td>
</tr>
<tr>
<td>Jitter Buffer</td>
<td>650</td>
</tr>
</tbody>
</table>
66.1.1.3 8-vCPU Mediant VE SBC

The following table lists maximum channel capacity for the 8-vCPU (4 vCPUs reserved for DSP) Mediant VE SBC.

Table 66-6: Channel Capacity for 8-vCPU Mediant VE SBC on OpenStack-KVM/VMware

<table>
<thead>
<tr>
<th>From Coder Profile</th>
<th>To Coder Profile</th>
<th>Extended</th>
<th>Basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile 1</td>
<td>Profile 1</td>
<td>1,000</td>
<td>1,200</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 1</td>
<td>525</td>
<td>600</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2</td>
<td>350</td>
<td>400</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>525</td>
<td>600</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>350</td>
<td>400</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-WB</td>
<td>175</td>
<td>200</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-WB</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-NB</td>
<td>475</td>
<td>500</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-NB</td>
<td>325</td>
<td>350</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-WB</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-WB</td>
<td>225</td>
<td>250</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-NB</td>
<td>300</td>
<td>325</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-NB</td>
<td>225</td>
<td>250</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-WB</td>
<td>175</td>
<td>200</td>
</tr>
</tbody>
</table>

Note:
- **Profile 1**: G.711 at 20ms only, without T.38 support.
- **Profile 2**: G.711, G.726, G.729, G.723.1, T.38.
- **Basic**: excludes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- **Extended**: includes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- Acoustic Echo Suppressor may reduce capacity. For more information, contact your AudioCodes sales representative.

The table below lists the maximum number of concurrent sessions per detection feature. The figures are based on the following assumptions:
- Timeout for fax detection is 10 seconds (default)
- Call duration is 100 seconds
Fax detection is required on both legs of the SBC call

Answer Detection (AD), Answering Machine Detection (AMD), Beep Detection, and Call Progress Tone Detection (CP) is only on one leg of the SBC call (should this not be the case, figures will be reduced)

Sessions are only for forwarding sessions (i.e., no transcoding)

Table 66-7: Channel Capacity per Detection Feature for 8-vCPU Mediant VE SBC on OpenStack-KVM/VMware

<table>
<thead>
<tr>
<th>Special Detection Features</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax Detection</td>
<td>9,600</td>
</tr>
<tr>
<td>AD/AMD/Beep Detection</td>
<td>9,600</td>
</tr>
<tr>
<td>CP Detection</td>
<td>9,600</td>
</tr>
<tr>
<td>Jitter Buffer</td>
<td>875</td>
</tr>
</tbody>
</table>

66.1.2 Amazon AWS EC2

The following tables list maximum channel capacity for Mediant VE SBC on the Amazon EC2 platform.

Table 66-8: Channel Capacity for Mediant VE SBC on c4.2xlarge

<table>
<thead>
<tr>
<th>Session Coders</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Coder Profile</td>
<td>To Coder</td>
</tr>
<tr>
<td>1</td>
<td>Profile 1</td>
</tr>
<tr>
<td>2</td>
<td>Profile 1</td>
</tr>
<tr>
<td>2</td>
<td>Profile 2</td>
</tr>
<tr>
<td>1</td>
<td>Profile 2 + AMR-NB / G.722</td>
</tr>
<tr>
<td>2</td>
<td>Profile 2 + AMR-NB / G.722</td>
</tr>
<tr>
<td>1</td>
<td>Profile 2 + AMR-WB</td>
</tr>
<tr>
<td>2</td>
<td>Profile 2 + AMR-WB</td>
</tr>
<tr>
<td>1</td>
<td>Profile 2 + SILK-NB</td>
</tr>
<tr>
<td>2</td>
<td>Profile 2 + SILK-NB</td>
</tr>
<tr>
<td>1</td>
<td>Profile 2 + SILK-WB</td>
</tr>
<tr>
<td>2</td>
<td>Profile 2 + SILK-WB</td>
</tr>
<tr>
<td>1</td>
<td>Profile 2 + Opus-NB</td>
</tr>
<tr>
<td>2</td>
<td>Profile 2 + Opus-NB</td>
</tr>
<tr>
<td>1</td>
<td>Profile 2 + Opus-WB</td>
</tr>
<tr>
<td>2</td>
<td>Profile 2 + Opus-WB</td>
</tr>
</tbody>
</table>
### 66.1.3 Mediant VE SBC for Hyper-V Hypervisor

The following tables lists maximum channel capacity for Mediant VE SBC 2.1 GHz running on Hyper-V hypervisor.

#### 66.1.3.1 2-vCPU Mediant VE SBC

The following table lists maximum channel capacity for the 2-vCPU (1 vCPU reserved for DSP) Mediant VE SBC.

<table>
<thead>
<tr>
<th>Session Coders</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Coder Profile</td>
<td>To Coder</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 1</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 1</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2</td>
</tr>
<tr>
<td>From Coder</td>
<td>To Coder</td>
</tr>
<tr>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-NB / G.722</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-NB / G.722</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-WB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-WB</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-NB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-NB</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-WB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-WB</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-NB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-NB</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-WB</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-WB</td>
</tr>
</tbody>
</table>

**Note:**
- *Profile 1*: G.711 at 20ms only, without T.38 support.
- *Basic*: excludes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- *Extended*: includes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- Acoustic Echo Suppressor may reduce capacity. For more information, contact your AudioCodes sales representative.

The table below lists the maximum number of concurrent sessions per detection feature. The figures are based on the following assumptions:
- Timeout for fax detection is 10 seconds (default)
- Call duration is 100 seconds
- Fax detection is required on both legs of the SBC call
- Answer Detection (AD), Answering Machine Detection (AMD), Beep Detection, and Call Progress Tone Detection (CP) is only on one leg of the SBC call (should this not be the case, figures will be reduced)
- Sessions are only for forwarding sessions (i.e., no transcoding)

**Table 66-11: Channel Capacity per Detection Feature for 2-vCPU Mediant VE SBC on Hyper-V**

<table>
<thead>
<tr>
<th>Special Detection Features</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax Detection</td>
<td>1,800</td>
</tr>
<tr>
<td>AD/AMD/Beep Detection</td>
<td>1,800</td>
</tr>
<tr>
<td>CP Detection</td>
<td>1,800</td>
</tr>
<tr>
<td>Jitter Buffer</td>
<td>150</td>
</tr>
</tbody>
</table>
66.1.3.2 4-vCPU Mediant VE SBC

The following table lists maximum channel capacity for the 4-vCPU (3 vCPUs reserved for DSP) Mediant VE SBC.

**Table 66-12: Channel Capacity for 4-vCPU Mediant VE SBC on Hyper-V**

<table>
<thead>
<tr>
<th>From Coder Profile</th>
<th>To Coder</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Extended</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 1</td>
<td>550</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 1</td>
<td>300</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2</td>
<td>200</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>275</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>200</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-WB</td>
<td>100</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-WB</td>
<td>75</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-NB</td>
<td>250</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-NB</td>
<td>175</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-WB</td>
<td>150</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-WB</td>
<td>125</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-NB</td>
<td>150</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-NB</td>
<td>125</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-WB</td>
<td>125</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-WB</td>
<td>100</td>
</tr>
</tbody>
</table>

**Note:**
- **Profile 1:** G.711 at 20ms only, without T.38 support.
- **Profile 2:** G.711, G.726, G.729, G.723.1, T.38.
- **Basic:** excludes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- **Extended:** includes in-band signaling (in voice channel), VAD, Silence Suppression and fax detection.
- Acoustic Echo Suppressor may reduce capacity. For more information, contact your AudioCodes sales representative.

The table below lists the maximum number of concurrent sessions per detection feature. The figures are based on the following assumptions:

- Timeout for fax detection is 10 seconds (default)
- Call duration is 100 seconds
- Fax detection is required on both legs of the SBC call
- Answer Detection (AD), Answering Machine Detection (AMD), Beep Detection, and
Call Progress Tone Detection (CP) is only on one leg of the SBC call (should this not be the case, figures will be reduced)

- Sessions are only for forwarding sessions (i.e., no transcoding)

### Table 66-13: Channel Capacity per Detection Feature for 4-vCPU Mediant VE SBC on Hyper-V

<table>
<thead>
<tr>
<th>Special Detection Features</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fax Detection</td>
<td>5,400</td>
</tr>
<tr>
<td>AD/AMD/Beep Detection</td>
<td>5,400</td>
</tr>
<tr>
<td>CP Detection</td>
<td>5,400</td>
</tr>
<tr>
<td>Jitter Buffer</td>
<td>500</td>
</tr>
</tbody>
</table>

#### 66.1.4 Mediant VE SBC with Media Transcoders

Mediant VE SBC with Virtual Media Transcoders allows increasing the number of transcoding sessions by using Media Transcoders. The maximum number of transcoding sessions depends on the following:

- The number of Media Transcoders in the media transcoding cluster.
- The cluster operation mode (Best-Effort or Full-HA mode).
- The maximum transcoding sessions that the Mediant VE SBC is capable of performing. Each transcoding session is weighted as two RTP-RTP sessions without transcoding. Therefore, the number of sessions without transcoding plus the doubled number of sessions with transcoding must be less than the maximum RTP-RTP value specified in the table. As a result, if all sessions are with transcoding, the maximum number of sessions is half the maximum RTP-RTP sessions without transcoding as specified in Table 66-1.

The following table lists maximum transcoding sessions capacity of a single Media Transcoder of MT Type:

### Table 66-14: Transcoding Capacity per Profile for a Single MT

<table>
<thead>
<tr>
<th>From Coder Profile</th>
<th>To Coder</th>
<th>1 x MPM12B</th>
<th>2 x MPM12B</th>
<th>3 x MPM12B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile 1</td>
<td>Profile 1</td>
<td>2875</td>
<td>5000</td>
<td>5000</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 1</td>
<td>2300</td>
<td>4025</td>
<td>5000</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2</td>
<td>1800</td>
<td>3175</td>
<td>4550</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>2000</td>
<td>3525</td>
<td>5000</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>1625</td>
<td>2850</td>
<td>4075</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-WB (G.722.2)</td>
<td>1425</td>
<td>2175</td>
<td>3600</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-WB (G.722.2)</td>
<td>1225</td>
<td>2175</td>
<td>3100</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-NB</td>
<td>1425</td>
<td>2500</td>
<td>3600</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-NB</td>
<td>1225</td>
<td>2175</td>
<td>3100</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-WB</td>
<td>850</td>
<td>1500</td>
<td>2150</td>
</tr>
</tbody>
</table>
The following table lists maximum transcoding sessions capacity of a single Media Transcoder of vMT Type:

### Table 66-15: Transcoding Capacity per Profile for a Single vMT

<table>
<thead>
<tr>
<th>From Coder Profile</th>
<th>To Coder</th>
<th>Number of Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1 x MPM12B</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-WB</td>
<td>850</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-NB</td>
<td>1050</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-NB</td>
<td>950</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>From Coder Profile</th>
<th>To Coder</th>
<th>Extended</th>
<th>Basic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile 1</td>
<td>Profile 1</td>
<td>1225</td>
<td>1600</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 1</td>
<td>650</td>
<td>775</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2</td>
<td>425</td>
<td>525</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>500</td>
<td>575</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-NB / G.722</td>
<td>350</td>
<td>425</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + AMR-WB</td>
<td>175</td>
<td>175</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + AMR-WB</td>
<td>150</td>
<td>150</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-NB</td>
<td>450</td>
<td>500</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-NB</td>
<td>325</td>
<td>375</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + SILK-WB</td>
<td>275</td>
<td>300</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + SILK-WB</td>
<td>225</td>
<td>250</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-NB</td>
<td>275</td>
<td>300</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-NB</td>
<td>225</td>
<td>250</td>
</tr>
<tr>
<td>Profile 1</td>
<td>Profile 2 + Opus-WB</td>
<td>200</td>
<td>225</td>
</tr>
<tr>
<td>Profile 2</td>
<td>Profile 2 + Opus-WB</td>
<td>175</td>
<td>200</td>
</tr>
</tbody>
</table>
66.2 Mediant SE SBC

Note: Mediant SE SBC does not implement digital signal processing (DSP). Therefore, it supports only SBC functionalities that do not require media signal processing.
## 67 Technical Specifications

The device's technical specifications are listed in the table below.

**Note:**
- 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>
<thead>
<tr>
<th>Function</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Security</strong></td>
<td></td>
</tr>
<tr>
<td>Access Control</td>
<td>DoS/DDoS line rate protection, bandwidth throttling, dynamic blacklisting</td>
</tr>
<tr>
<td>VoIP Firewall</td>
<td>RTP pinhole management, rogue RTP detection and prevention, SIP message policy, advanced RTP latching</td>
</tr>
<tr>
<td>Encryption and Authentication</td>
<td>TLS, DTLS, SRTP, HTTPS, SSH, client/server SIP Digest authentication, RADIUS Digest</td>
</tr>
<tr>
<td>Privacy</td>
<td>Topology hiding, user privacy</td>
</tr>
<tr>
<td>Traffic Separation</td>
<td>VLAN/physical interface separation for multiple media, control and OAMP interfaces</td>
</tr>
<tr>
<td>Intrusion Detection System</td>
<td>Detection and prevention of VoIP attacks, theft of service and unauthorized access</td>
</tr>
<tr>
<td><strong>Interoperability</strong></td>
<td></td>
</tr>
<tr>
<td>SIP B2BUA</td>
<td>Full SIP transparency, mature &amp; broadly deployed SIP stack, stateful proxy mode</td>
</tr>
<tr>
<td>SIP Interworking</td>
<td>3xx redirect, REFER, PRACK, session timer, early media, call hold, delayed offer</td>
</tr>
<tr>
<td>Registration and Authentication</td>
<td>User registration restriction control, registration and authentication on behalf of users, SIP authentication server for SBC users</td>
</tr>
<tr>
<td>Transport Mediation</td>
<td>SIP over UDP/TCP/TLSPORTS/WebSocket, IPv4-IPv6, RTP-SRTP (SDES/DTLS)</td>
</tr>
<tr>
<td>Message Manipulation</td>
<td>Ability to add/modify/delete SIP headers and message body using advanced regular expressions (regex)</td>
</tr>
<tr>
<td>URI and Number Manipulations</td>
<td>URI user and host name manipulations, ingress and egress digit manipulation</td>
</tr>
<tr>
<td>Transcoding and Vocoders</td>
<td>Coder normalization including transcoding, coder enforcement and re-prioritization, extensive vocoder support: G.711, G.723.1, G.726, G.729, GSM-FR, AMR-NB, AMR-WB (G.722.2), SILK-NB/WB, Opus-NB/WB</td>
</tr>
<tr>
<td>Signal Conversion</td>
<td>DTMF/RFC 2833/SIP, T.38 fax, packet-time conversion</td>
</tr>
<tr>
<td>WebRTC Controller</td>
<td>Interworking between WebRTC devices and SIP networks</td>
</tr>
<tr>
<td></td>
<td>Supports WebSocket, Opus, VP8 video coder, lite ICE, DTLS, RTP</td>
</tr>
<tr>
<td>Function</td>
<td>Specification</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>multiplexing, secure RTCP with feedback</td>
<td></td>
</tr>
<tr>
<td>NAT</td>
<td>Local and far-end NAT traversal for support of remote workers</td>
</tr>
<tr>
<td>Voice Quality and SLA</td>
<td></td>
</tr>
<tr>
<td>Call Admission Control</td>
<td>Based on bandwidth, session establishment rate, number of connections/registrations</td>
</tr>
<tr>
<td>Packet Marking</td>
<td>802.1p/Q VLAN tagging, DiffServ, TOS</td>
</tr>
<tr>
<td>Standalone Survivability</td>
<td>Maintain local calls in the event of WAN failure.</td>
</tr>
<tr>
<td>Impairment Mitigation</td>
<td>Packet Loss Concealment, Dynamic Programmable Jitter Buffer, Silence Suppression/Comfort Noise Generation, RTP redundancy, broken connection detection</td>
</tr>
<tr>
<td>Voice Enhancement</td>
<td>Transrating, RTCP-XR, acoustic echo cancellation, replacing voice profile due to impairment detection, fixed and dynamic voice gain control</td>
</tr>
<tr>
<td>Direct Media (No Media Anchoring)</td>
<td>Hair-pinning of local calls to avoid unnecessary media delays and bandwidth consumption</td>
</tr>
<tr>
<td>Voice Quality Monitoring</td>
<td>RTCP-XR, AudioCodes Session Experience Manager (SEM)</td>
</tr>
<tr>
<td>High Availability (Redundancy)</td>
<td>SBC high availability with two-box redundancy, active calls preserved</td>
</tr>
<tr>
<td>Quality of Experience</td>
<td>Access control and media quality enhancements based on QoE and bandwidth utilization</td>
</tr>
<tr>
<td>Test Agent</td>
<td>Ability to remotely verify connectivity, voice quality and SIP message flow between SIP UAs</td>
</tr>
<tr>
<td>SIP Routing</td>
<td></td>
</tr>
<tr>
<td>Routing Methods</td>
<td>Request URL, IP address, FQDN, ENUM, advanced LDAP, third-party routing control through REST API</td>
</tr>
<tr>
<td>Advanced Routing Criteria</td>
<td>QoE, bandwidth, SIP message (SIP request, coder type, etc.), Layer-3 parameters</td>
</tr>
<tr>
<td>Redundancy</td>
<td>Detection of proxy failures and subsequent routing to alternative proxies</td>
</tr>
<tr>
<td>Routing Features</td>
<td>Least-cost routing, call forking, load balancing, E911 gateway support, emergency call detection and prioritization</td>
</tr>
<tr>
<td>SIPRec</td>
<td>IETF standard SIP recording interface</td>
</tr>
<tr>
<td>Management</td>
<td></td>
</tr>
<tr>
<td>OAM&amp;P</td>
<td>Browser-based GUI, CLI, SNMP, EMS, INI Configuration file, REST API</td>
</tr>
<tr>
<td>Multi Tenancy</td>
<td>Advanced multi-tenant SBC partitioning.</td>
</tr>
</tbody>
</table>

**Mediant VE SBC - Hardware Requirements**

Refer to the *Mediant Virtual Edition SBC Installation Manual.*

**Mediant SE SBC - Hardware Requirements**

This page is intentionally left blank.